GEOPHYSICS, VOL. 81, NO. 2 (MARCH-APRIL 2016); P. E69–E87, 17 FIGS., 3 TABLES.
10.1190/GEO2015-0217.1



# Survey decomposition: A scalable framework for 3D controlled-source electromagnetic inversion

Dikun Yang[1] and Douglas W. Oldenburg[1]

## ABSTRACT

Numerical modeling and inversion of electromagnetic (EM) data is a computationally intensive task. To achieve efficiency, we have developed algorithms that were constructed from a smallest practical computational unit. This "atomic" building block, which yields the solution of Maxwell's equations for a single time or frequency datum due to an infinitesimal current or magnetic dipole, is a self-contained EM problem that can be solved independently and inexpensively on a single core of CPU. Any EM data set can be composed from these units through assembling or superposition. This approach takes advantage of the rapidly expanding capability of multiprocessor computation. Our decomposition has allowed us to handle the computational complexity that arises because of the physical size of the survey, the large number of transmitters, and the large range of time or frequency in a data set; we did this by modeling every datum separately on customized local meshes and local time-stepping schemes. The counterpart to efficiency with atomic decomposition was that the number of independent subproblems could become very large. We have realized that not all of the data need to be considered at all stages of the inversion. Rather, the data can be significantly downsampled at late times or low frequencies and at the early stages of inversion when only long-wavelength signals are sought. We have therefore developed a random data subsampling approach, in conjunction with cross-validation, that selects data in accordance to the spatial scales of the EM induction and the degree of regularization. Alternatively, for many EM surveys, the atomic units can be combined into larger subproblems, thus reducing the number of subproblems needed. These trade-offs were explored for airborne and ground large-loop systems with specific survey configurations being considered. Our synthetic and field examples showed that the proposed framework can produce 3D inversion results in uncompromised quality in a more scalable manner.

## INTRODUCTION

Since the 1970s, a variety of basic 3D electromagnetic (EM) modeling and inversion algorithms have been published using different numerical techniques, including the integral equation method (Hohmann, 1975; SanFilipo and Hohmann, 1985; Newman et al., 1986; Wang et al., 1994), the finite-difference method (Wang and Hohmann, 1993; Commer and Newman, 2004), the finite-element method (Pridmore et al., 1981; Badea et al., 2001; Schwarzbach and Haber, 2013), and the finite-volume method (Haber et al., 2000; Haber and Ascher, 2001). Regardless of the specific choices of numerical strategy, eventually an EM forward or inverse problem needs to solve a system of equations at each frequency or time step for each source in the form of

$$\mathbf{Au} = \mathbf{q} \tag{1}$$

or its equivalent. Here, $\mathbf{A}$ denotes the Maxwell matrix bearing the information about the discretization, physical properties, and frequency or time; $\mathbf{u}$ is an unknown vector of EM fields; and the right side $\mathbf{q}$ contains the sources, boundary conditions, and primary fields. The forward solutions are obtained by solving equation 1. The sensitivity can also be computed by solving a system having the same $\mathbf{A}$ but a different $\mathbf{q}$.

The computational costs of a 3D inversion can become challenging when equation 1 is large and/or many such equations need to be solved. For example, expanding the survey area can result in geometric increase of the size of equation 1, a wideband survey requires

Manuscript received by the Editor 8 April 2015; revised manuscript received 22 October 2015; published online 26 February 2016.
[1]University of British Columbia, Geophysical Inversion Facility, Vancouver, Canada. E-mail: yangdikun@gmail.com; doug@eos.ubc.ca.

the solution of the large system at many frequencies or time steps, and a survey using many sources needs to solve the system for each source. In the iterative inversion, the complete forward problem must be solved repeatedly to update the forward-modeled responses and the sensitivity. The difficulties associated with equation 1 have been intensively investigated in many previous publications, and a number of improvements have been proposed. Some notable examples include using an unstructured mesh rather than a rectangular mesh to reduce the number of cells (Schwarzbach and Haber, 2013; Jahandari and Farquharson, 2015), using the optimal rational Krylov subspace projection approach (model reduction) to efficiently model the full spectrum with fewer solves of equation 1 (Börner et al., 2008; Zaslavsky et al., 2011, 2013), and using the concept of multilevel or multiresolution approach to reduce the size of equation 1 (Commer and Newman, 2006a; Yang and Oldenburg, 2012a, 2012b). Recently, direct solvers have become popular because if $\mathbf{A}$ is factorized, many different sources and many time steps (if the step size does not change) can be quickly solved at a cost almost equal to one solving of equation 1 (Grayver et al., 2013; Oldenburg et al., 2013).

All of the previous contributions and faster computers have made a medium-sized 3D inversion on a desktop computer or a cluster possible, but the inversion of large-scale and wideband surveys is still very challenging. The computational difficulty lies in the poor scalability of equation 1: The computing time and the usage of memory grow geometrically as the survey expands (Yang et al., 2014). Attempts to decompose the large system and make use of parallel processing have taken different routes.

The most straightforward approach of parallelization is to use domain decomposition and solve the subdomain problems in parallel (Alumbaugh et al., 1996; Newman and Alumbaugh, 1997; Xie et al., 2000; Commer and Newman, 2004; Commer et al., 2008; Holtham and Oldenburg, 2012). Unfortunately, a brute-force decomposition of the spatial domain usually results in strongly coupled subdomains. The dependency between subproblems requires significant intercommunication that deteriorates the scalability more quickly when massive parallelization is implemented.

Another approach is to separate the inversion and forward modeling meshes, and makes the forward mesh "locally" optimized. The computation then enjoys improved scalability because of the independency between local problems. Commer and Newman (2006b) propose the decoupling of the simulation mesh and the model mesh and anticipate the potential of further adaptation to a small subset of the parameter space. Later on, they use multiple simulation meshes optimized to the frequency and source-receiver orientation for marine controlled-source EM (CSEM) survey (Commer and Newman, 2008). Wang et al. (2009) invert borehole EM data by multiple meshes with moving refinements at different intervals along the well and modeling the regions outside of the focused sections by coarse background cells. Key and Ovall (2011) propose that a triangular mesh can be adaptively refined for a subset of the EM data and the parallelization over those subsets could achieve much better efficiency. Cox et al. (2010, 2012) use the concept of a moving "footprint" in the integral equation formulation to invert airborne EM data, so that each sounding used only a small portion of the entire 3D mesh. Because the local problems are more independent and they use the same Green's function, this footprint approach has shown much better scalability than has domain decomposition. In Yang et al. (2014), we implement our local mesh method with finite-volume discretization for airborne EM using nonconformal local and global meshes to achieve better adaptiveness; we also formally address the difficulty of having too many local problems by using a random subsampling algorithm. Haber and Schwarzbach (2014) further develop the approach by more efficient octree meshes.

The idea of improving computational efficiency by multiple data-specific local meshes has been suggested for nearly a decade. When using this idea in practice, one usually faces a dilemma: If the local meshes are made too specific to small data subsets, there will be an unmanageable number of subproblems; if more data are grouped together for mesh sharing, the efficiency of an individual local mesh can be compromised (see the discussions in Key and Ovall, 2011; Haber and Schwarzbach, 2014). Therefore, most of the above works choose to find a trade-off by grouping the data in terms of different parameters (source, receiver, frequency, etc.) according to the specific problems. With the success in airborne EM, we think the local mesh approach can be further pursued by modeling one datum in a single subproblem in conjunction with the adaptive subsampling. This combination enables highly efficient subproblems and a significantly reduced number of subproblems to model. Another challenge that the existing local mesh methods have not addressed occurs when the transmitter size is physically large; then, a local mesh may need to be as large as the original global mesh, thus making the idea of mesh decoupling less appealing.

In this paper, we establish a generic inversion framework, referred to as survey decomposition (SD), which can be applied to all types of CSEM surveys. SD inversion has two main features: (1) It allows "atomic" decomposition of the data that represents an arbitrary CSEM survey with infinitesimal point sources and receivers and (2) it does random, dynamic, and adaptive subsampling of the data. The first feature seeks the smallest possible subproblem, and the second feature minimizes the number of subproblems needed in the inversion. As an overall summary, we first analyze the computational complexity for the EM inversion problem, and this drives our SD using atomic building blocks. We also discuss the methodology of decomposition and present its general formulation in forward and inverse problems.

