Title: **An efficient 2D inversion scheme for airborne frequency domain data**

Authors:

Tue Boesen Tue@geo.au.dk, Aarhus University, Denmark

Esben Auken Esben.auken@geo.au.dk, Aarhus University, Denmark

Anders Vest Christiansen Anders.vest@geo.au.dk, Aarhus University, Denmark

Gianluca Fiandaca gianluca.fiandaca@geo.au.dk , Aarhus University, Denmark

Casper Kirkegaard, Casper.kirkegaard@gmail.com, QIAGEN Aarhus, Denmark

Andreas Aspmo Pfaffhuber Andreas.A.Pfaffhuber@ngi.no, Norwegian Geotechnical Institute, Norway

Malte Vöge Malte.voege@ngi.no, Norwegian Geotechnical Institute

Running head: 2D Hybrid HEM inversion

Corresponding author: Tue Boesen, Aarhus University, Denmark

Email: tue@geo.au.dk

Submission date: 05-05-2017

Key words:

2D, HEM, Hybrid, Finite element, inversion, modelling.

ABSTRACT

In many cases, inversion in 2D gives a better description of the subsurface compared to 1D inversion, but computationally 2D inversion is expensive, and it can be hard to employ for large-scale surveys. We present an efficient hybrid 2D airborne frequency-domain electromagnetic inversion algorithm. Our hybrid scheme combines 1D and 2D inversions in a three-stage process, where each step is progressively more accurate and computationally more expensive than the previous. This results in a $\sim 2\mathrm{x} - 6\mathrm{x}$ speedup compared to full 2D inversions, and with only minor changes to the inversion results. Our inversion structure is based on a regular grid, where each sounding is discretized individually. The 1D modelling code uses layered models with derivatives derived through the finite difference method, while our 2D modelling code uses an adaptive finite element mesh, and the adjoint-state method to calculate the derivatives. By incorporating the inversion grid structure into the 2D finite element mesh, interpolation between the different meshes becomes trivial. Large surveys are handled by utilizing local meshing to split large surveys into small sections, which retains the 2D information. The algorithm is heavily optimized, and parallelized over both frequencies and sections, with a good scalability even on non-uniform memory architecture systems, on which it is generally hard to achieve a satisfactory scaling. The algorithm has been tested successfully with various synthetic studies as well as field examples, of which results from two synthetic studies and a field example are shown.

INTRODUCTION

Airborne electromagnetic surveys typically contain thousands of line kilometers of data, and are routinely flown for mapping of geology, groundwater, saltwater intrusion, etc. Most data are inverted using 1D model algorithms, which have proven to be robust and computationally fast. However, specific targets with a high conductivity contrast between undulating bedrock and sediments, or conductive sheet-like mineralizations, need higher dimensionality in the underlying model to be resolved accurately (Wilson et al. 2012; Doyle et al. 1999; Yang and Oldenburg 2012b). The challenge in moving beyond 1D modeling is that it is prohibitively computationally expensive to invert for a 2D or a 3D model and this limits the usage of these inversion algorithms on a routine basis, even for frequency domain-electromagnetic datasets of just a few discrete frequencies.

Full 3D EM inversion algorithms have existed for more than a decade (Haber et al. 2007b), and with the concept of moving footprint (Cox et al. 2010), several 3D codes using local meshes have been presented (Cox et al. 2012; Yang et al. 2014). All these algorithms are capable of handling large surveys, in time- or frequency-domain, by sectioning the survey into smaller parts using a local meshing approach. Local meshing means that the survey is split into smaller parts, where each part contains a small subset of transmitter-receiver pairs, as well as all the models within their footprint domain (Liu and Becker 1990; Beamish 2003; Reid et al. 2006). Here, the footprint is defined as the area of significant lateral sensitivity of the survey system, and its size is thus dependent upon both the system itself and the resistivity of the earth. Reid et al. (2006) showed that, for a frequency domain system, the footprint may be as large as 10 times the flight altitude for low induction numbers. Common survey systems operate either in time-domain, or with multiple frequencies spread across the frequency spectrum, and while these latter systems usually have one transmitter frequency operating within the low induction approximation, the majority of their transmitter frequencies operate at higher induction numbers, for which the footprint is much smaller. From this, we argue that when a survey is flown with a line separation of 200-500 m, the majority of any crossline information is lost, and the survey results are essentially reduced to only contain inline information. Considering this and the inherent computational burden of doing full 3D inversions, it is clear that there are areas where it is sufficient, and even desirable to operate within a 2D formulation. Several 2D inversion algorithms have been presented over the years: Mitsuhata and Uchida (2002); Wilson et al. (2006); Li et al. (2016); Key and Ovall (2011) have all developed 2D finite element algorithms, while Abubakar et al. (2008) uses a finite difference approach, and Yu and Haber (2012) present a finite volume approach. In general, finite difference approaches are considered the most simple and inaccurate of the three approaches, but can sometimes be justified due to their superior parallel scaling. The finite element is the most accurate of the methods, but also the most computationally heavy, and at large the question of whether the finite element or finite volume is the superior choice remains open (Jahandari et al. 2017).

In this paper, we present a hybrid 1D/2D inversion code for frequency-domain HEM data designed for inverting field scale surveys on desktop computers. Since an efficient 2D modelling algorithm is vital to achieve this goal, and since our 2D algorithm has not previously been published, this paper begins with the construction of our 2D algorithm and the foundation it is built on. The 2D algorithm is based on the 2.5D formulation by Stoyer and Greenfield (1976), along with field separation into primary and secondary fields, where the high frequency singularity is handled by the introduction of a finite resistivity of the air (Mitsuhata 2000). The algorithm has a triangular finite element mesh for the 2D forward and derivative calculations, and a regular grid for the inversion. When having multiple meshes, interpolation schemes are needed to map variables between the meshes. In general, interpolation between meshes is a non-trivial task (Caudillo-Mata et al. 2016), but in our case the task is made trivial, by using the regular grid as a skeletal structure for building the finite element mesh. Due to limited memory, as well as performance concerns, we introduce sectioning, which splits large survey lines into smaller sections. Sectioning is only done when carrying out the 2D forward and derivative calculations, during which we enforce sufficient overlap, such that vital 2D information is preserved. Based on the overlap size, we show how section sizes should be chosen in order to reach optimal performance. The algorithm is written in Fortran and utilizes: OpenMP, Intel MKL libraries, as well as a custom-built block-parallel sparse iterative linear solver. The algorithm is part of AarhusInv, which is provided as freeware for non-commercial academic purposes (Auken et al. 2014).

Following the presentation of our 2D algorithm, we present a hybrid scheme, inspired by the work presented in Christiansen et al. (2015); Christiansen and Auken (2004). The method starts by performing 1D forward and inverse calculations, later it switches to 2D forward calculations and 1D derivatives, and finally it ends with full 2D calculations. The result of this is a code, which produces 2D results, but with a substantially shorter computational time than traditional 2D algorithms. We demonstrate these computational benefits using two synthetic models and a field example. Finally, we discuss the trade-off between computational speed and accuracy, how the algorithm is best parallelized, and we illustrate the code's parallel scaling and performance on a multiprocessor system.

METHODOLOGY

The 2.5D forward algorithm is based upon the framework of Stoyer and Greenfield (1976). In a 2.5D formulation the earth is described in 2D (homogeneous in the cross-section-direction), but the source is 3D. The source needs to be 3D, in order to describe accurately the sources used in AEM, since AEM sources do not produce source fields, which are reasonably homogenous in any direction. Our 2.5D algorithm was originally developed for marine EM measurements (Vöge 2010), but sources in the air have been added so it can now be used to invert airborne frequency domain EM data (Vöge et al. 2015). For hybrid inversion we include the 1D algorithm of Auken et al. (2014), which is based on the layered 1D model solution presented in Ward and Hohmann (1988).

Governing equations

Starting from Maxwell's equations in the frequency domain:

$$\boldsymbol{\nabla} \times \boldsymbol{E} + i\omega\boldsymbol{\mu} \cdot \boldsymbol{H} = 0, \tag{1}$$

$$\nabla \times \boldsymbol{H} - i\omega\boldsymbol{\varepsilon} \cdot \boldsymbol{E} = \boldsymbol{J}, \tag{2}$$

where $\boldsymbol{E}$ and $\boldsymbol{H}$ are the electric- and magnetic-fields, $\boldsymbol{J}$ is the electric source current, i is the imaginary unit, ω is the frequency, $\boldsymbol{\mu}$ is the magnetic permeability, and $\boldsymbol{\varepsilon}$ is the complex dielectric permittivity, with $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon_0} - i\frac{\sigma}{\omega}$, where σ is the conductivity. In order to minimize the forward inaccuracy, the fields are split into primary and secondary fields:

$$\boldsymbol{E} = \boldsymbol{E_p} + \boldsymbol{E_s}, \boldsymbol{H} = \boldsymbol{H}_p + \boldsymbol{H_s}. \quad (3)$$

So equation 1 and 2 can be written as

$$\left(\nabla \times \boldsymbol{E_p} + i\omega\boldsymbol{\mu} \cdot \boldsymbol{H}_p\right) + (\nabla \times \boldsymbol{E_s} + i\omega\boldsymbol{\mu} \cdot \boldsymbol{H}_s) = 0, \quad (4)$$

$$\left(\nabla \times \boldsymbol{H_p} - i\omega\boldsymbol{\varepsilon} \cdot \boldsymbol{E}_p\right) + (\nabla \times \boldsymbol{H}_s - i\omega\boldsymbol{\varepsilon} \cdot \boldsymbol{E_s}) = J. \quad (5)$$

We separate the conductivity into a conductivity for the primary field model, $\boldsymbol{\sigma}_p$, and secondary field model, $\boldsymbol{\sigma}_s$, where $\boldsymbol{\sigma}_s = \boldsymbol{\sigma} - \boldsymbol{\sigma}_p$. From this we get $i\omega\boldsymbol{\varepsilon} = i\omega\boldsymbol{\varepsilon_0} - \left(\boldsymbol{\sigma}_p + \boldsymbol{\sigma}_s\right) = i\omega\boldsymbol{\varepsilon_p} - \boldsymbol{\sigma}_s$, , which allows us to split equation 4 and 5 into separate equation systems for primary field: :

$$\nabla \times \boldsymbol{E_p} + i\omega\boldsymbol{\mu} \cdot \boldsymbol{H}_p = 0, \quad (6)$$

$$\nabla \times \boldsymbol{H_p} - i\omega\boldsymbol{\varepsilon_p} \cdot \boldsymbol{E}_p = \boldsymbol{J}, \quad (7)$$

and the secondary field:

$$\nabla \times \boldsymbol{E_s} + i\omega\boldsymbol{\mu} \cdot \boldsymbol{H}_s = 0, \quad (8)$$

$$\nabla \times \boldsymbol{H}_s - i\omega\boldsymbol{\varepsilon} \cdot \boldsymbol{E_s} = \boldsymbol{\sigma}_s \cdot \boldsymbol{E}_p. \quad (9)$$

The primary field is computed analytically, so the inaccuracy of the finite element method affects only the secondary field, which is several orders of magnitude smaller than the total field. In our case, we chose the primary field model to be a uniform full-space air model with magnetic point sources and receivers, which can be easily calculated analytically. Since $\boldsymbol{\sigma}_p$ is the conductivity of air, we have $\boldsymbol{\sigma} = \boldsymbol{\sigma}_p$ and $\boldsymbol{\sigma}_s = 0$ in the air layer.