Our motivating survey is the ground large-loop survey commonly used in mineral exploration. Such a survey usually deploys a transmitter loop that is arbitrarily shaped, a few kilometers on the side, and has data extending over a broad time range, say, from $10^{-4}$ to $10^{1}$ s. Existing local mesh techniques are challenged because the modeling domain that includes the extended transmitter and a receiver is very big. Our SD framework overcomes this difficulty by decomposing the CSEM transmitter into a combination of electric or magnetic dipoles. Each local mesh is then concerned with a single datum at a particular frequency or time and a dipole transmitter. This can greatly reduce the computing time of a subproblem.

We note that SD is a high-level computing strategy compatible with any spatial discretization, time integration, and optimization method. Because the comparison of specific EM solving techniques is out of the scope of this paper, we assume that a set of solving techniques is given, and we concentrate on how the efficiency is improved when using SD with the same set of techniques. In particular, we adopt the forward and inverse algorithms described by Oldenburg et al. (2013), who use the finite-volume method on a rectangular mesh, implicit (backward-Euler) time stepping, direct solver for the forward problem, and a Gauss-Newton (GN) method for the inversion. These implementations may not necessarily be

optimal in all circumstances, but they serve as a demonstration for how the equations and workflows developed here can be applied in general. The user may decide to implement an alternate numerical technique. For simplicity of presentation, we use time-domain EM (TEM) as examples, but inversion of frequency-domain EM data proceeds in the same way.

We also note that this paper presents the basic theory of SD along with examples that validate the correctness of the framework. Although the examples are implemented with small-scale parallelization on single server, we design SD with massive parallelization in mind. As readers will see in the following sections, SD produces a large number of independent subproblems, each of which makes a cumulative contribution to the whole problem. This enables SD to be robustly scaled up in a large computing environment that is becoming available on either in-house clusters or cloud-based servers. However, we will not explicitly discuss the structure of those computing devices here.

The paper proceeds as follows: We first look globally at the computational complexities for the EM problem. SD, along with definition of atomic building blocks, is presented followed by decomposition of transmitters into looplets and linelets and carrying out forward modeling. The necessary background for our inversion procedure, as well as the introduction of our adaptive subsampling and cross-validation technique, is then presented. Synthetic and field inversion examples for large ground loop data are presented, and the paper concludes with a discussion about implementation and practical considerations.

## COMPUTATIONAL COMPLEXITIES

The multiscale nature of the EM inverse problem gives rise to computational complexity in three ways: space complexity, time complexity, and optimization complexity.

Space complexity refers to the large number of cells required to model the EM data because of the geographic size of a survey and the contrast in the scale length of the cells needed for early time and late time data. Conventionally, a universal mesh is designed to contain cells that are fine everywhere to accurately model all the transmitters and receivers. However, from the viewpoint of a particular transmitter or receiver, overcomputing (computing at an unnecessarily fine level) may occur on fine cells that are far away from that transmitter or receiver. Similarly, from the viewpoint of datum at a particular time, there is also overcomputing when using the universal mesh that accommodates the spatial scales corresponding to the earliest and the latest times. If the data at different locations and times are modeled separately and each subproblem is concerned with a narrower band of space and time, then the local meshes can be customized to minimize the overcomputing.

The solution to Maxwell's equations is a time-dependent problem, which can be solved by either directly stepping in time or sweeping the spectrum after the frequency-domain transform. The former requires equation 1 to be solved at many time steps and the latter at many discrete frequencies. Some explicit stepping methods do not solve the linear system like equation 1, but the computations are equivalent. In this paper, we will mostly assume the implicit stepping methods or unconditionally stable explicit stepping (Du Fort and Frankel, 1953; Wang and Hohmann, 1993) so the step length does not affect stability. Either way, the time (or frequency) complexity is proportional to the number of frequencies or time steps required. In the frequency domain, optimal frequencies can be found by using a Krylov subspace approach (van den Eshof and Hochbruck, 2006; Zaslavsky et al., 2011). These frequencies can also lead to an optimal time-stepping scheme in the time domain. When straightforwardly stepping in the time domain, overcomputing occurs if the time steps are unnecessarily small, so adaptive step lengths have been proposed to avoid too many small steps at late times (Newman and Commer, 2005; Commer and Newman, 2006a; Haber et al., 2007; Um et al., 2010; Oldenburg et al., 2013). However, there is still overcomputing when all times are modeled together using a universal adaptive time discretization. From the viewpoint of a datum at a particular time, the small steps accommodating the earlier times are unnecessarily fine, and the extra steps for the later times are useless. We generally observe more over computing for data with multiple time scales, and therefore the best efficiency can be achieved by modeling different delay times separately using customized time steps that is adjusted to the proper time scale in each of the subproblems.

Optimization complexity is caused by repeated forward solutions to simulate each datum. If the EM data are modeled separately using local meshes and local time discretizations, then the optimization complexity is proportional to the number of data (subproblems) modeled. There exist two kinds of scale contrast in the optimization: (1) The early time data need to be sampled closely in space to handle the variation of the near-surface conductivity, whereas the late-time data can have a coarse spatial sampling because they are sensitive to larger volumes. (2) A regularized inversion looks for large-scale features in the model at the beginning of the inversion, and then it gradually adds more small-scale structure as the trade-off parameter decreases. Overcomputing happens if the early and late times are sampled at the same spacing and every datum is indiscriminately computed to high accuracy throughout the inversion. If the survey can be modeled as many separate data, we can use judicious subsampling to achieve efficiency.

Our examinations above show that the conventional design of "one size fits all" could give rise to significant overcomputing because such a design attempts to resolve multiple scales in a single system. To reduce the scale contrasts, the new framework promotes separate modeling of data, so that the algorithms for each subproblem are designed with the concept of "adapt to the scale." When used in the extreme, this idea leads to the independent modeling of a single datum and the introduction of an atomic building block.

## SURVEY DECOMPOSITION

### Atomic building block

We define an atomic building block to be the solution of Maxwell's equations due to a point source for a point receiver that measures a datum at one time or at one frequency. The point source or receiver can be a short line segment or a small loop in practice, and it is treated as an electric dipole or a magnetic dipole in numerical modeling. An atomic building block, characterized by its source location (S), receiver location (R), and time (T), is a self-contained EM problem with a small Maxwell's system equation 1 that can be quickly solved on a single core. Such atomic building blocks can be used to construct any type of CSEM survey, so if the small Maxwell's systems in the atomic building blocks are solved, then the solution to the original problem is also obtained immediately.

The efficiency of the atomic building block comes from the customized discretizations specifically designed for the data it models.

Figure 1 shows a source and receiver pair in an atomic building block and its local mesh. It is also possible to design the local mesh based on the nonregular grid, but we concentrate on the rectangular mesh here because it offers great simplicity. Although Figure 1 provides a general idea of a local mesh's appearance, the specific design can vary. Irrespective of details, a local mesh in our framework can be characterized by the following features:

1) Geometric coarsening of the cells: Because of the diffusive nature of the EM fields, a local mesh needs to finely sample the fields around the source but it can gradually become coarser farther away. The sensitivity in the inverse problem also has short-wavelength variation near the receiver, so the mesh must be locally refined at the source and the receiver. How fast the cells can be coarsened depends on the conductivity contrast and the distance between the contrast and the receiver. Slower expansion of cell size, which leads to more cells, may be needed if the contrast is sufficiently large and close to the receiver. In our practice, an empirically determined cell expansion rate of 1.2–1.5 is a good balance between accuracy and efficiency. More sophisticated cell coarsening schemes can be developed (e.g., Asvadurov et al., 2007).
2) Adaptive cell sizes: The smallest cell size around the source and receiver adapts to the scale of investigation for the specific atomic problem. Because the scale of investigation varies with the conductivity model and the time or frequency, the smallest cell size needed is sought by starting with larger (and hence fewer) cells and automatically reducing its size until a desired accuracy is achieved. If the cell expansion rate is over estimated, the adaptive cell size can still find a local mesh appropriate for the modeling, but it will have more cells than necessary.
3) Adaptive domain size: The distance between the source/receiver and the outer boundaries also depends on the model and the time or frequency. This parameter is sought using the same adaptive approach used for finding the smallest cell size. The procedure begins with an underestimated distance (and hence fewer cells) to the outer boundaries. The adaptive mesh design takes some iterations to find a good cell size and domain size, and each iteration costs one forward modeling. The extra computing for the iterations only occurs at the beginning of local mesh design, and it is not needed when computing the sensitivity on the same mesh.

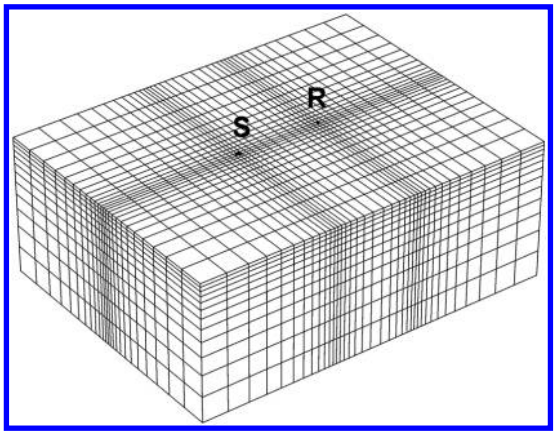


Figure 1. An atomic problem with one dipole source (S), one dipole receiver (R), and its local mesh.

4) Local coordinate system: A smaller Maxwell system can sometimes be achieved by having a local mesh use a different co-ordinate system than that of the global problem.

A local mesh can have a small domain size with small cells around the source and receiver and increasingly larger cells outside. Overall, it has far fewer cells than the global mesh, and this avoids the solution of a large Maxwell's system. The rate of increasing cell size, or grid coarsening, on the local mesh will be dependent upon the medium. The new framework's adaptiveness to the scale of investigation, depending on the time or frequency and updated when the conductivity changes, further eliminates the overcomputing in the local problem without losing accuracy. Figure 2 illustrates the difference between the global mesh and one of the local meshes after decomposition of some typical surveys. An extended source is treated as the superposition of many small dipole sources ($\mathbf{S}_i$), each of which, bonded with the receiver, is an atomic building block solved on a local mesh (Figure 2c and 2d).

In an analogous manner, a local time discretization for the time stepping scheme can also be designed for a particular measurement time in an atomic problem. An illustrative example is presented in Figure 3, in which the goal is to model two times: the early time $t_1$ and the late time $t_2$. Different stepping schemes exist. For example, the step length ($\delta t$) can increase gradually from the initial time to $t_2$, adapting to the time scale of EM diffusion. Changing $\delta t$ will result in the change of the Maxwell matrix $\mathbf{A}$ in equation 1, so this scheme needs to compute $\mathbf{A}^{-1}$ at every time step; this is an expensive operation for iterative and direct solvers. If a constant $\delta t$ is used for stepping from the initial time to $t_2$, one only needs to compute $\mathbf{A}^{-1}$ with a direct solver at the beginning and then carry out inexpensive forward and back substitution at every step. Unfortunately, this scheme usually requires a prohibitively large number of steps. As a trade-off between changing $\delta t$ at every step and not changing $\delta t$ at all, Oldenburg et al. (2013) show that a fast method of time

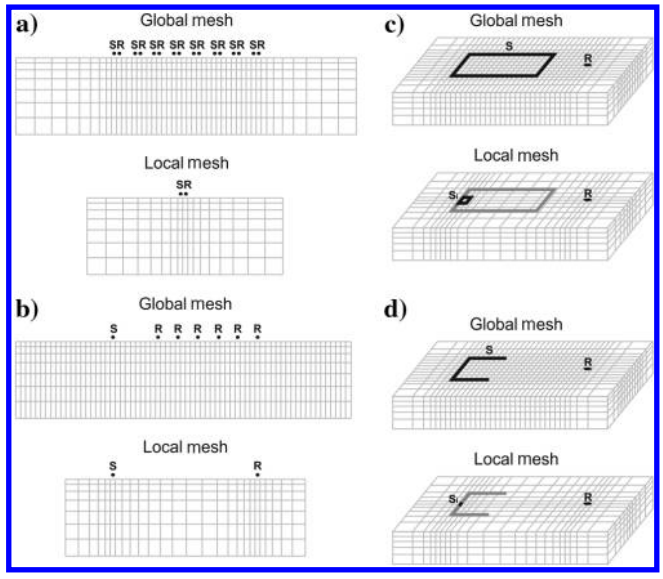


Figure 2. Modeling of atomic problems on local meshes: (a) airborne EM, (b) marine CSEM, (c) ground large-loop EM, and (d) ground EM with long wire source. S indicates the source, and R indicates the receiver.

stepping is to change $\delta t$ a few times. This scheme first chooses a few different $\delta t$ corresponding to the different time scales from the earliest to the latest times; then, the associated $\mathbf{A}$ matrices are factorized in parallel; finally, the time stepping takes place sequentially using the stored factors. When a certain factor is in use, a constant $\delta t$ will be kept for some steps. For illustrative purposes, we assume in this example that $\delta t$ is chosen as one eighth of the desired delay time. If data at time $t_1$ and $t_2$ are modeled with a universal global time discretization that contains two different step lengths $\delta t_1$ and $\delta t_2$ following Oldenburg et al. (2013), at least 15 time steps are required to complete the forward modeling. Alternatively, upon decomposition, $t_1$ and $t_2$ can be modeled independently using a local time discretization appropriate for each. The factorization time is still the same, but the stepping time can be reduced to eight time steps if parallelized. The saving in time can be further exaggerated if more times over a broader range are modeled.

The time-stepping scheme in Oldenburg et al. (2013) already achieved the same degree of time efficiency as the frequency-domain approaches, because computing $\mathbf{A}^{-1}$ for a few different $\delta t$ is equivalent to computing $\mathbf{A}^{-1}$ at a few different frequencies and the sequential stepping with forward and back substitution is equivalent to transforming the solutions from frequency domain to time domain. Using local time discretization does not reduce the total amount of computation, but it makes the sequential time stepping parallelizable, and thus it can be much faster.

We also note other benefits using local time discretizations:

1) Less computation for explicit sensitivity calculation: In many cases, explicitly forming the sensitivity matrix is a better choice than working with implicit sensitivities (see the discussion in Yang et al., 2014). The sensitivity matrix is often computed row by row with back-propagation time stepping, so computing the sensitivity for a particular datum requires a complete stepping from the latest to the earliest time. With a universal global time stepping scheme, the total number of steps for the sensitivity is the number of data times the number of steps in the global time discretization. When using a local time discretization, the total number of steps is the number of data times the number of steps in the local time discretization.
2) Memory efficiency: If memory is limited, one usually has other options. For example, a memory-efficient iterative solver can be used instead of a direct solver that produces dense matrix factors. If using a direct solver, the matrix factorization can be computed on the fly, so at any moment, the memory only stores one factor; or one just uses one constant step length throughout the entire stepping (not change $\delta t$ at all). Those methods are more memory efficient than is the algorithm described in Oldenburg et al. (2013) that stores multiple matrix factors at the same time. However, they are not time efficient because their computing time can be unacceptably long. Using local time discretizations achieves the same time efficiency as the stepping in Oldenburg et al. (2013). It is also able to maintain high memory efficiency by sequentially solving the local time stepping problems.
3) Better accuracy: A global time discretization, as discussed above, has to trade off between the number of different step lengths and the total number of time steps. This can lead to a large step that generates more numerical errors immediately after the change of step length. A local discretization is likely to have better accuracy because every time has a constant step length specifically tuned to it.

### Generating subproblems

The atomic building block described above has high efficiency in theory; however, if every datum is modeled separately, the number of atomic problems can be massive and unmanageable. To circumvent this roadblock, we generally need to combine some atomic problems to form a larger subproblem. This will enlarge the scale contrast, and hence the size, of Maxwell's system, but as long as the system can still be quickly solved on a single core, such groupings are efficient.

An example of an efficient grouping is the airborne moving-footprint method that groups atomic blocks sharing the same source and receiver measuring different times into one subproblem. We do the same grouping in Yang et al. (2014) for the airborne data, in which every subproblem is an airborne sounding. Such a grouping requires a global time discretization and a larger number of cells because the local mesh has to accommodate the early and late times. This decision is deemed acceptable because the time span for airborne EM data is usually small, and thus the over computing of not separating the times is tolerable. The potential for grouping multiple soundings, as well as times, is explored by Haber and Schwarzbach (2014). The advantages, or disadvantages, are problem dependent.

Although the airborne EM data lead to simple subproblems, the large-loop surveys, commonly used in deep mining exploration, are challenging because (1) the large loop transmitter cannot be approximated by a single dipole. Thus, every transmitter-receiver pair must be modeled as superposition of many atomic blocks. (2) Ground EM is usually wideband and possibly measures from 0.1 ms up to 1 s; this is a scale contrast that can make the separate modeling of data in time necessary. These complications require that atomic decomposition be used, in which each subproblem is an atomic problem that models an infinitesimally small part of the large loop source and solves for data at a particular receiver and at a particular time. In the remainder of this paper, we focus upon this more complicated situation.