The Fourier transform is defined with respect to y (i.e. the strike direction of the survey) as:

$$\tilde{F}(x, k_y, z) = \int_{-\infty}^{+\infty} F(x, y, z) e^{ik_y y} dy, \tag{10}$$

where k_y is the wavenumber. A Fourier transformation of the primary field is carried out numerically, following the approach of Streich et al. (2011). The governing equations of the 2D forward response emerge by applying the Fourier transform to equation 8 and 9:

$$\tilde{E}_{sx} = \frac{1}{k_y^2 - \omega^2 \mu_z \varepsilon_x} \left(ik_y \frac{\partial \tilde{E}_{sy}}{\partial x} - i\omega\mu_z \frac{\partial \tilde{H}_{sy}}{\partial z} - i\omega\mu_z \sigma_{sx} \tilde{E}_{px} \right), \tag{11}$$

$$\tilde{E}_{sz} = \frac{1}{k_y^2 - \omega^2 \mu_x \varepsilon_z} \left(ik_y \frac{\partial \tilde{E}_{sy}}{\partial z} + i\omega\mu_x \frac{\partial \tilde{H}_{sy}}{\partial x} - i\omega\mu_x \sigma_{sz} \tilde{E}_{pz} \right), \tag{12}$$

$$\tilde{H}_{sx} = \frac{1}{k_y^2 - \omega^2 \mu_x \varepsilon_z} \left(i\omega\varepsilon_z \frac{\partial \tilde{E}_{sy}}{\partial z} + ik_y \frac{\partial \tilde{H}_{sy}}{\partial x} - ik_y \sigma_{sz} \tilde{E}_{pz} \right), \tag{13}$$

$$\tilde{H}_{sz} = \frac{1}{k_y^2 - \omega^2 \mu_z \varepsilon_x} \left(-i\omega\varepsilon_x \frac{\partial \tilde{E}_{sy}}{\partial x} + ik_y \frac{\partial \tilde{H}_{sy}}{\partial z} + ik_y \sigma_{sx} \tilde{E}_{px} \right), \tag{14}$$

$$
\begin{aligned}
-i\omega\varepsilon_y\tilde{E}_{sy} &+ \frac{\partial}{\partial x}\left(\frac{i\omega\varepsilon_x}{c_{zx}}\frac{\partial\tilde{E}_{sy}}{\partial x}\right) + \frac{\partial}{\partial z}\left(\frac{i\omega\varepsilon_z}{c_{xz}}\frac{\partial\tilde{E}_{sy}}{\partial z}\right) - \frac{\partial}{\partial x}\left(\frac{ik_y}{c_{zx}}\frac{\partial\widetilde{H}_{sy}}{\partial z}\right) \\
&+ \frac{\partial}{\partial z}\left(\frac{ik_y}{c_{xz}}\frac{\partial\widetilde{H}_{sy}}{\partial x}\right) \\
&= -\left(i\omega\varepsilon_{py} - i\omega\varepsilon_y\right)\tilde{E}_{py} \\
&+ \frac{\partial}{\partial x}\left(\left(\frac{i\omega\varepsilon_{px}}{c_{pzx}} - \frac{i\omega\varepsilon_x}{c_{zx}}\right)\frac{\partial\tilde{E}_{py}}{\partial x}\right) \\
&+ \frac{\partial}{\partial z}\left(\left(\frac{i\omega\varepsilon_{pz}}{c_{pxz}} - \frac{i\omega\varepsilon_z}{c_{xz}}\right)\frac{\partial\tilde{E}_{py}}{\partial z}\right) \\
&- \frac{\partial}{\partial x}\left(\left(\frac{ik_y}{c_{pzx}} - \frac{ik_y}{c_{zx}}\right)\frac{\partial\widetilde{H}_{py}}{\partial z}\right) \\
&+ \frac{\partial}{\partial z}\left(\left(\frac{ik_y}{c_{pxz}} - \frac{ik_y}{c_{xz}}\right)\frac{\partial\widetilde{H}_{py}}{\partial x}\right),
\end{aligned}
\tag{15}
$$

and

$$
\begin{aligned}
-i\omega\mu_y\widetilde{H}_{sy} &+ \frac{\partial}{\partial x}\left(\frac{i\omega\mu_x}{c_{xz}}\frac{\partial\widetilde{H}_{sy}}{\partial x}\right) + \frac{\partial}{\partial z}\left(\frac{i\omega\mu_z}{c_{zx}}\frac{\partial\widetilde{H}_{sy}}{\partial z}\right) + \frac{\partial}{\partial x}\left(\frac{ik_y}{c_{xz}}\frac{\partial\tilde{E}_{sy}}{\partial z}\right) \\
&- \frac{\partial}{\partial z}\left(\frac{ik_y}{c_{zx}}\frac{\partial\tilde{E}_{sy}}{\partial x}\right) \\
&= -\left(i\omega\mu_{py} - i\omega\mu_y\right)\widetilde{H}_{py} + \frac{\partial}{\partial x}\left(\left(\frac{i\omega\mu_{px}}{c_{pxz}} - \frac{i\omega\mu_x}{c_{xz}}\right)\frac{\partial\widetilde{H}_{py}}{\partial x}\right) \\
&+ \frac{\partial}{\partial z}\left(\left(\frac{i\omega\mu_{pz}}{c_{pzx}} - \frac{i\omega\mu_z}{c_{zx}}\right)\frac{\partial\widetilde{H}_{py}}{\partial z}\right) + \frac{\partial}{\partial x}\left(\left(\frac{ik_y}{c_{pxz}} - \frac{ik_y}{c_{xz}}\right)\frac{\partial\tilde{E}_{py}}{\partial z}\right) \\
&- \frac{\partial}{\partial z}\left(\left(\frac{ik_y}{c_{pzx}} - \frac{ik_y}{c_{zx}}\right)\frac{\partial\tilde{E}_{py}}{\partial x}\right),
\end{aligned}
\tag{16}
$$

where $c_{ij} = k_y^{\ 2} - \omega^2 \mu_i \varepsilon_j$. Equations 11-14 only need to be evaluated at the receiver positions, which in our application are in the air. Thus, σ_{sx}, σ_{sy} and σ_{sz}, are always zero, so all primary field terms in equation 11-14 can be ignored. While the left-hand-sides of equation 15-16 are identical to the governing equations presented in Mitsuhata (2000) , the right-hand-sides are expressed by $\widetilde{E}_{py}$ and $\widetilde{H}_{py}$ instead of $\widetilde{E}_{px}$, $\widetilde{E}_{py}$ and $\widetilde{E}_{pz}$. However, a simple arithmetic reformulation of the right-hand-side of equation 15 using equations 11-14 can show that both forms are equivalent (not shown). Having the same field components and derivatives on both sides of the equations, allows us to speed-up the assembly of the linear equation system.

With the governing equations defined in equation 11-16, we use the standard finite element approach to define a set of local equations for each element. By combining all these using the Galerkin method (Zienkiewicz et al. 1977) with 2nd order nodal elements and Dirichlet boundary conditions, a global set of linear equations is found for the secondary electromagnetic fields (See appendix A for more details). From equation 15-16 a linear system of equations for the secondary field is found:

$$\boldsymbol{A}\widetilde{\boldsymbol{x}} = \boldsymbol{b}, \tag{17}$$

where $\boldsymbol{A}$ is the global symmetric stiffness matrix, $\widetilde{\boldsymbol{x}}$ contains the Fourier transformed secondary fields $\widetilde{E}_{sy}$ and $\widetilde{H}_{sy}$ at the mesh nodes, and $\boldsymbol{b}$ contains the source terms, where each column represents one source component.

The procedure to solve the system is as follows:

- Assemble the matrix equation resulting from equation 17
- Solve the linear system using a direct LU-decomposition solver and interpolate the field values at the receiver positions using the same 2nd order shape function (which has already been used to assemble the finite element system) to find $\widetilde{\boldsymbol{x}}$, which contains the Fourier transformed secondary field components $\widetilde{E}_{sy}$ and $\widetilde{H}_{sy}$.

- Insert the solution into equation 11-14 to find the remaining components of the Fourier transformed fields of $\tilde{E}_{sx}$, $\tilde{E}_{sz}$, $\tilde{H}_{sx}$ and $\tilde{H}_{sz}$.
- Interpolate the solution to the receiver positions.
- Apply the inverse Fourier transform in order to obtain the fields E_{sx}, E_{sy}, E_{sz} , H_{sx} H_{sy}, and H_{sz} at the receiver positions in the frequency-domain, which, when all combined, are referred to as the forward response vector $\boldsymbol{d}$.

Given this procedure, the forward response can formally be written as

$$\boldsymbol{d} = \boldsymbol{\mathcal{F}}^{-\mathbf{1}}(\beth(\tilde{\boldsymbol{x}})), \tag{18}$$

Where $\boldsymbol{\mathcal{F}}^{-\mathbf{1}}$ is the inverse Fourier transform operator and $\beth$ is the interpolating operator.

This inverse Fourier transform is done numerically by logarithmic spaced k_y-samples, which are splined together over the relevant k_y-domain. Tests show that five wavenumbers per decade between $10^{-5}\ \mathrm{m}^{-1}$ to $10\ \mathrm{m}^{-1}$ provide sufficiently accurate results. One important point related to the inverse Fourier transform is that the air conductivity needs to be larger than zero, otherwise a singularity at $k_y^2 \approx \omega^2\mu\varepsilon$ is encountered (Mitsuhata 2000). We found that setting the air conductivity to $\sigma = 10^{-6}\ \mathrm{Sm}^{-1}$ keeps the air sufficiently resistive, while avoiding the singularity within a frequency range of 0.4-130 kHz. The interpolation to the receiver positions is carried out using the shape functions of the finite elements.

Derivative calculation

The 2D derivatives of the forward response with regards to the model parameters, $\boldsymbol{m}$, can be written as:

$$\frac{d\boldsymbol{d}}{d\boldsymbol{m}} = \frac{d(\boldsymbol{\mathcal{F}}^{-\mathbf{1}}(\beth(\tilde{\boldsymbol{x}})))}{d\boldsymbol{m}}. \tag{19}$$

For model parameters related to source/receiver altitude; the derivatives are calculated by a standard finite difference approach with small perturbations, as done in the 1D case (Auken et al. 2014). For model parameters related to subsurface resistivities, $\boldsymbol{\rho}$; the derivatives are calculated as follows.

For inversion parameters related to resistivities, both the Fourier transform operator and the interpolation operator are independent of the inversion parameter, so we can write

$$\frac{d\boldsymbol{d}}{d\boldsymbol{\rho}} = \boldsymbol{\mathcal{F}}^{-1}\left(\beth\left(\frac{d\widetilde{\boldsymbol{x}}}{d\boldsymbol{\rho}}\right)\right) \tag{20}$$

In 2D the derivative of $\widetilde{\boldsymbol{x}}$ is found through the adjoint-state method (McGillivray and Oldenburg 1990). Equation 17 is differentiated with regards to the model parameters $\boldsymbol{m}$:

$$\frac{d(\boldsymbol{A}\widetilde{\boldsymbol{x}})}{d\boldsymbol{\rho}} = \frac{d\boldsymbol{b}}{d\boldsymbol{\rho}}, \tag{21}$$

which through the product rule gives the Jacobi elements:

$$\frac{d\widetilde{\boldsymbol{x}}}{d\boldsymbol{\rho}} = \boldsymbol{A}^{-1}\left(\frac{d\boldsymbol{b}}{d\boldsymbol{\rho}} - \frac{d\boldsymbol{A}}{d\boldsymbol{\rho}}\widetilde{\boldsymbol{x}}\right). \tag{22}$$

Since only the coefficients of the governing equations are depended on the resistivity, $\frac{d\boldsymbol{A}}{d\boldsymbol{\rho}}$ can be analytically calculated and assembled. The term $\frac{d\boldsymbol{b}}{d\boldsymbol{\rho}}$ is zero, because all sources are in the air, and thus are not affected by any of the inverted resistivity cells. As each inversion cell usually contains only a small number of finite elements, $\frac{d\boldsymbol{A}}{d\boldsymbol{\rho}}$ is extremely sparse and $\frac{d\boldsymbol{A}}{d\boldsymbol{\rho}}\widetilde{\boldsymbol{x}}$ can be calculated efficiently, and the result can still be considered sparse. To calculate $\boldsymbol{A}^{-1}$, however, would be far too expensive. Instead, we use the fact that only the field derivatives at the receiver position are of interest, and thus replacing $\boldsymbol{A}^{-1}$ with $\boldsymbol{\lambda}^T$, where $\boldsymbol{\lambda}^T$ contains all rows of $\boldsymbol{A}^{-1}$ that correspond to those column in $\frac{d\widetilde{\boldsymbol{x}}}{d\boldsymbol{\rho}}$ which are necessary to calculate the field values at the receiver positions:

$$\frac{d\tilde{x}_{rec}}{d\boldsymbol{\rho}} = \boldsymbol{\lambda}^T \left(\frac{d\boldsymbol{b}}{d\boldsymbol{\rho}} - \frac{d\boldsymbol{A}}{d\boldsymbol{\rho}} \tilde{\boldsymbol{x}}\right). \tag{23}$$

Because $\boldsymbol{A}$ is symmetric, $\boldsymbol{\lambda}$ can be calculated by solving $\boldsymbol{A\lambda} = \boldsymbol{I}_{rec}$, with $\boldsymbol{I}_{rec}$ being constructed from those columns of the identity matrix necessary to calculate the field values at the receivers. Thus, the same direct LU-decomposition used for the regular forward solutions can be used here. The matrix multiplication in equation 21 then results in $\frac{d\tilde{x}_{rec}}{d\boldsymbol{\rho}}$, which is a dense matrix, however, with rather small dimensions, where the number of rows is equal to the number of sources, and the number of columns is equal to 12 times the number of receivers (6 nodes per finite element with 2 field components each). The derivatives at the receiver positions are then calculated from $\frac{d\tilde{\boldsymbol{x}}}{d\boldsymbol{\rho}}$ using the second order shape function as interpolator**,** and the derivative of the forward response are calculated by the interpolated derivatives given in equation 20 .

Meshing

The 2D modelling is performed on a triangular finite element mesh as shown in Figure 1 (b), while the inversion operates on a regular grid, as seen in Figure 1 (a). The column spacing of the inversion grid is determined by the sounding distance, the row spacing by the model layers, and the layer thickness is chosen to be logarithmically increasing, which reflects the decreasing sensitivity of HEM systems with depth. Separating the meshes of forward calculations and inversion has some clear computational benefits, as the inversion grid is much coarser than the 2D forward mesh. This decreases the size of the inversion problem, while maintaining the accuracy of the forward modelling.

Interpolation between the inversion grid and the forward mesh is avoided by using the inversion grid as a skeletal structure for the forward mesh (Figure 1). By incorporating the inversion grid into the forward mesh, it is guaranteed that each finite element is fully residing in just one inversion cell, which aligns the forward modelling mesh nodes and edges with the inversion grid.

The mesh density for the forward mesh is adjusted according to the key parameters like frequency and source/receiver height. The highest mesh density is needed near the surface below the sources, where the primary field is strongest. Here, the mesh density is selected as a function of the source height and, thus, of the strength of the primary field. For very low altitudes of 1m and below, a maximum edge length of 0.2 m is required. For altitudes of 20 m and above, 5 m edge length is sufficient. At deeper locations and at larger horizontal offsets from the source, the mesh density can be reduced without losing accuracy. The mesh density is interpolated between the surface/zero offset mesh density, defined by the altitude, and a background mesh density of 50m for larger depths and offsets. This interpolation is done by a 2D Gaussian function, with the 2D distance from the closest source/receiver as parameter and a frequency dependent standard deviation. Standard deviations in z direction are logarithmically interpolated between 200 m at 10^{-2} Hz and 20 m at 10^{6} Hz. Test showed, that the mesh density along the surface could not be coarsened as quickly, so the standard deviation in x direction is logarithmically interpolated between 800 m at 10^{-2} Hz and 80 m at 10^{6} Hz. Additionally, the mesh density is increased near the receivers, in order to calculate the spatial derivatives in equations (11) and (12) accurately.

The mesh of the 2D forward model is appended with large absorbing boundary domains that extend 10 km in each direction. As the mesh density is coarsening quickly in these boundary domains, the computational overhead is not very high, but tests showed that 10 km boundary domains allow the field to attenuate enough, so that the validity of the Dirichlet boundary conditions is assured.

Sectioning

Even for relatively small surveys, it is computationally inefficient to create and store a sufficiently fine finite element mesh and do 2D calculations on all soundings at once. Because of this, it is imperative to split large surveys into smaller sections. Sectioning, or local meshing as it is also often called in the literature, can be accomplished in several different ways. Our sectioning method is somewhat similar to the method used in Yang and Oldenburg (2012a). Their method involves a global mesh, and a local mesh for each sounding. While the forward problem is handled on the local meshes, the inversion is made by subsampling the global mesh. In our case the local meshes contain multiple soundings, since that is more efficient, and is used for both the 2D forward and derivative calculations. In order for these sections to retain the 2D information of the survey, they need to overlap as shown in Figure 2. Thus, each section, L, consists of a core section, l, and one or two overlapping regions, Δl. Continuity between different sections are ensured by using sufficient overlap between different sections, and by placing lateral constraints on all soundings irregardless of section boundaries. The size of the overlap and sections will be addressed later.

Forward modelling validation

The 2D finite element forward response was validated against the 1D code of Auken et al. (2014) for a range of frequencies relevant for HEM (0.4-130 kHz) across different half spaces with resistivities of: 10 Ωm, 100 Ωm, and 1000 Ωm. A halfspace comparison between 1D and 2D is shown in Figure 3. The mesh density is selected such that the resulting responses deviate less than 5 % from the 1D responses within the frequency range. The inaccuracy of the 1D response is estimated to be between 0.1–0.3 % and is insignificant in this context. Note that the deviation between 1D and 2D responses is not just a single number, but instead a range, because of variations in the mesh density between soundings near the edge of the mesh and those near the center. The overall coarseness surrounding a sounding near the center of a section is will be slightly lower than the overall coarseness surrounding a sounding near the edge of a section. This is reflected in Figure 3 (b), where the deviation from 1D is shown for a 300 m long section with 30 soundings equally spaced over the section. The flight height of the system is 30 m and the halfspace resistivity is set to 100 Ωm. Similar accuracies are obtained from halfspace resistivities at 10 Ωm and 1000 Ωm, but for brevity we only show one representative example. In this case, the inaccuracy is generally less than 2%, while reaching as high as 5% for frequencies beyond the range shown here.

Inversion algorithm

Our inversion technique utilizes linearized minimization, following the Levenberg-Marquardt adaptive scheme (Menke 1989). The following is a brief review of our inversion algorithm, see Auken and Christiansen (2004); Auken et al. (2014) for the full details.

The minimized objective function is given as:

$$q = q_{obs} + q_{prior} + q_{reg}, \tag{24}$$

with q_{obs} being the observed data (secondary field) misfit, q_{prior} being the prior constraint misfit, and q_{reg} being the regularization misfit. Smooth regularizations are used both laterally and vertically. To determine the misfit, we use a standard least-square solution (L2-norm). With this, the n'th iterative update of the model vector $\boldsymbol{m}$ is given as:

$$\boldsymbol{m}_{n+1} = \boldsymbol{m}_n + \left(\widehat{\boldsymbol{G}}_n^T \widehat{\boldsymbol{C}}_n^{-1} \widehat{\boldsymbol{G}}_n \ + \lambda_n \boldsymbol{I}\right)^{-1} \cdot \left(\widehat{\boldsymbol{G}}_n^T \widehat{\boldsymbol{C}}_n^{-1} \delta\widehat{\boldsymbol{d}}_n\right) \tag{25}$$

where $\boldsymbol{I}$ is the identity matrix, λ is the damping parameter (Marquardt 1963), $\delta\widehat{\boldsymbol{d}}$ is the extended perturbed data vector, $\widehat{\boldsymbol{G}}$ is the extended Jacobian, and $\widehat{\boldsymbol{C}}$ is the extended covariance matrix, where the extensions comes from the inclusion of prior information and regularization:

$$\delta\widehat{\boldsymbol{d}} = \begin{bmatrix} \boldsymbol{d} - \boldsymbol{d}_{obs} \\ \boldsymbol{m} - \boldsymbol{m}_{prior} \\ -\boldsymbol{Rm} \end{bmatrix} \tag{26}$$

$$\widehat{\boldsymbol{G}} = \begin{bmatrix} \boldsymbol{G} \\ \boldsymbol{P} \\ \boldsymbol{R} \end{bmatrix} \tag{27}$$

$$\widehat{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C}_{obs} & 0 & 0 \\ 0 & \boldsymbol{C}_{prior} & 0 \\ 0 & 0 & \boldsymbol{C}_{reg} \end{bmatrix} \tag{28}$$

where, $\boldsymbol{d}$ is the forward response (see above), $\boldsymbol{d}_{obs}$ is the observed data, $\boldsymbol{m}$ is the model parameters, $\boldsymbol{m}_{prior}$ is the a priori model parameters, $\boldsymbol{R}$ is the roughness matrix, which binds neighboring models/model-parameters together, $\boldsymbol{G}$ is the Jacobian (see above), $\boldsymbol{P}$ is a matrix containing the a priori information, $\boldsymbol{C}_{obs}$ is the covariance of the observed data, $\boldsymbol{C}_{prior}$ is the covariance of the a priori information, and $\boldsymbol{C}_{reg}$ is the covariance stemming from the roughness matrix.

Calculating the iterative model update as shown in equation 25, requires solving a large linear system. In 1D, this system is sparse, but in 2D the linear system is in principle dense. However, in practice it can be assumed sparse if only the part with most sensitivity is considered (this will be covered in more detail later). Nevertheless, the 2D linear system will always be considerably less sparse than the 1D case. Solving large sparse linear systems is non-trivial and the optimal approach is dependent on the system being solved. Our current approach to solving the linear system in the 1D case is thoroughly described in Kirkegaard and Auken (2015), and starts with a reverse Cuthill-Mckee reordering algorithm (Cuthill and McKee 1969), which is used on the ordering of the initial soundings. This results in the matrix being created in such a way that all vital non-zero elements lie relatively close to the diagonal, while retaining the sounding structure in the matrix. The actual matrix is solved in parallel using an iterative sparse solver, which uses CG propagation (Hestenes and Stiefel 1952; Saad 2003), along with a preconditioner, which depends on the dimensionality of the inversion problem. For the 1D inversion problem our method of choice is a block-parallelized version of an incomplete LU factorization with a dual dropping strategy (Saad 1994). However, due to the increased bandwidth of the sparse matrix in the 2D case, consistent convergence is not obtained when applying the LU decomposition as a preconditioner. Direct solvers work well for small surveys (up to around 5,000 soundings), but for larger surveys direct solvers become inefficient due to memory consumption as well as factorization time. Neither scale linearly with the size of the survey. Instead, we have found that applying the symmetric-Gauss-Seidel (SGS) preconditioner leads to stable convergence when doing 2D inversions. Furthermore, if applied in cases where the linear system is sufficiently diagonally dominant, the SGS preconditioner is even more efficient than the incomplete LU factorization, and can lead to a significant speedup in 1D inversions.