## FORWARD MODELING

### Forward solutions on local mesh

Under the framework of SD, the information about the data and model must be managed at, and exchanged between, the global and local levels. In the basic procedure for forward modeling: (1) a main processor broadcasts the global model, data, and survey information to the local subproblems on an array of workers; (2) the subproblems are solved at the local level in parallel; and (3) the workers return the results, usually a few scalars or a vector, to the main processor.

The conductivity models, as well as the model updates generated during an inversion, are all presented on a predefined global mesh

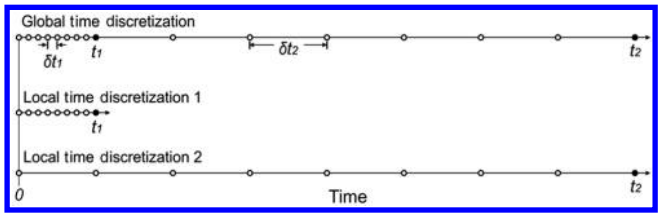


Figure 3. An illustration of time stepping using one global scheme or separate local schemes.

and then transferred to the local meshes on which the subproblems are solved. Equation 1 is never solved on a global mesh, but we assume that the global mesh is properly designed to handle the smallest scale of EM induction. Converting models from the global mesh to local meshes mostly involves representing some small cells by a bulky cell, a procedure often referred to as upscaling or homogenization. The opposite procedure, called downscaling, is usually not needed if the cells in the global mesh are sufficiently small. Upscaling can be nontrivial, especially when the two meshes are not conformal. We acknowledge that this topic deserves sophisticated treatment, but for now, we choose a straightforward formula inspired by the material averaging in Moskow et al. (1999), which finds the mutual intersection between the global and the local cells and uses them as the weights in averaging. This mesh conversion can be represented by a sparse matrix $\mathbf{R}$, in which $r_{ij}$ is the intersecting volume of the $i$th cell in the local mesh and the $j$th cell in the global mesh. Every subproblem has its own $\mathbf{R}$ matrix reflecting the volumetric relation between the local mesh and the global mesh. The local model derived for the $i$th local mesh is expressed as

$$\mathbf{m}_{li} = \mathbf{V}_{li}^{-1}\mathbf{R}_i\mathbf{m}_g, \tag{2}$$

where $\mathbf{m}_l$ and $\mathbf{m}_g$ are the local and global model vectors, respectively. The $\mathbf{V}_l$ is a diagonal matrix of local cell volumes and its inverse normalizes each row of $\mathbf{R}$ by the volume of corresponding cell in the local mesh. The subscript $i$ is the index of subproblem. Matrix $\mathbf{R}$ can be computed using either the rigorous computational geometry, or the fast approximate methods such as the point-cloud algorithm documented in Appendix A. When severe topography is present, additional manipulation to $\mathbf{R}$ may be required to avoid averaging with the extremely low conductivity in the air. We note that matrix $\mathbf{R}$ can be more rapidly formed if the grid of a local mesh is conformal to the global grid or the global cells are nested in the local cells, such as in Cox et al. (2010) and Haber and Schwarzbach (2014). Those make the mesh conversion very straightforward. Here, however, because SD may need a rotation of a local grid to achieve the minimum meshing, the material averaging in equation 2 is generally inevitable. In most cases, we consider that the arbitrary mesh conversion is worthwhile because solving equation 2 is usually more economical than solving Maxwell's equation on a local mesh that has unnecessary cells due to the restriction of grid conforming.

After computations on a local mesh, the modeled three-component data are transformed back to the global coordinate system through

$$\begin{pmatrix} d_{xgi} \\ d_{ygi} \\ d_{zgi} \end{pmatrix} = \mathbf{M}_i^{-1} \begin{pmatrix} d_{xli} \\ d_{yli} \\ d_{zli} \end{pmatrix}, \tag{3}$$

where the subscript $g$ and $l$ indicate the "global" and "local" co-ordinate systems, respectively; the subscript $i$ means the $i$th subproblem; and $\mathbf{M}$ is the 3D rotation matrix mapping a vector from the global to the local coordinate system of the $i$th subproblem.

There is one more linear operator, denoted by a sparse matrix $\mathbf{S}$, that is often needed. The value $\mathbf{S}$ has the dimensions of the number of data by the number of subproblem solutions, and it carries information about how the modeling results from the subproblems are summed to render the desired data. If there is no decomposition of the source or receiver, $\mathbf{S}$ can be ignored or treated as the identity; otherwise, $\mathbf{S}$ reflects the superposition of the decomposed sources or receivers. The specific structure of $\mathbf{S}$ for a ground large loop is discussed later. The final data vector is expressed as

$$\mathbf{d}_g = \mathbf{S}\begin{pmatrix} \vdots \\ \mathbf{M}_i^{-1}\mathbf{d}_{li} \\ \vdots \end{pmatrix}, \tag{4}$$

where

$$\mathbf{d}_{li} = \begin{pmatrix} d_{xli} \\ d_{yli} \\ d_{zli} \end{pmatrix}. \tag{5}$$

The simulated data from the $i$th subproblem in equation 5 can be any kind of EM field. If more than one time is modeled in the $i$th subproblem, $\mathbf{d}_{li}$ can have multiple columns corresponding to multiple times.

## Forward modeling of atomic building blocks

The variable receiver offsets in a ground loop survey require the modeling of atomic blocks with different combination of sources, receivers, and times. We carry out a forward test to evaluate the accuracy of modeling on local meshes that are rotated from the global coordinate system and that have two local refinements. The synthetic model is a 0.1 S/m conductive sphere buried in a 0.001 S/m uniform half-space (Figure 4). The receivers move along two lines, and the source is a vertical magnetic dipole placed very close to the target on the surface. This layout provides different rotation angles and offsets. The forward modeling algorithm is the same finite-volume, implicit time stepping with the direct solver multifrontal massively parallel sparse direct solver (MUMPS) as used in Oldenburg et al. (2013).

Four times, $10^{-4}$, $10^{-3}$, $10^{-2}$, and $10^{-1}$ s, at every receiver location are modeled in this test. Each atomic problem has its own time discretization requiring 15 steps from $t = 0$ to its modeled time and its own adaptively designed local mesh based on the location of the source and receiver and also on time. The initial smallest cell size and domain size of a local mesh are inferred using the diffusion length

$$d = \sqrt{\frac{2t}{\sigma\mu}}, \tag{6}$$

which implies the scale of diffusion can be primarily determined by the delay time $t$ and conductivity $\sigma$ if the magnetic permeability $\mu$ is assumed to be the free space value (Nabighian and Macnae, 1991). The domain size then expands at a rate of 1.5; the cell size refines at a rate of 0.6; and the cell expansion rate is a fixed 1.2.

The results at the four times obtained with SD are compared with the results from the global discretization (GD) modeling in Figure 5. A high degree of agreement is observed for most of the data, especially for the two early times, $10^{-4}$ and $10^{-3}$ s. The $H_y$ data of line 1 are null-coupled and are supposed to be zero due to the symmetry. Although the GD and the SD results differ by orders of magnitude, the numbers are still very small compared with other components (the SD is too close to zero to be on the chart). A similar discrepancy also happens to other weakly coupled components,

such as $H_x$ and $H_y$ on line 2. To quantitatively assess the accuracy of SD, we calculate the normalized difference, defined as the absolute difference between GD and SD results normalized by the total-field strength of the GD. The responses at $10^{-2}$ s have the largest normalized difference, approximately 20% at maximum because for that model time channel $10^{-2}$ corresponds to the scale of investigation comparable with the spatial scale of the target sphere. The most anomalous data are usually more sensitive to the meshing.

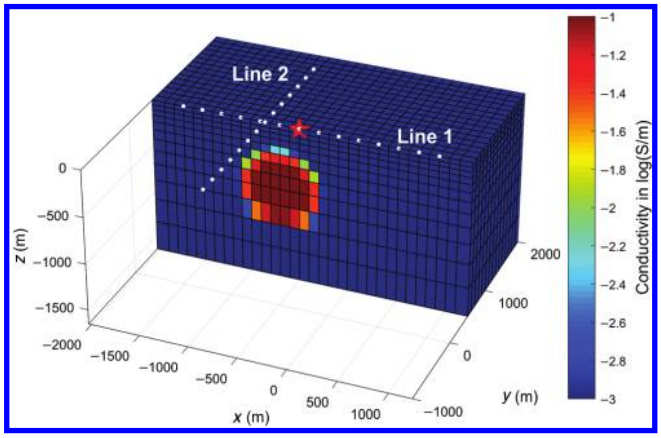


Figure 4. Conductivity model and locations of source and receivers for the atomic modeling test. The magnetic dipole source is marked by the red star and the receivers by the white dots. The padding cells outside of the volume of interest are not shown.