Optimizing section sizes

Previously, we discussed the need for dividing large surveys into smaller sections due to memory concerns. However, even if memory had not been an issue it still proves computationally advantageous to split a survey into smaller sections. The reason for this is that for unstructured meshes, the computational time scales quadratically with the number of elements as the number of elements becomes large. On the other hand, there is also a size-independent initialization cost associated with each section that needs to be considered. This cost comes from setting up the mesh padding, establishing the equations, spawning the parallel thread pool, and other similar tasks. Even more importantly, it also needs to be assured that each section overlaps its neighboring section by a fixed amount, in order to retain the 2D information from the survey.

Analysis of our parallel algorithm has led to the identification of an optimal section size. One that is defined by the initial computational cost, the quadratic computational scaling with section size, as well as the overlap size. In order to find this optimal section size, performance tests for the RESOLVE system (shown in Table 1) were conducted. The experiments were performed over a sweep of section sizes, and the results for both forward and derivative computations can be seen in Figure 4 (a).

As seen in Figure 4 (a), the computational times present a global minimum, different for forward and derivative calculations. The reason why the derivative calculation favors smaller section sizes than the forward calculation is due to the heavier computational burden associated with derivative calculations. This increased computational burden makes the initialization cost less significant and thus naturally shifts the optimal section size for derivative calculation towards smaller sizes. With the results presented in Figure 4 (a), the optimal section size as a function of the overlap is determined as shown in Figure 4 (b). The optimal section size is determined by using the data in Figure 4 (a). By subtracting two times the desired overlap from the section sizes given in Figure 4 (a), and interpolating the remaining positive core section sizes, an estimate of the computational time for a given section size with a given overlap region can be determined (not shown). In order to find the optimal section size, the section sizes that fall within 5 % of the fastest time for a given overlap are used and shown in Figure 4 (b). Note that, once again, the optimal section sizes are different for forward and derivative calculations. One caveat to this is that the results in Figure 4 (b) change depending on the 2D finite element mesh density, which changes slightly between different surveys and systems. Therefore, Figure 4 (b) should not be considered the absolute truth, but rather serve as a guide for picking a sensible section size based on overlap size.

2D Jacobian and sensitivity analysis

As mentioned earlier, the structure of the 2D Jacobian is a dense matrix. However, due to the decay in sensitivity as a function of distance, a threshold can be defined, where anything that falls below this threshold is assumed negligible. Thus, in practice our Jacobian matrix can still be considered sparse even in 2D, even though it has considerably wider non-zero bands around the diagonal than in the 1D case. Our 2D Jacobian matrix resulting from the inversion grid is shown in equation 29-30 . Where equation 29 shows our 2D Jacobian in block form, where each column/row refers to a single sounding. Note that the 1D Jacobian structure is identical to the one for the 2D Jacobian, but in the 1D case all the off-diagonal blocks shown in equation 29 would be zero. equation 30 shows the structure of a single Jacobian block element, which contains a number of elements equal to the number of perturbable model parameters for this particular model (altitude, and resistivities) times the number of data points for the corresponding sounding.

$$\boldsymbol{G} = \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\ \cdots & \frac{\partial \boldsymbol{D}_{i-2}}{\partial \boldsymbol{M}_{i-2}} & \frac{\partial \boldsymbol{D}_{i-2}}{\partial \boldsymbol{M}_{i-1}} & 0 & 0 & 0 & \cdots \\ \cdots & \frac{\partial \boldsymbol{D}_{i-1}}{\partial \boldsymbol{M}_{i-2}} & \frac{\partial \boldsymbol{D}_{i-1}}{\partial \boldsymbol{M}_{i-1}} & \frac{\partial \boldsymbol{D}_{i-1}}{\partial \boldsymbol{M}_{i}} & 0 & 0 & \cdots \\ \cdots & 0 & \frac{\partial \boldsymbol{D}_{i}}{\partial \boldsymbol{M}_{i-1}} & \frac{\partial \boldsymbol{D}_{i}}{\partial \boldsymbol{M}_{i}} & \frac{\partial \boldsymbol{D}_{i}}{\partial \boldsymbol{M}_{i+1}} & 0 & \cdots \\ \cdots & 0 & 0 & \frac{\partial \boldsymbol{D}_{i+1}}{\partial \boldsymbol{M}_{i}} & \frac{\partial \boldsymbol{D}_{i+1}}{\partial \boldsymbol{M}_{i+1}} & \frac{\partial \boldsymbol{D}_{i+1}}{\partial \boldsymbol{M}_{i+2}} & \cdots \\ \cdots & 0 & 0 & 0 & \frac{\partial \boldsymbol{D}_{i+2}}{\partial \boldsymbol{M}_{i+1}} & \frac{\partial \boldsymbol{D}_{i+2}}{\partial \boldsymbol{M}_{i+2}} & \cdots \\ \iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{29}$$

Illustrates our 2D Jacobian in block-matrix form, where the number of off-diagonal bands is equal to the number of surrounding soundings above the sensitivity threshold (here, only the nearest neighbor is above the threshold). Each entry in the Jacobian block matrix is a dense matrix block, which is given as:

$$\frac{\partial \boldsymbol{D}_j}{\partial \boldsymbol{M}_k} = \begin{pmatrix} \frac{\partial d_{j,1}}{\partial m_{k,1}} & \frac{\partial d_{j,1}}{\partial m_{k,2}} & \frac{\partial d_{j,1}}{\partial m_{k,3}} & \cdots & \frac{\partial d_{j,1}}{\partial m_{k,N_{m_k}}} \\ \frac{\partial d_{j,2}}{\partial m_{k,1}} & \frac{\partial d_{j,2}}{\partial m_{k,2}} & \frac{\partial d_{j,2}}{\partial m_{k,3}} & \cdots & \frac{\partial d_{j,2}}{\partial m_{k,N_{m_k}}} \\ \frac{\partial d_{j,3}}{\partial m_{k,1}} & \frac{\partial d_{j,3}}{\partial m_{k,2}} & \frac{\partial d_{j,3}}{\partial m_{k,3}} & \cdots & \frac{\partial d_{j,3}}{\partial m_{k,N_{m_k}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial d_{j,N_{d_j}}}{\partial m_{k,1}} & \frac{\partial d_{j,N_{d_j}}}{\partial m_{k,1}} & \frac{\partial d_{j,N_{d_j}}}{\partial m_{k,1}} & \cdots & \frac{\partial d_{j,N_{d_j}}}{\partial m_{k,N_{m_k}}} \end{pmatrix} \tag{30}$$

where j and k represent the individual sounding indices, and $\boldsymbol{D}_j$ contains $N_{d\,j}$ forward responses associated with the j'th sounding. While $\boldsymbol{M}_k$ contains $N_{m\,k}$ model parameters associated with sounding k.

Accurately determining the resulting sensitivity range is important, not just when building the Jacobian, but also when optimizing section size. This is due to the obvious connection between the sensitivity threshold distance, and the required overlap distance between adjacent sections. To determine the sensitivity threshold, we follow the convention of Liu and Becker (1990) and define a significant sensitivity range as the distance at which 90% of the full sensitivity is contained. Following this approach, a sensitivity analysis was performed for the coils shown in Table 1, for altitudes ranging between 20-50m. Figure 5 (a) demonstrates the cumulated sensitivity as a function of distance for a 0.4 kHz signal originating at an altitude of 30 m, while Figure 5 (b,c) show the correlation between depth and footprint size for a 0.4 kHz signal and a 1.8 kHz signal. Based on the sensitivity analysis as well as performance concerns, we decide to use an overlap of 150 m. While this is less than the footprint size for the real part of the 0.4 kHz signal, more than 75 % of the sensitivity for a 100 Ωm halfspace is retained, and if considering total sensitivity over all frequencies then the total loss of sensitivity is around 7 %, which we deem an acceptable loss. Based on Figure 4 (b) we use a section size of 750 m for forward calculations, and 550 m for derivative calculations.

1D/2D Hybrid inversion scheme

The conceptual idea behind the hybrid inversion scheme is to use computationally inexpensive approximate forward and derivative computations in the first inversion steps where accuracy is of little importance. As the iterations start to converge, one can then gradually switch to higher accuracy computations that are more expensive. Within such a scheme, the overall computational time can be greatly reduced without sacrificing the quality of the final model. Such a scheme can be constructed in several ways; Christiansen et al. (2015) have created a hybrid scheme using increasingly accurate 1D modelling responses, and a similar approach could be envisioned in 2D by using a coarse mesh in the early iterations and a more refined mesh in the later stages, as is done in Haber et al. (2007a). However, we believe that our hybrid scheme is computationally superior to such a scheme, since 1D modelling is so computationally inexpensive compared to 2D modelling, and the number of full 2D iterations utilized in our scheme is quite low as will be demonstrated later. Our hybrid scheme is a 3-stage scheme with:

1. 1D forward and derivative calculations
2. 2D forward, 1D derivative calculations

3. 2D forward and 2D derivative calculations

Each stage is executed with a fixed number of iterations. By running the hybrid scheme over a large number of synthetic models, we have empirically found that the optimal number of iterations are four in the first stage, and eight in the second stage. The third stage runs until the algorithm converges.

The inversion is said to have converged if the relative misfit change is less than 1% between two iteration steps. If convergence is reached in stage 1 or 2, then the inversion is advanced to the next stage and the process continues.

RESULTS AND DISCUSSION

Synthetic model

The 2D hybrid inversion algorithm is demonstrated on two synthetic models. A system resembling the RESOLVE system with the parameters shown in Table 1 is modelled. In both examples, the inversion is started from a 100 Ωm halfspace with a model discretization of 20 layers where the thickness of each layer increases logarithmically from 3 m – 10 m. Horizontal smoothing constraints are employed with a covariance factor of 1.6 and vertical smoothing with a factor of 3.0. There are 51 equidistant soundings distributed over a 500 m long line. All data have a uniform 5 % uncertainty.

Figure 6 shows the results of an inversion of a conductive lens. Figure 6 (a) Illustrates the true model, which consists of a 50 Ωm lens in a 500 Ωm halfspace, Figure 6 (b) shows a 1D inversion, Figure 6 (c) shows a hybrid inversion, and Figure 6 (d) shows a full 2D inversion. The 1D inversion mostly manages to recover the conductive lens at the correct depth, but strong pant legs are produced. The 1D inversion is shown with both a 1D and 2D residual curve. Both residual curves use the model arrived at through the 1D inversion, but the 1D residual evaluates the 1D forward responses, while the 2D residual is relative to 2D forward responses. Variation between the two residuals can therefore be regarded as an indicator of areas where 1D modeling is insufficient. Both the hybrid and the full 2D inversion reproduce the lens as good as can be expected from an AEM measurement, without any pant legs effect and with a good determination of the lens boundaries, and a misfit well below 1, which in our synthetic model without noise is a good thing. The speedup gained by utilizing the hybrid inversion was 2.7x compared to the 2D inversion, and will be discussed in detail in the Performance subsection.

Figure 7 shows the results of an inversion of a sharp horizontal conductivity contrast. Figure 7 (a) Illustrates the true model, where the left side is 10 Ωm and the right side is 200 Ωm, Figure 7 (b) shows a 1D inversion, Figure 7 (c) shows a hybrid inversion, and Figure 7 (d) shows a full 2D inversion. In this case, the 1D inversion creates a rather wide region around the conductivity contrast where the conductivities are smeared and there are clearly visible pant legs. Once again, both 1D and 2D residuals are shown. The full 2D inversion demonstrates a better determination of the vertical boundary and while smearing is still observed, the affected region is significantly smaller. The hybrid model again converges to a model, which is significantly better than the 1D model, as noticed both by the size of the smearing region as well as the residual, which is only slightly higher than for the 2D inversions. The differences between the 2D and the hybrid model are likely a result of the 1D model doing a poor job of accurately modelling the sharp conductivity contrast, combined with model equivalences, as evidenced the similarity of the hybrid/2D residual. The speedup gained by utilizing the hybrid inversion was 6.6x compared to the 2D inversion, and will be discussed in detail in the Performance subsection.

Field example

As a final test, the 2D algorithm is used on a field example collected by a RESOLVE system owned by the German Bundesanstalt für Geowissenschaften und Rohstoffe (BGR). The field data was collected on a small island named Langeoog, where the target is a mapping of the freshwater/saltwater boundary (Siemon et al. 2015). Langeoog comprises three geological features: the base is formed from glaciofluvial sediments, from the Pleistocene age. These sediments contain Lauenburg clay, which lies at a depth of 15-35 m below sea level, with a typical thickness of a few meters. Overlaying the Pleistocene layer is a Holocene marine deposit consisting of primarily silt, which lies at 10-20 m depth below sea level. The top-layer consist of dunes and beach sand. For more information about the geology of Langeoog, see Costabel et al. (2017).

The field data profile is 1400 m long and consists of 144 soundings. Inversion results are shown in Figure 8. Figure 9 (a) Shows a 1D inversion, Figure 9 (b) shows a hybrid inversion, while Figure 9 (c) shows a 2D inversion. Overall, the three different inversions show very consistent results, though there are notable differences in the top layers of the soundings at a distance of ~1 km. While the models deviate in this area, the data residual for the 2D and hybrid code are only negligibly lower than for the 1D inversion. Upon a closer look at the fit of each individual transmitter frequency, it is revealed that there is excellent correspondence between measured data and modelled response for all coils except coil 3. Coil #3 is off by several standard deviations in the high residual area at a distance of 1 km. Figure 9 shows an example of this for sounding 112, which is marked in Figure 8 by a vertical red line. The speedup gained by utilizing the hybrid inversion was 6x compared to the 2D inversion, and will be discussed in detail in the Performance subsection.

Parallelization and scalability

Since the introduction of commodity multicore CPUs in 2005, parallelization has become increasingly important. While computational speed continues to grow exponentially, it has become a non-trivial issue to fully harness this power. Algorithms often have to be specifically tailored to enable optimal parallelization, and with the shift away from uniform memory access (UMA) systems, and towards non-uniform memory access (NUMA) systems architecture, this becomes an even harder problem. The architectures are illustrated in Figure 10. The consequences of having a NUMA system is that data placement becomes paramount. If data is not placed in the local memory associated with the processor working on it, it will need to be accessed over the interconnect by the processor. Not only does this add significant latency, but the interconnect also has limited bandwidth and becomes saturated much before the direct channels to local memory. For this reason, good scaling on the NUMA system is harder to achieve in general than on uniform memory access (UMA) systems, and if not done carefully can actually lead to decreased performance, unlike for UMA systems (Dong et al. 2010).

The 2D FEM problem can be parallelized in several ways, but to get the best possible scalability for large surveys, we chose to put our parallelization across the sections. In other words, multiple sections are computed in parallel. This requires more memory than putting the parallelization over the wavenumbers, however with the sectioning used, the memory requirement during the 2D modelling is less than 1 GB per thread utilized (not shown), and thus the total memory requirement is inconsequential on modern hardware. While parallelization over the sections require more memory than other approaches it also gives the best scalability for large surveys, because there is practically no inter-communication between the different threads. Section parallelization provides good largescale scaling, but it does not provide much benefit for small-scale problems. In order to remedy this, an additional parallelization over the frequencies of each section was implemented using OpenMP's collapse directive. By parallelizing over both sectioning and frequency, good scaling can be achieved for surveys of all sizes. Figure 11 shows the parallel scalability of the code. It can be seen that the scaling is almost linear for low numbers of threads, whereas linear scalability is lost for higher numbers of threads due to memory bandwidth limitations.

Another key concept, when doing parallelization is affinity. That is how the parallel threads are bound to the various cores in the system. If thread affinity is not employed it can severely affect performance, especially on NUMA systems. Without affinity, the calculations of a thread are never confined to a single core, but rather executed in small portions executed on random cores of the system. This can have dramatic consequences since memory locality cannot be assured and data kept in cache is constantly lost. Figure 11 demonstrates two different affinity schemes, which are commonly employed: compact affinity and scattered affinity. When using compact affinity, thread spawning tends to cluster together on a NUMA node until all processors on the NUMA are engaged, whereas scattered affinity tends to spread out the thread spawning across all NUMA nodes. The two different affinity modes can have a significantly different performance depending on the problem to which they are employed. Because of the low level of intercommunication between the parallel threads, a practically identical scaling for the two affinities can be seen in Figure 11.

As a final comment to scaling, it should be mentioned that due to the way we do sectioning and inversion, our code has a linear scaling in compute time as a function of survey size (not shown here).

Performance

Our 2D code is capable of inverting surveys of virtually any size, due to the scalability introduced by sectioning. Thus far, the code has been successfully tested on a 100-line km survey with 10000 soundings. For such a survey, the code performs a full 2D forward and derivative calculation in ~5 hours on a NUMA system with two Intel Xeon E5-2650 v3 CPUs, each with 10 cores.

The total inversion time for the 100-line km survey can be seen in Table 2. Both the hybrid and 2D inversion reach comparable misfits, but the hybrid scheme does so 2.3 times faster than the 2D. Though the performance numbers presented are representative, it should be mentioned that the number of iterations needed to reach convergence can vary quite heavily between different surveys. Obviously, this also makes inversion times vary quite heavily. Roughly speaking a pure 2D inversion can usually be done in around 10-20 iterations, while a hybrid inversion requires 14-20 iterations. Note that the number of iterations and hence inversion time, depends heavily on the stopping criteria, which we have chosen to be a relative misfit change of less than 1%.

In the examples shown previously, the speedup gained from utilizing the hybrid scheme was 2.7x for the synthetic conductive lens, 6.6x for the horizontal conductivity contrast, 6x for the small field example, and 2.3x for a 100-line km. These are all very significant speedups, and they are generally generated without notably worsening the resulting model. The reason why the speedups vary largely depends on the number of iterations spent in stage 3 of the hybrid scheme, if only a few iterations are spent then the speedup is high in general, whereas if more than a few iterations are spent in stage 3, the speedup will generally be in the low end.

CONCLUSION

We have presented an algorithm for hybrid 2D frequency domain forward modelling and inversions. The 2D forward and derivative calculations are done on a triangular finite element mesh using sectioning, while inversions are done on a regular grid. The finite element mesh is created with the inversion grid as the foundation, which makes interpolation between the meshes significantly easier. By using sectioning and a regular grid for inversion, the code is able to handle large-scale inversions, which are otherwise often problematic for higher dimensional inversion codes. We have demonstrated how section sizes should be chosen to optimize computational times, and shown how forward and derivative calculations are optimally performed using different section sizes. Our parallelization goal was to achieve maximum speed; hence, the code is parallelized over both frequencies and sections. This gives the code high efficiency for both large and small surveys, as well as excellent scaling properties even on non-uniform memory architectures. Though focus was on computational speed, the memory consumption is less than 1GB per thread, and thus memory consumption for this algorithm was deemed inconsequential. Furthermore, we presented a hybrid 1D/2D scheme, which boosts the computational speed of 2D inversions by $\sim 2 - 6x$, without significantly reducing the accuracy. The concept of combining lower- and higher-dimensional algorithms in a hybrid scheme to significantly increase computational speed, is a largely unused optimization within the scientific community. We have demonstrated our algorithm with two successful synthetic examples and a field example.

ACKNOWLEDGMENTS

This study was supported by Innovation fund Denmark, through the AirTech4Water project. We furthermore thank Dr. Bernhard Siemon and BGR for the field data.

APPENDIX A

In order to recast the governing equations 15-16 into a system of linear equations, the finite element method is employed. In this regard, we substitute $\widetilde{E}_{sy}$ and $\widetilde{H}_{sy}$ by interpolated fields combined with second order shape functions:

$$\widetilde{E}_{sy}(x,z) = \sum_{i=1}^{n} N_i(x,z)\widetilde{E}_{sy,i}, \qquad \widetilde{H}_{sy}(x,z) = \sum_{i=1}^{n} N_i(x,z)\widetilde{H}_{sy,i}\ ,$$

where n is the number of nodes attached to one element. Replacing $\widetilde{E}_{sy}$ and $\widetilde{H}_{sy}$ in this way leads to an approximation of the Maxwell equations, which bear a residual. We use the weighted residual procedure to minimize the residual averaged over the area of each grid cell. As our procedure applies to both equation 15 and 16 in the same way, we will only focus on equation 15. The weighting function for the residual is the same as the interpolation function, i.e. $N_i(x,z)$, and integration of both sides of the equation and applying the rule for integration by parts combined with Gauss' Theorem leads us to:

$$-i\omega\varepsilon_y \int_\Omega N^T N d\Omega\, \widetilde{E}_{sy} + \frac{i\omega\varepsilon_x}{c_{zx}}\left[\oint_\Gamma N^T \frac{\partial N}{\partial x} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial x} d\Omega\right] \widetilde{E}_{sy} + i\omega\frac{\varepsilon_z}{c_{xz}}\left[\oint_\Gamma N^T \frac{\partial N}{\partial z} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial z} d\Omega\right] \widetilde{E}_{sy} - \frac{ik_y}{c_{zx}}\left[\oint_\Gamma N^T \frac{\partial N}{\partial x} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial z} d\Omega\right] \widetilde{H}_{sy} + \frac{ik_y}{c_{xz}}\left[\oint_\Gamma N^T \frac{\partial N}{\partial x} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial x} d\Omega\right] \widetilde{H}_{sy} =$$

$$-(i\omega\varepsilon_{py} - i\omega\varepsilon_y) \int_\Omega N^T N d\Omega\, \widetilde{E}_{py} + \left(\frac{i\omega\varepsilon_{px}}{c_{pzx}} - \frac{i\omega\varepsilon_x}{c_{zx}}\right)\left[\oint_\Gamma N^T \frac{\partial N}{\partial x} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial x} d\Omega\right] \widetilde{E}_{py} + \left(\frac{i\omega\varepsilon_{pz}}{c_{pxz}} - \frac{i\omega\varepsilon_z}{c_{xz}}\right)\left[\oint_\Gamma N^T \frac{\partial N}{\partial z} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial z} d\Omega\right] \widetilde{E}_{py} - \left(\frac{ik_y}{c_{pzx}} - \frac{ik_y}{c_{zx}}\right)\left[\oint_\Gamma N^T \frac{\partial N}{\partial x} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial z} d\Omega\right] \widetilde{H}_{py} + \left(\frac{ik_y}{c_{pxz}} - \frac{ik_y}{c_{xz}}\right)\left[\oint_\Gamma N^T \frac{\partial N}{\partial x} d\Gamma + \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial x} d\Omega\right] \widetilde{H}_{py}\ ,$$