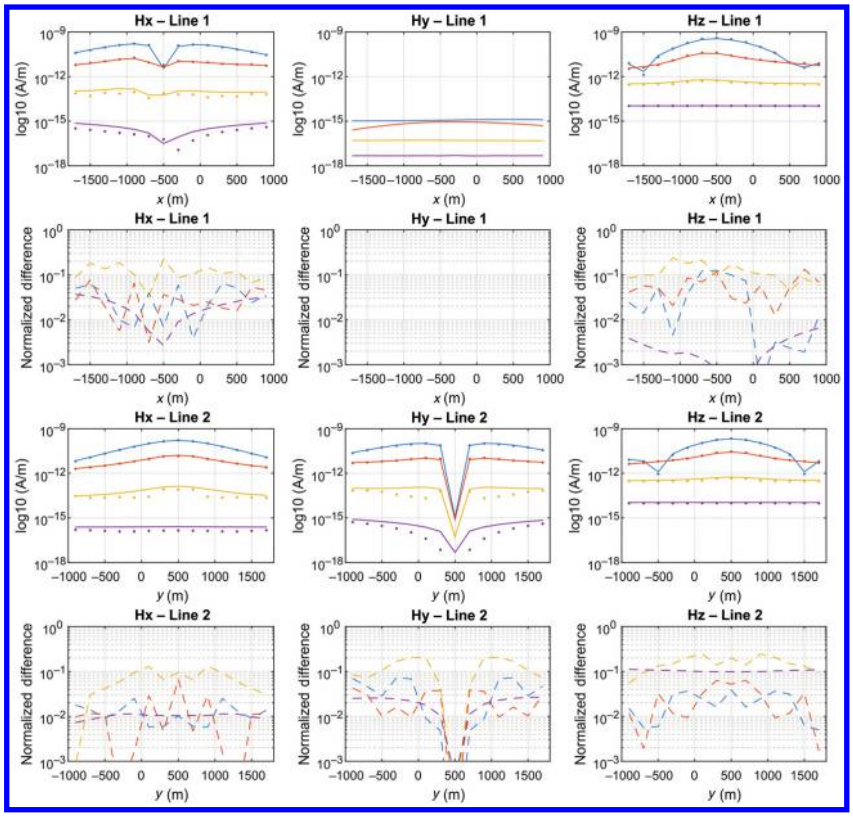


Figure 5. Comparison of the atomic problem modeling between SD (dots) and GD (solid lines). The four time channels are distinguished by colors: $10^{-4}$ (blue), $10^{-3}$ (red), $10^{-2}$ (yellow), and $10^{-1}$ s (purple).

The normalized difference of time channel $10^{-2}$ also presents a declining trend when the receiver is moving further away from the sphere. This implies that coarse cells can be more tolerated if the target is not very close to the receiver. We also notice a strong symmetric pattern on the normalized difference for line 2, which, unlike line 1, requires the rotation of local mesh. The maximum differences are observed when the local meshes are rotated approximately 45° with respect to the global coordinate system, representing the most nonconformal scenario. The accuracy of SD can be improved by using finer cells in the local mesh, either by coarsening the cells more slowly or by initializing the coarsening with smaller cell. There is a trade-off between accuracy and cost.

## Decomposition of large-loop transmitter

The forward solutions due to an extended source are obtained by superposition. A large loop can be represented in two ways: by small line segments (electric dipoles along the wire) or by small current loops (magnetic dipoles filling the loop area). We refer to the former as "linelets" and the latter as "looplets" for the extended sources, and an atomic problem can have either a linelet or a looplet as the source. Our experiments have suggested that the looplet is a better choice over the linelet because a linelet requires a larger modeling domain for its local mesh than does a looplet.

Representing a large loop using the looplets is mathematically equivalent to the numerical integration of a 2D function, which is the EM response measured at a fixed receiver and time and moving the looplet to different locations within the loop area. Given an error tolerance, the integration can be computed with a finite number of looplet locations. Because the number of looplets needed to represent a large loop varies with the transmitter-receiver offset, time, and the conductivity, we do not use a fixed regular grid of looplets. Instead, we develop an adaptive refinement algorithm to find the minimum number of looplet locations, and the associated weights, for a particular receiver and time pair (R-T modeling given the large loop).

Starting from an underestimated number of looplets, the adaptive looplet procedure gradually adds randomly located new looplets to refine the sampling to the 2D function until the result of integration numerically approaches a stable value. Every time a new looplet is added, two operations follow: First, an atomic modeling is carried out using the new looplet as the source. Second, a Voronoi tessellation (Aurenhammer, 1991) takes place based on the updated looplet locations, and the updated areas of the Voronoi cells become the new weights, effectively the looplet dipole moments, for the superposition. The addition of looplets does not require recomputing the atomic modeling for the existing looplets, but because the weights have been changed, the sum of forward responses from all the looplets is also updated. In the limit, as the number of looplets approaches infinity, the sum becomes the true integration. Given a tolerance of model-

ing error, the addition can stop at a reasonably small number of looplets. Figure 6 shows an instance of adaptive looplet refinement in a time-lapse manner; six Voronoi diagrams are generated as the number of looplets $N$ increases from 1 to 6 with the updated weights at every $N$; no additional looplets are needed if the summed responses from $N = 6$ and $N = 5$ are sufficiently close.

With its own collection of looplets, every R-T modeling has a specific summing matrix $\mathbf{S}$ (see equation 4) to combine the results from looplet subproblems together. If the data from a looplet are ordered in $x$-$y$-$z$ components (see equation 5), $\mathbf{S}$ can be assembled using weighted identity matrices. For example, for the $i$th R-T pair that has $n$ looplets, the summing matrix can be

$$\mathbf{S}_i = \begin{pmatrix} w_1 & & & w_2 & & & \cdots & & w_n & \\ & w_1 & & & w_2 & & & w_n & & \\ & & w_1 & & & w_2 & & & & w_n \end{pmatrix}, \quad (7)$$

where $w_n$ is the weight for the $n$th looplet derived from the Voronoi diagram. The final $\mathbf{S}$ for the entire survey is therefore obtained by putting many R-T summation matrices in equation 7 together in a block-diagonal pattern

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_1 & & & \\ & \mathbf{S}_2 & & \\ & & \mathbf{S}_3 & \\ & & & \ddots \end{pmatrix}. \quad (8)$$

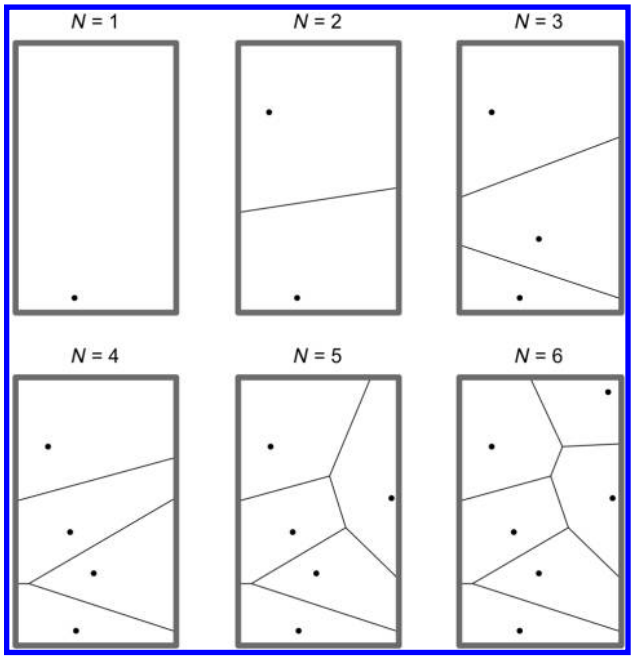


Figure 6. Decomposition of a rectangular transmitter loop using adaptive looplet refinement based on the dynamic Voronoi tessellation. The locations of the looplets are marked by the black dots, and the area of each cell represents the dipole moment of the looplet it contains.