where Ω is the area of the respective element and Γ the element's boundary. Within the model domain the integrals of connected elements cancel each other out and at the model domain boundary, we assume Dirichlet boundary conditions, i.e. $\widetilde{E}_{sy} = \widetilde{H}_{sy} = 0$, hence the boundary integrals can be ignored. This finally results in the following system of linear equations:

$$-i\omega\varepsilon_y \int_\Omega N^T N d\Omega\, \tilde{E}_{sy} + \frac{i\omega\varepsilon_x}{c_{zx}} \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial x} d\Omega\ \tilde{E}_{sy} + i\omega\varepsilon_z \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial z} d\Omega\ \tilde{E}_{sy} - \frac{ik_y}{c_{zx}} \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial z} d\Omega\ \tilde{H}_{sy} + \frac{ik_y}{c_{xz}} \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial x} d\Omega\, \tilde{H}_{sy} = -\left(i\omega\varepsilon_{py} - i\omega\varepsilon_y\right) \int_\Omega N^T N d\Omega\, \tilde{E}_{py} + \left(\frac{i\omega\varepsilon_{px}}{c_{pzx}} - \frac{i\omega\varepsilon_x}{c_{zx}}\right) \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial x} d\Omega\ \tilde{E}_{py} + \left(\frac{i\omega\varepsilon_{pz}}{c_{pxz}} - \frac{i\omega\varepsilon_z}{c_{xz}}\right) \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial z} d\Omega\ \tilde{E}_{py} - \left(\frac{ik_y}{c_{pzx}} - \frac{ik_y}{c_{zx}}\right) \int_\Omega \frac{\partial N^T}{\partial x}\frac{\partial N}{\partial z} d\Omega\ \tilde{H}_{py} + \left(\frac{ik_y}{c_{pxz}} - \frac{ik_y}{c_{xz}}\right) \int_\Omega \frac{\partial N^T}{\partial z}\frac{\partial N}{\partial x} d\Omega\, \tilde{H}_{py}.$$

By creating and combining this system of equations and its counterpart resulting from equation 16, the desired linear system of equations is found:

$$\boldsymbol{A}\tilde{\boldsymbol{x}} = \boldsymbol{b},$$

where $\boldsymbol{A}$ is the global symmetric stiffness matrix, $\tilde{\boldsymbol{x}}$ contains the Fourier transformed EM-fields, and $\boldsymbol{b}$ contains the source terms.

REFERENCES

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

Auken, E. & A.V. Christiansen. 2004. Layered and laterally constrained 2D inversion of resistivity data. *Geophysics* **69**, 752-761.

Auken, E., A.V. Christiansen, C. Kirkegaard, G. Fiandaca, C. Schamper, A.A. Behroozmand, A. Binley, E. Nielsen, F. Effersø, N.B. Christensen & others. 2014. An overview of a highly versatile forward and stable inverse algorithm for airborne, ground-based and borehole electromagnetic and electric data. *Exploration Geophysics* **46**, 223-235.

Beamish, D. 2003. Airborne EM footprints. *Geophysical Prospecting* **51**, 49-60.

Caudillo-Mata, L.A., E. Haber, L.J. Heagy & C. Schwarzbach. 2016. A Framework for the Upscaling of the Electrical Conductivity in the Quasi-static Maxwell's Equations. *Journal of Computational and Applied Mathematics* **317**, 388-402.

Christiansen, A.V. & E. Auken. 2004. Optimizing a layered and laterally constrained 2D inversion of resistivity data using Broyden's update and 1D derivatives. *Journal of Applied Geophysics* **56**, 247-261.

Christiansen, A.V., E. Auken, C. Kirkegaard, C. Schamper & G. Vignoli. 2015. An efficient hybrid scheme for fast and accurate inversion of airborne transient electromagnetic data. *Exploration Geophysics* **47**, 323-330.

Costabel, S., B. Siemon, G. Houben & T. Günther. 2017. Geophysical investigation of a freshwater lens on the island of Langeoog, Germany–Insights from combined HEM, TEM and MRS data. *Journal of Applied Geophysics* **136**, 231-245.

Cox, L.H., G.A. Wilson & M.S. Zhdanov. 2010. 3D inversion of airborne electromagnetic data using a moving footprint. *Exploration Geophysics* **41**, 250-259.

Cox, L.H., G.A. Wilson & M.S. Zhdanov. 2012. 3D inversion of airborne electromagnetic data. *Geophysics* **77**, WB59-WB69.

Cuthill, E. & J. McKee. 1969. Reducing the bandwidth of sparse symmetric matrices. *Proceedings of the 1969 24th national conference*, 157-172.

Dong, X., G. Cooperman & J. Apostolakis. Year. Multithreaded Geant4: Semi-automatic Transformation into Scalable Thread-Parallel Software, Berlin, Heidelberg, Euro-Par 2010 - Parallel Processing, 287-303.

Doyle, B., K. Kivi & S.S. BH. 1999. The TU Kwi Cho (D027 and D018) Diamondiferous Kimberlite Complex, Northwest Territories, Canada.

Haber, E., S. Heldmann & U. Ascher. 2007a. Adaptive finite volume method for distributed non-smooth parameter identification. *Inverse Problems* **23**, 1659-1676.

Haber, E., D.W. Oldenburg & R. Shekhtman. 2007b. Inversion of time domain three-dimensional electromagnetic data. *Geophysical Journal International* **171**, 550-564.

Hestenes, M.R. & E. Stiefel. 1952. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* **49**, 409-436.

Jahandari, H., S. Ansari & C.G. Farquharson. 2017. Comparison between staggered grid finite–volume and edge–based finite–element modelling of geophysical electromagnetic data on unstructured grids. *Journal of Applied Geophysics* **138**, 185-197.

Key, K. & J. Ovall. 2011. A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling. *Geophysical Journal International* **186**, 137-154.

Kirkegaard, C. & E. Auken. 2015. A parallel, scalable and memory efficient inversion code for very large-scale airborne electromagnetics surveys. *Geophysical Prospecting* **63**, 495-507.

Li, W.-B., Z.-F. Zeng, J. Li, X. Chen, K. Wang & Z. Xia. 2016. 2.5 D forward modeling and inversion of frequency-domain airborne electromagnetic data. *Applied Geophysics* **13**, 37-47.

Liu, G. & A. Becker. 1990. Two-dimensional mapping of sea-ice keels with airborne electromagnetics. *Geophysics* **55**, 239-248.

Marquardt, D. 1963. An Algorithm for Least Squares Estimation of Nonlinear Parameters. *SIAM, Journal of Applied Mathematics* **11**, 431-441.

Menke, W. 1989. *Geophysical data analysis: discrete inverse theory.* Academic Press, ISBN 0124909213 (alk. paper).

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

Mitsuhata, Y. & T. Uchida. 2002. 2-D inversion of frequency-domain EM data caused by a 3-D source. *Methods in Geochemistry and Geophysics* **35**, 153-171.

Reid, J.E., A. Pfaffling & J. Vrbancich. 2006. Airborne electromagnetic footprints in 1D earths. *Geophysics* **71**, G63-G72.

Siemon, B., S. Costabel, W. Voß, U. Meyer, N. Deus, J. Elbracht, T. Günther & H. Wiederhold. 2015. Airborne and ground geophysical mapping of coastal clays in Eastern Friesland, Germany. *Geophysics* **80**, WB21-WB34.

Stoyer, C. & R.J. Greenfield. 1976. Numerical solutions of the response of a two-dimensional earth to an oscillating magnetic dipole source. *Geophysics* **41**, 519-530.

Streich, R., M. Becken & O. Ritter. 2011. 2.5D controlled-source EM modeling with general 3D source geometries. *Geophysics* **76**, F387-F393.

Saad, Y. 1994. ILUT: A dual threshold incomplete LU factorization. *Numerical linear algebra with applications* **1**, 387-402.

Saad, Y. 2003. *Iterative methods for sparse linear systems.* Siam, ISBN 0898715342.

Vöge, M. 2010. ngi25em Version 4.0 User Documentation.

Vöge, M., A. Pfaffhuber, E. Auken, C. Kirkegaard, T. Boesen, S. Hendricks & P. Hunkeler. Year. 2.5 D Inversion of Sea Ice Thickness from Helicopter EM Data. First European Airborne Electromagnetics Conference, Turin.

Ward, S.H. & G.W. Hohmann. 1988. Electromagnetic theory for geophysical applications. *Electromagnetic methods in applied geophysics* **1**, 131-311.

Wilson, G., A. Raiche & F. Sugeng. 2006. 2.5 D inversion of airborne electromagnetic data. *Exploration Geophysics* **37**, 363-371.

Wilson, G.A., L.H. Cox, M. Čuma & M.S. Zhdanov. 2012. Inverting airborne geophysical data for mega-cell and giga-cell 3D Earth models. *The Leading Edge* **31**, 316-321.

Yang, D. & D.W. Oldenburg. 2012a. Practical 3D inversion of large airborne time domain electromagnetic data sets. *ASEG Extended Abstracts* **2012**, 1-4.

Yang, D. & D.W. Oldenburg. 2012b. Three-dimensional inversion of airborne time-domain electromagnetic data with applications to a porphyry deposit. *Geophysics* **77**, B23-B34.

Yang, D., D.W. Oldenburg & E. Haber. 2014. 3-D inversion of airborne electromagnetic data parallelized and accelerated by local mesh and adaptive soundings. *Geophysical Journal International* **196**, 1492-1507.

Yu, W.W. & E. Haber. Year. A 2.5 D Inversion of Airborne Electromagnetic data. 2012 SEG Annual Meeting.

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

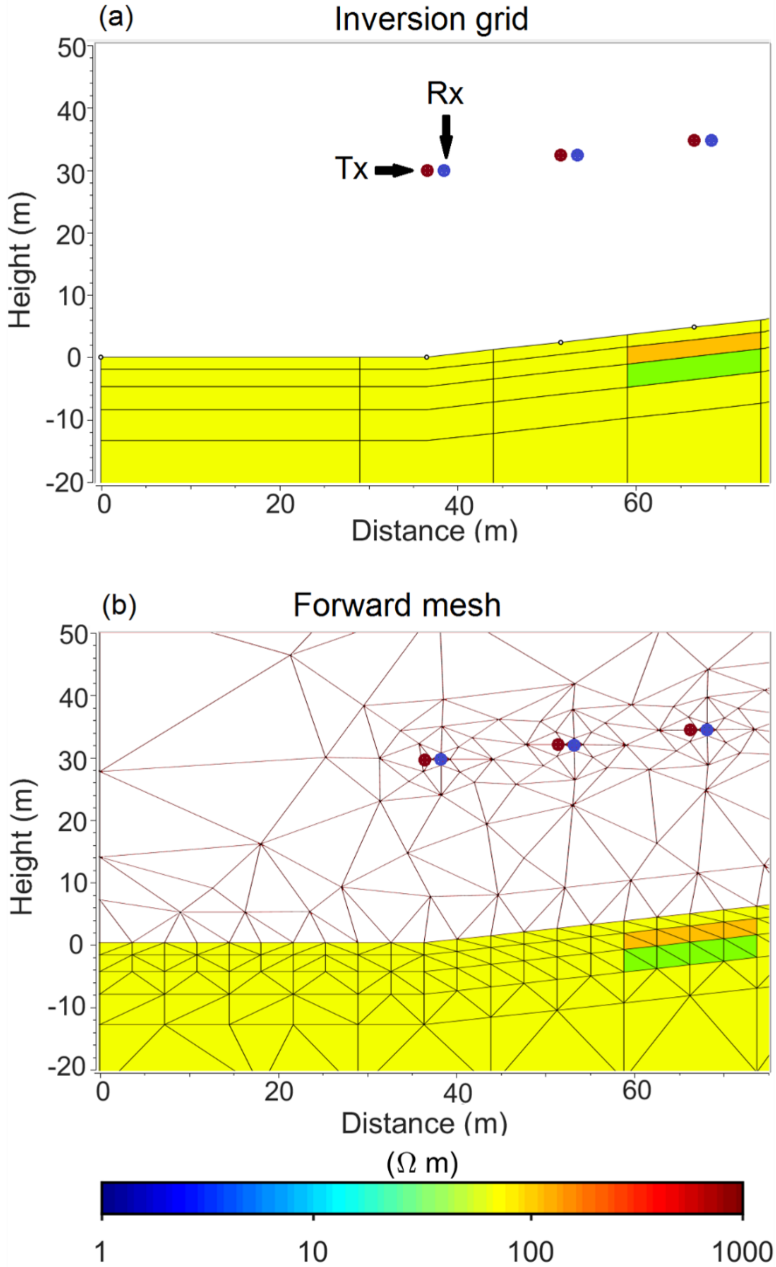


Figure 1	Inversion grid and forward mesh for a small section of a HEM survey. Transmitters are indicated by a brown circle, receivers by a blue circle. (a) Shows the inversion grid. In the inversion grid, the column width is determined by the sounding distance, while the column depth is reflecting the thickness of the model layers. (b) Shows the resulting forward modelling finite element mesh. Note how the inversion grid is still present in the

forward modelling mesh, since this is used as the skeletal structure for building the finite element mesh.

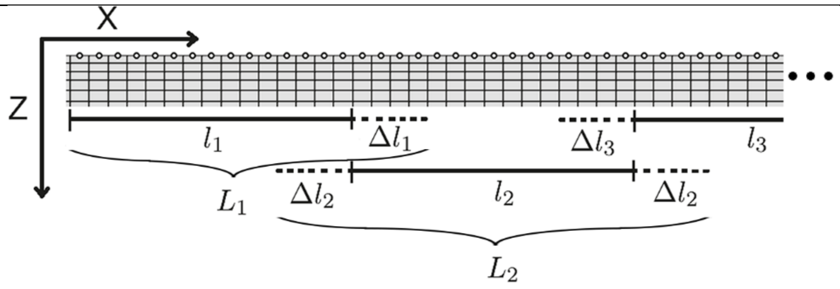


Figure 2 Sectioning of survey lines. Each section, L, consists of a core section, l, and overlap regions, Δl . Sections at the end of a survey line only have one overlap region, while all other sections have two overlap regions. The dots on top of the ground surface indicate the individual soundings.

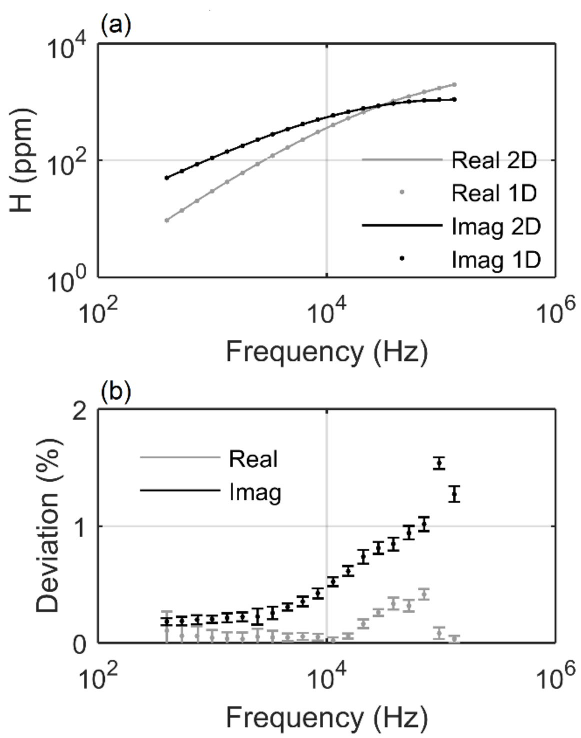


Figure 3	1D and 2D forward responses and deviations, on a 100 Ωm halfspace at an instrument altitude of 30 m, as a function of frequency (a) Shows an example of a 1D and 2D forward response for just a single sounding. (b) Shows the relative deviations between 2D and 1D forward response. The deviation is expressed as a range, because the accuracy of the 2D responses varies between soundings near the edge and near the center due to mesh variations. The 2D responses are from a 300 m section with 30 equally spaced soundings. Deviations at all frequencies and all positions are below 2%.

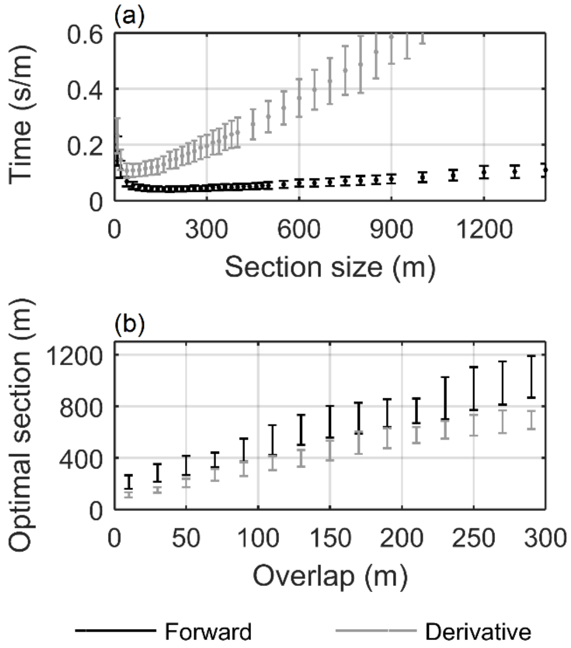
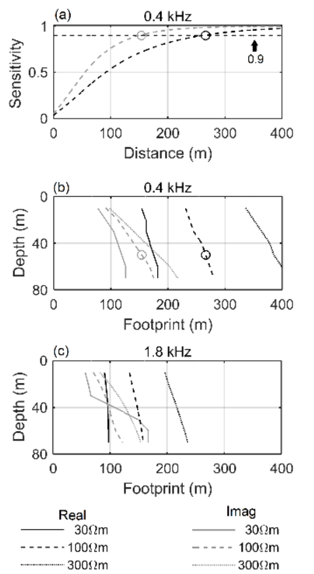


Figure 4	(a) Time required for an iteration of forward and derivative calculations for the RESOLVE system presented in Table 1 using a 10 m sounding separation, as a function of section size. For small section sizes the initialization cost becomes dominant and visible as a sharp increase in iteration time per m, while for large section sizes the quadratic scaling of the finite elements becomes dominant, and is visible as a linear rise in iteration time per meter (it appears linear in the figure, because time is normalized with section size, hence

	time is given in seconds per meter). The range bars indicate variability in computational time between various repetitions, and indicate 1 standard deviation. (b) Shows optimal section sizes as a function of the overlap, where the range bars represent section sizes that are within 5% of the optimal compute time.

	
Figure 5	Sensitivity analysis for a frequency domain system at an altitude of 30 m. (a) Cumulated sensitivity as a function of distance for a 0.4 kHz signal, on a 100 Ωm halfspace at a depth of 50-60 m, with the horizontal dashed line indicating the footprint size at 90% threshold. The two circles indicate the distances where the 0.4 kHz signal reaches the threshold for the real and imaginary response. (b,c) Correlation between footprint size and depth for a 0.4 kHz signal and a 1.8 kHz signal, on a 30 Ωm, 100 Ωm, and 300 Ωm halfspace. On (b) the two circles from (a) are also plotted.

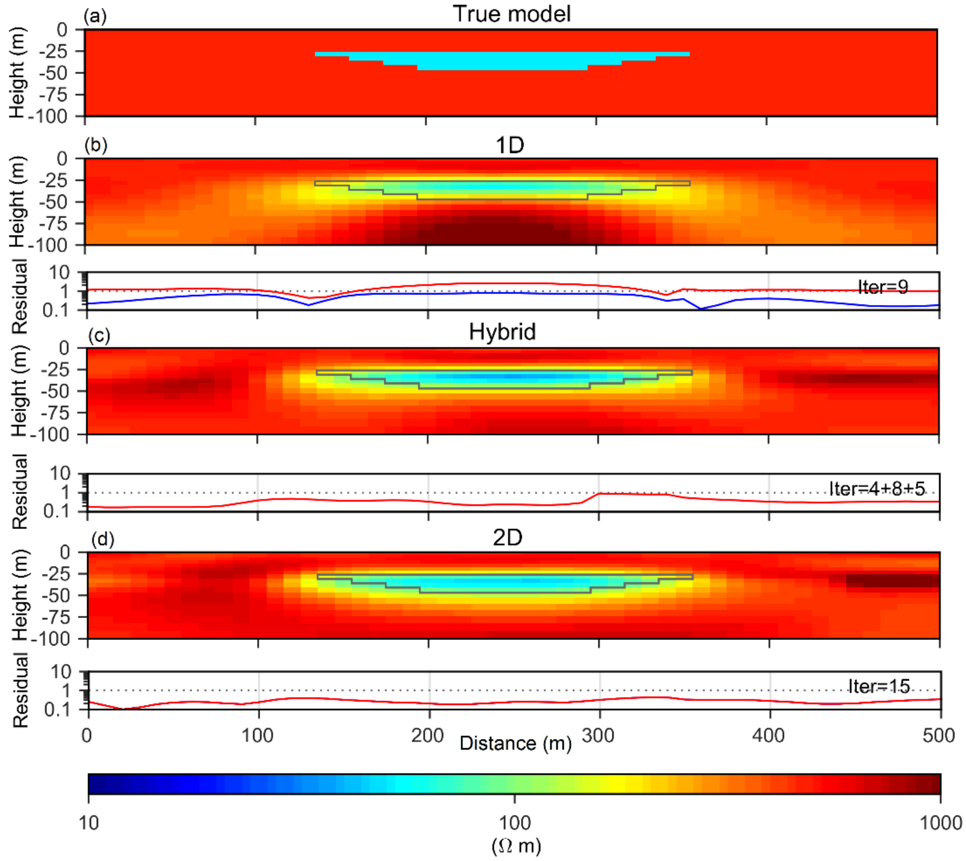


Figure 6	Inversion results for a synthetic conductive lens. (a) True model, which consists of a 50 Ωm lens in a 500 Ωm halfspace. (b) 1D inversion, where the blue line is the 1D data residual, and the red line is the 2D data residual. (c) Hybrid inversion, where the number of iterations are given in each hybrid stage separately. (d) 2D inversion. For all inversions, the number of iterations until convergence is written inside the residual box. Note that the data residuals are normalized by the standard deviation.