The selection of the looplets is not permanent. Even though the modeling of same R-T is requested at different iterations during the inversion, a new collection of looplets is sought, and the summing matrix is also changed because the updated conductivity model may require different looplets.

## Forward modeling using looplets

To illustrate the ability to model data from a large loop, we carry out a synthetic survey as illustrated in Figure 7 on the same synthetic model in Figure 4. Here, we pay particular attention to the receiver at the cross point of lines 1 and 2 ($x = -900$ and $y = 500$) and four times ($10^{-4}$, $10^{-3}$, $10^{-2}$, and $10^{-1}$ s), corresponding to four R-T modelings.

We first model the desired data using the GD. The global mesh, encompassing the entire survey and numerically validated for all the delay times, has 169,136 ($62 \times 62 \times 44$) cells. A complete forward modeling of this survey for time from $10^{-4}$ to $10^{-1}$ requires four different step lengths and 15 time steps per step length (60 steps in total). The CPU time is 234 s on a six-core computer in parallel mode. The peak memory usage is 11.4 GB. Then, we model the same data using the SD. With a tolerance of 2%, the adaptive looplet algorithm determines that 10, 7, 4, and 3 looplets are, respectively, required for the same R-T problems at $10^{-4}$, $10^{-3}$, $10^{-2}$, and $10^{-1}$ s. The final looplet weights for different times are obtained from the Voronoi cells in Figure 8. It is evident that the late times require fewer looplets to approximate the large transmitter loop than do the early times. The multiscale nature of EM diffusion is thus exploited to save over computing at later times when the scale of investigation is large.

Figure 9 shows the local mesh and model from one of the 10 looplet subproblems at $10^{-4}$ s. This local mesh contains the source and receiver, respectively, marked by the red star and white dot, and it has only 12,000 ($30 \times 20 \times 20$) cells, which is considerably fewer than the number of cells in the global mesh. Because only one time is modeled, only 15 time steps at a constant step length from 0 s to the time of measurement are needed. A forward modeling on this mesh takes 2 s on one core of the same computer, and the memory usage is at the order of 10–100 MB. The forward modeling time of other subproblems ranges from approximately less than 1–5 s depending on the looplet location and the measurement time. The total CPU time required for modeling the entire survey on the same six-core computer was approximately 50 s. This total time includes the time needed for the adaptive mesh design and the job scheduling for load balancing. Once the local meshes are determined, a complete forward modeling only takes approximately 20 s on the six parallel cores when computing the sensitivity. Because the 24 looplet subproblems are randomly distributed to the six cores, on average, each core receives four sequential subproblems. This total time can be further reduced if more cores are available and if the subproblems are scheduled so that all CPUs are continually active. The modeled magnetic field data using the GD and SD formulations are compared in Figure 10. There is good overall agreement especially with components that have large amplitude. It is important to note that adding more receivers (data) to the problem does not increase the computational cost of GD, because GD computes the EM fields everywhere in the global mesh at once. In contrast, the computational cost of SD is proportional to the number of receivers, but this problem can be mitigated by using more cores so that each core still maintains the same load.

## INVERSE PROBLEM

### Gauss-Newton method

We formulate our inversion using Tikhonov regularization and the L-2 norm constraints on the data misfit and model complexity. The objective function in its discrete form is

$$\phi = \frac{1}{2}\|\mathbf{W}_d[F(\mathbf{m}) - \mathbf{d}^{\text{obs}}]\|_2^2 + \beta\frac{1}{2}\|\mathbf{W}_m(\mathbf{m} - \mathbf{m}^{\text{ref}})\|_2^2, \quad (9)$$

where $\mathbf{W}_d$ is the data weighting matrix, $F(\mathbf{m})$ and $\mathbf{d}^{\text{obs}}$ are the vectors of the predicted and observed data, $\beta$ is the regularization parameter, $\mathbf{m}^{\text{ref}}$ is the reference model, and $\mathbf{W}_m$ is the model weighting matrix. The structure of $\mathbf{W}_d$ and $\mathbf{W}_m$ can vary, but we choose the simplest: $\mathbf{W}_d$ is the diagonal matrix of the reciprocal of the data uncertainties; $\mathbf{W}_m$ is an assembly of a diagonal matrix that measures the magnitude of the model and three differential matrices that measure the smoothness of the model in the $x$-, $y$-, and $z$-directions.

A variety of optimization techniques can be used to solve equation 9, but we choose the GN method as an example to demonstrate our framework. Because of the nonlinearity of the problem, the minimizer of equation 9 is sought iteratively by solving

$$[\mathbf{J}(\mathbf{m}^k)^\top\mathbf{W}_d^\top\mathbf{W}_d\mathbf{J}(\mathbf{m}^k) + \beta\mathbf{W}_m^\top\mathbf{W}_m]\delta\mathbf{m}^{k+1} = -\mathbf{g}(\mathbf{m}^k) \quad (10)$$

at the $k+1$th iteration, where $\mathbf{J}(\mathbf{m}^k)$ is the Jacobian matrix (sensitivity) of the model $\mathbf{m}^k$ and $\mathbf{g}(\mathbf{m}^k)$ is the gradient expressed as

$$\mathbf{g}(\mathbf{m}^k) = \mathbf{J}(\mathbf{m}^k)^\top\mathbf{W}_d^\top\mathbf{W}_d[F(\mathbf{m}^k) - \mathbf{d}^{\text{obs}}] + \beta\mathbf{W}_m^\top\mathbf{W}_m(\mathbf{m}^k - \mathbf{m}^{\text{ref}}) \quad (11)$$

at model $\mathbf{m}^k$. The model update may need to be scaled after line search so that $\mathbf{m}^{k+1} = \mathbf{m}^k + \alpha\delta\mathbf{m}^{k+1}$, where $\alpha$ is the scaling parameter.

Because the optimization is an iterative process and the model update $\delta\mathbf{m}^{k+1}$ does not need to be exact, equation 10 is often solved by iterative solvers, such as conjugate gradient (CG) methods (Hestenes and Stiefel, 1952). In a CG solver, the coefficient matrix in equation 10 multiplies a vector $\mathbf{v}$ at every CG iteration. Because $\mathbf{W}_d$ and $\mathbf{W}_m$ are sparse and trivial, the efficiency of the optimization depends on the operation of the dense matrix $\mathbf{J}$ or $\mathbf{J}^\top$ times a vector $\mathbf{v}$ ($\mathbf{Jv}$ or $\mathbf{J}^\top\mathbf{v}$). There are two options: (1) The explicit sensitivity method first computes the elements in $\mathbf{J}$ by carrying out many forward-equivalent computations and storing $\mathbf{J}$ in the memory, and then it does $\mathbf{Jv}$ or $\mathbf{J}^\top\mathbf{v}$ upon request during CG iterations. (2) The implicit method expresses $\mathbf{J}$ as a multiplication of some other matrices, including the forward-modeling operation $\mathbf{A}^{-1}$ in equation 1; when $\mathbf{v}$ is applied to $\mathbf{J}$, forward modeling takes place at every $\mathbf{Jv}$ or $\mathbf{J}^\top\mathbf{v}$ operation. As discussed in Yang et al. (2014), the implicit method is more efficient when a large number of data are modeled together, but we prefer the explicit method here because upon decomposition each subproblem only has a few data to model and $\mathbf{J}$ can be inexpensively stored on local meshes. In practice, the CG solver is often preconditioned to achieve faster convergence, and we use the diagonal of $\mathbf{J}^\top\mathbf{J}$ as the preconditioner. Although we choose the GN method in our inversion, the efficient computing of $F(\mathbf{m})$, $\mathbf{J}$, and $\mathbf{Jv}$ (or $\mathbf{J}^\top\mathbf{v}$) under the framework of SD also allows other optimization methods (e.g., steepest descent, BFGS, and quasi-Newton) to be implemented.

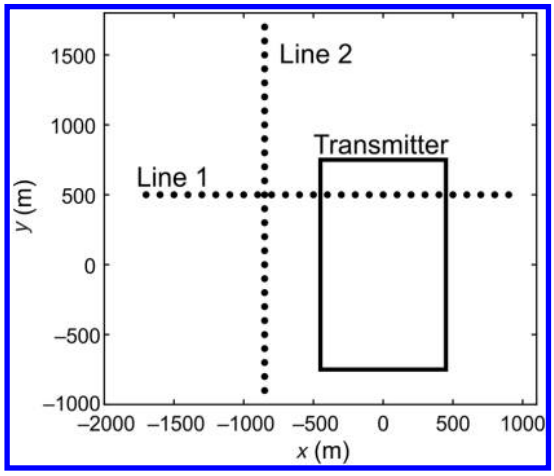


Figure 7. Layout of the synthetic ground large-loop survey.

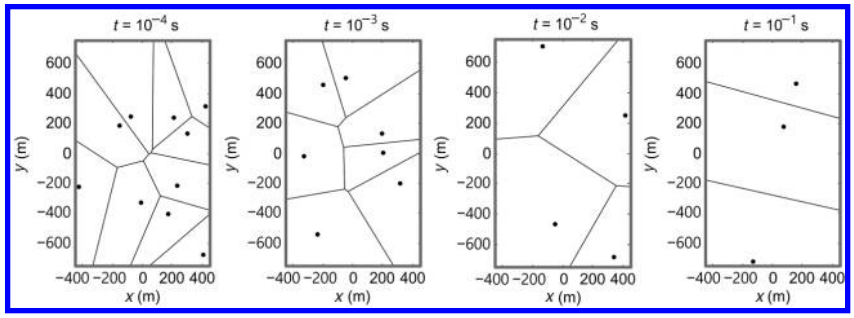


Figure 8. Locations of looplets and the associated Voronoi tessellations of the transmitter loop for the four delay times at the example receiver.

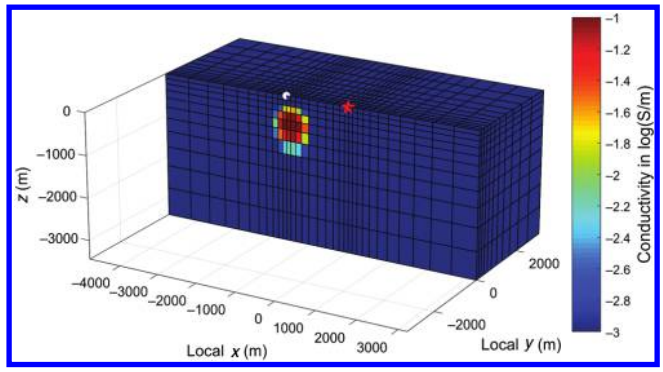


Figure 9. Local mesh and model used in one of the looplet subproblems at $10^{-4}$ s (coordinates in local system with rotation and translation). The looplet source location is marked by the red star, and the receiver is marked by the white dot.

## Computing sensitivity

The computation of sensitivity is very similar to that for forward modeling data. The local discretizations and summing matrix $\mathbf{S}$, designed and formed in the forward modeling, can also be used in the sensitivity. In the explicit method, the computed sensitivity can be stored on the local meshes for the use of sensitivity-vector production, which is also parallelized. Entries of the sensitivity matrix have dimensions of "data per unit model parameter." Column wise, they can be considered as a special kind of data; every column of the sensitivity matrix is a vector pertaining to one model parameter. Row wise, they can be considered as a special kind of model; every row is a vector pertaining to one datum. Therefore, a local sensitivity matrix is transformed to its global form for use in inversion in two steps: First, the sensitivity, as a model on the local mesh, is mapped to the global mesh using the same cell-intersecting matrix $\mathbf{R}$ in equation 2 needed for the material averaging, but here the operation is additive. The sensitivity matrix of the data in the $i$th subproblem on the global mesh in the local coordinate system is

$$\mathbf{J}_{li} = \begin{pmatrix} \cdots \mathbf{j}_{xli} \cdots \\ \cdots \mathbf{j}_{yli} \cdots \\ \cdots \mathbf{j}_{xli} \cdots \end{pmatrix} \mathbf{V}_{li}^{-1} \mathbf{R}_i, \tag{12}$$

where the matrices $\mathbf{V}_{li}^{-1}$ and $\mathbf{R}_i$ together make the sensitivity available on the global mesh. If the local mesh is smaller than the global mesh, the global cells not intersecting with any local cells are assigned zero sensitivity. Second, the sensitivities, just as the data were, must be rotated back to the global system. Similar to equation 4, the sensitivity matrix in the global coordinate system and on the global mesh is

$$\mathbf{J}_g = \mathbf{S} \begin{pmatrix} \vdots \\ \mathbf{M}_i^{-1} \mathbf{J}_{li} \\ \vdots \end{pmatrix}. \tag{13}$$

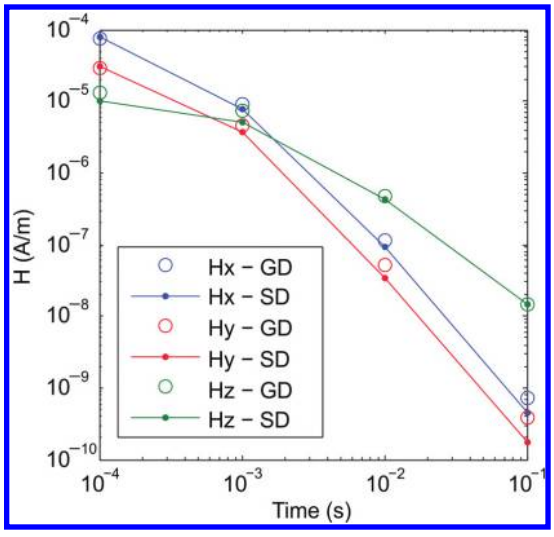


Figure 10. Forward-modeled data using GD and SD with adaptive looplets (SD).

The full sensitivity $\mathbf{J}_g$ is rarely formed in practice. For the explicit sensitivity method and direct solver in the SD, the sensitivity matrices along with all other local auxiliary matrices are stored locally. When a $\mathbf{J}_g\mathbf{v}$ or $\mathbf{J}_g^\top\mathbf{v}$ operation is requested, the operation can be carried out concurrently within the subproblems.

To implement $\mathbf{J}_g$ times $\mathbf{v}$, the vector $\mathbf{v}$ is broadcast to all of the subproblems and applied to equation 12 in parallel; then, every local process returns a short vector, equivalent to $\mathbf{M}_i^{-1}\mathbf{J}_{li}\mathbf{v}$, to the global process; the global process assembles those short vectors into a long vector; and finally, this long vector is applied to $\mathbf{S}$ for the final result. This procedure can be expressed as

$$\mathbf{J}_g\mathbf{v} = \mathbf{S}\mathbf{J}_l\mathbf{v} = \mathbf{S} \begin{pmatrix} \vdots \\ \mathbf{M}_i^{-1}\mathbf{J}_{li}\mathbf{v} \\ \vdots \end{pmatrix}. \tag{14}$$

The value $\mathbf{J}_g^\top\mathbf{v}$ requires the transpose of a dense matrix $\mathbf{J}_{li}$, so it can be slow. This is overcome by first computing $\mathbf{v}^\top\mathbf{J}_g$ and then taking the transpose. The term $\mathbf{v}^\top$ first left multiplies $\mathbf{S}$, yielding another row vector with its length equal to the number of data. This vector is then divided into many short row vectors based on data ordering and segmentation, and the short row vectors are sent to the corresponding subproblems. Within each subproblem, the short row vector left multiplies $\mathbf{M}_i^{-1}$ and $\mathbf{J}_{li}$, returning another row vector to the global process. The global process sums up those row vectors for the final result of $\mathbf{v}^\top\mathbf{J}_g$. This procedure can be expressed as

$$\mathbf{v}^\top\mathbf{J}_g = \sum_i \mathbf{v}^\top\mathbf{S} \begin{pmatrix} \vdots & \vdots & \vdots \\ \mathbf{e}_{xi} & \mathbf{e}_{yi} & \mathbf{e}_{zi} \\ \vdots & \vdots & \vdots \end{pmatrix} \mathbf{M}_i^{-1}\mathbf{J}_{li}, \tag{15}$$

where $\mathbf{e}_{xi}$, $\mathbf{e}_{yi}$, and $\mathbf{e}_{zi}$ have all zero entries except for the unit value in the row that corresponds to the datum from the $i$th subproblem.