636

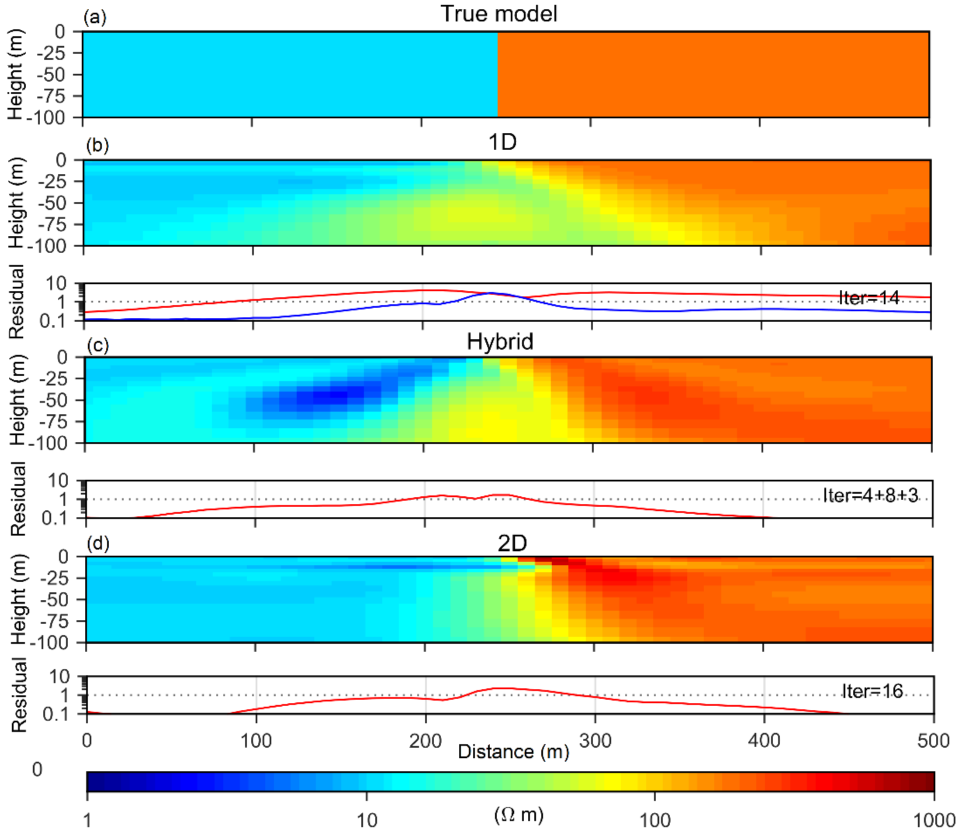
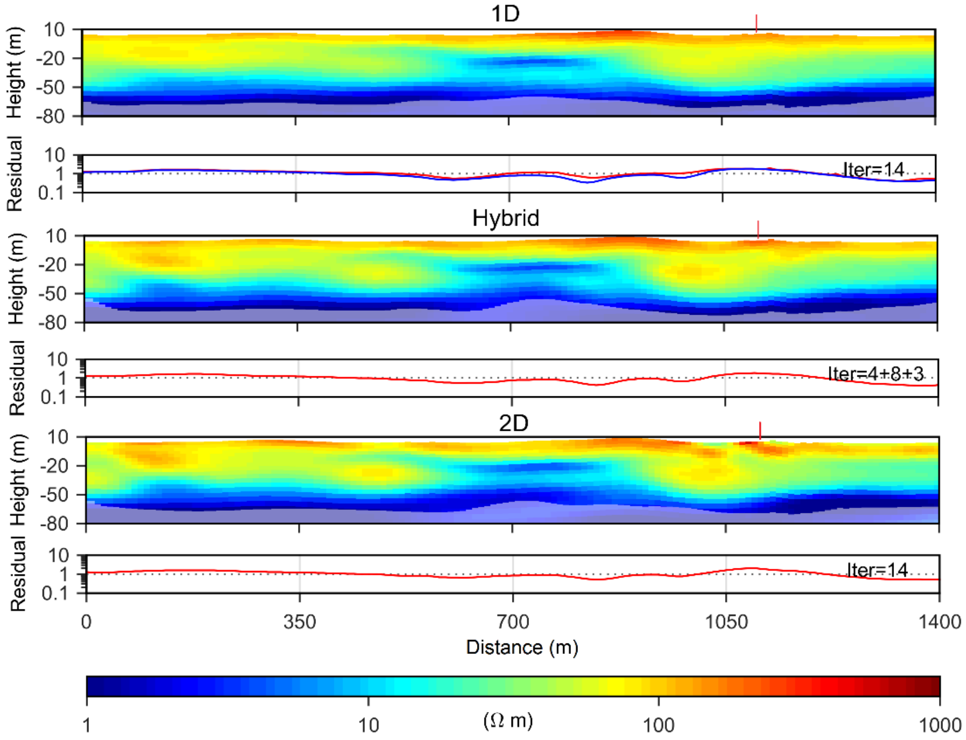
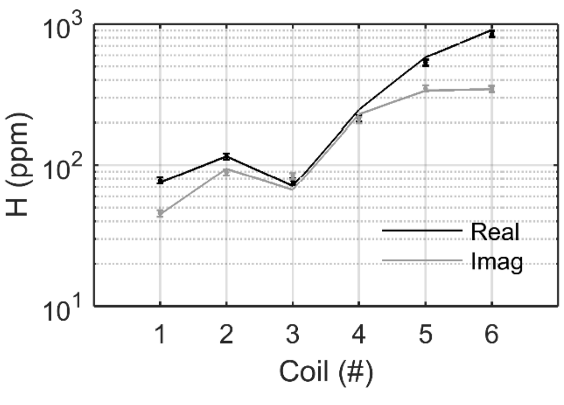


Figure 7 Inversion results for a horizontal conductivity contrast. (a) Illustrates the true model; the left side has a resistivity of 10 Ωm, while the right side has a resistivity of 200 Ωm. (b) Shows the result from a 1D inversion, where the blue line is the 1D data residual, and the red line is the 2D data residual. (c) Shows the result from a hybrid inversion. (d) Shows the result from a 2D inversion. The number of iterations until convergence is written inside the residual box. Note that the data residuals are normalized by the standard deviation.

	
Figure 8	Inversion results from a survey in northern Germany (Siemon et al. 2015). (a) Shows a 1D inversion, where the blue line is the 1D data residual, and the red line is the 2D data residual. (b) Shows a hybrid inversion, where the number of iterations are provided for each separate stage. (c) Shows a 2D inversion. The number of iterations until convergence is written inside the residual box. The vertical red line at the top of the figures around 1.1 km is sounding 112, which is shown in Figure 9. Note that the data residuals are normalized by the standard deviation.

	
Figure 9	Example of a 2D sounding curve for the inversion results shown in Figure 8 (c). The sounding curve is marked in Figure 8 by a vertical red line at the top of the figures, around 1.1 km distance. The coil # configuration can be found in Table 1.

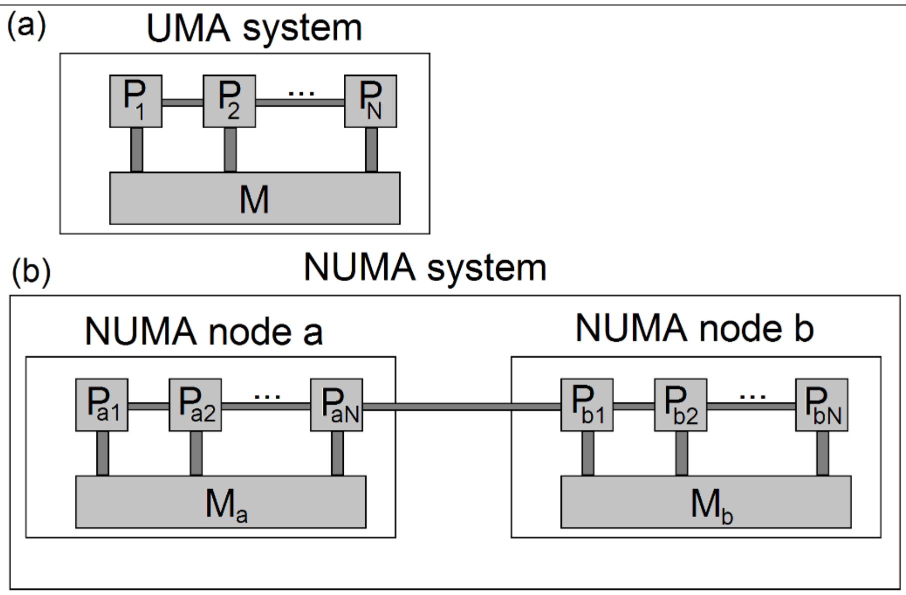


Figure 10 UMA and NUMA system architectures. P represents physical cores, while M represents memory. (a) an UMA system; here there is only one memory bank, and all the cores have the same bandwidth connection to the memory. (b) a two-node NUMA system, which essentially consists of two connected UMA systems. Thus, each NUMA node has a number of cores attached to a memory bank, and the different memory banks are then connected together by a relatively small bandwidth.

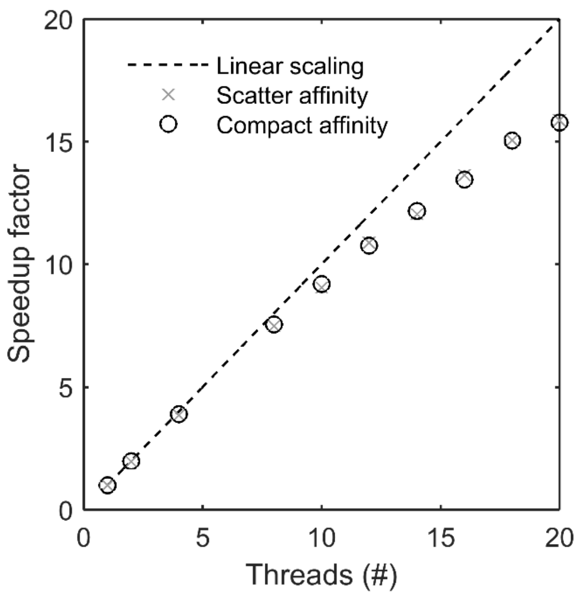


Figure 11	Parallel scaling of the code. The data are generated on a NUMA system with two Intel Xeon E5-2650 v3 CPUs, each with 10 cores. The threads are bound to specific logical processors, following either a compact affinity approach or a scattering affinity approach.

Coil #	Orientation	Frequency (kHz)	Separation (m)
1	Z	0.395	7.9
2	Z	1.822	7.9
3	X	5.4	9.06
4	Z	8.199	7.9
5	Z	38.76	7.9
6	Z	128.76	7.9

Table 1 Acquisition parameters mimicking a RESOLVE system used in our computational cost analysis simulation. The flight altitude is 30 m and the uncertainty on the data is 5%.

	100 line km	
	Iterations	Runtime (hours)
1D	15	0.1 h
Hybrid	4+8+5	36 s+4.9 h+26.0 h = 31 h
2D	15	71 h

Table 2 Runtime and iteration number for an inversion of a 100-line km survey conducted with the RESOLVE system. For the hybrid system, the iteration numbers and runtimes are given for each of the 3 stages.