Sometimes, it is particularly useful to compute the diagonal of the square matrix $\mathbf{J}_g^\top\mathbf{J}_g$. In this calculation, every row of $\mathbf{S}$, denoted by $\mathbf{s}_j^\top$ for the $j$th row, left multiplies $\mathbf{J}_g$ at the local level in the same way of equation 15, and then all of the elements in the resulting row vector are squared; finally, those row vectors across different rows of $\mathbf{S}$ are summed up and transposed to obtain the diagonal of $\mathbf{J}_g^\top\mathbf{J}_g$. This procedure can be expressed as

$$\mathbf{diag}(\mathbf{J}_g^\top\mathbf{J}_g)^\top = \sum_j \left( \sum\nolimits_i \mathbf{s}_j^\top \mathbf{M}_i^{-1}\mathbf{J}_{li} \right)^{\circ 2}, \tag{16}$$

where $\circ 2$ denotes the operation of element-wise (Hadamard) square. Because $\mathbf{s}_j^\top$ is sparse, a short cut can be made by sending only nonzero segments of $\mathbf{s}_j^\top$ to the active subproblems if the data are properly ordered.

Using the same synthetic model in Figure 4, we take the sensitivity of a particular datum measured at the same receiver location ($x = -900$ and $y = 500$) at $10^{-3}$ s for $H_z$ as an example. First, the sensitivity is computed using the GD without any approximation; the row vector of sensitivity is presented as a 3D model in Figure 11a. Then, the sensitivities are computed using the SD with the looplets and projected to the global mesh using equation 13 (Figure 11b). The two results are very similar in that the positive and negative values caused by the conductive sphere and survey

geometry are seen on both cross sections at the same locations. The difference at the highest positive sensitivity is noticeable; this is likely due to the coarser local meshes used in the subproblems. From our experience, this degree of discrepancy in the sensitivity is tolerable in the inversion.

## Adaptive subsampling using cross-validation

With many subproblems after decomposition, it is desirable to reduce the number of data used in the inversion as much as possible because we know there is oversampling from the redundancy of data and the regularization in the inversion. A common practice is to spatially downsample the data on a coarse grid with uniform spacing. However, such a static grid has a risk of biasing the inversion result because it acts as a low-pass filter. For example, if the downsampling grid is set to 100 m, the signals varying in a wavelength of 1000 m can still be preserved, but not the signals in a wavelength of 10 m. One solution is to sample the continuous signals dynamically and randomly, which, unlike a uniform grid that only passes low-frequency signals, is not frequency selective (Herrmann, 2010). Using a random sampling scheme with a uniform probability distribution also simplifies the process of subsampling to finding only the optimal number of samples over the sampling domain.

The number of samples required in an inversion depends on the scale of investigation. We use cross-validation in an adaptive search to find a practically optimal number of samples needed for working at any scale. This procedure usually involves two independently chosen subsets of the data: (1) a training subset and (2) a test subset. Ideally, a certain type of operation or investigation is supposed to apply to the entire data set; however, due to the cost of accessing all the data, one can choose to only operate on the training subset, from which a decision or result is then obtained. To prevent the decision from being misled by insufficient sampling, the test subset is used as an independent check to verify whether the decision made based on the training subset also satisfies the test subset. If the validation fails, the number of samples in the training subset is deemed not representative enough, so more samples are added. This adaptive procedure can keep increasing the number of samples until an extreme situation is reached, in which the training subset is the entire data set. The number of samples in the two subsets does not necessarily need to be the same, but we make them equal so that the computing done for the test subset can be reused in the training subset for the next iteration.

Before introducing the subsampling algorithm, we use a quick example to show how cross-validation can be indicative in matching the required number of samples to the scale of investigation. The operation chosen is to calculate the mean value of a data vector. For a given vector of length 100, the training and test subsets estimate the mean value when the number of random samples $N$ varies from 1 to 40. The consistency error, which evaluates the subsampling performance, is defined as the absolute difference between the results from the training and the test subsets for a given $N$. In the first experiment, we have a high-frequency signal (Figure 12a). The mean values estimated by the training and test subsets at different $N$ values are, respectively, indicated by the solid dots and circles in Figure 12b. There are local fluctuations in the consistency error, but the overall trend shows that larger $N$ improves the accuracy. For this high-frequency signal, approximately $N = 30$ samples are required to maintain a consistency error of one (Figure 12c). The second experiment is on a low-frequency signal, representing a broader scale of investigation (Figure 12d). The mean values from the training and test subsets converge quickly at a smaller $N$ (Figure 12e) so that the consistency error of one is achieved for an $N$ value of less than 10 (Figure 12f). Therefore, if a tolerance for the consistency error is given, a minimum value of $N$ can be found, so that the operation achieves the desired answer but with fewer data samples. By using the same idea, we can find the number of samples that matches the scale of investigation at different stages in the inversion. We have successfully implemented this idea for airborne EM, in which a sample was an airborne sounding. Here, because of further decomposition in time, we propose an upgraded algorithm that allows different numbers of samples at different times.

Treating each R-T modeling in a CSEM data set as a sample, our subsampling scheme expects two outcomes: The overall number of samples gradually increases as more structure is added, and more samples of early time data are required compared with late time data. In the algorithm described in Appendix B, we allow each time to have its own number of samples (receivers) $N_i$ (for the $i$th time), its own training subset and test subset of $N_i$ samples, and its own normalized data misfit $\phi_i$. Each subsampling iteration computes the following: (1) a model update using the forward responses and sensitivities from the R-T pairs in the training subset, (2) the data misfits of the preupdate model from the R-T pairs in the test subset, and (3) the data misfits of the postupdate model from the R-T pairs in the same test subset. Then, the two misfits from the test subset, taken before and after the model update, are compared at every time to decide whether the model update can be accepted. The misfit

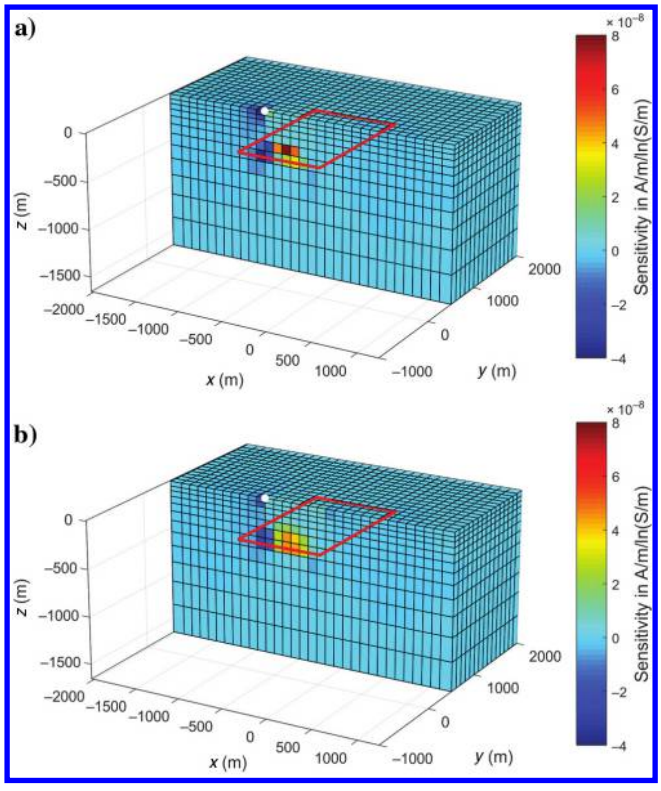


Figure 11. Example sensitivities computed using (a) GD and (b) SD with adaptive looplet refinement. The transmitter loop is indicated by the red lines, and the receiver location is indicated by the white dot.

check at a particular time is acceptable if at least one of the following two statements is true: (1) The postupdate misfit is below a prescribed tolerance and (2) the postupdate misfit has made reasonable improvement from the preupdate misfit. This check is carried out at each time, yielding three possible results:

1) End of the inversion: This happens when the postupdate misfit at every time is below a prescribed tolerance.
2) Accept the model update: This happens when all the times pass the check. In this situation, the model update is accepted and the number of receivers at each time remains unchanged. The regularization parameter is then reduced for the next iteration.
3) Reject the model update: This happens when one or more times cannot pass the check, implying that the current number of samples is deemed insufficient for the current scale of investigation, so more samples need to be added into the training subset to generate a new proposed model update. The times that fail the check will increase $N_i$ for the training and test subsets, whereas other times do not need to change. Then, the inversion starts over without changing the model and regularization parameter.

Gradually, this adaptive scheme will add more receivers across all times, but the emphasis is given to the early times. It is important to note that the random selection of R-T pairs is dynamically renewed at every iteration to ensure that the inversion model is not biased.

## Synthetic inversion

Now that the forward data and sensitivity can be computed using SD, we demonstrate the effectiveness of the new framework with a synthetic inversion example. The synthetic inversion uses the same sphere model in Figure 4 and the same survey layout in Figure 7. Synthetic data at seven times from $10^{-4}$ to $10^{-1}$ s at 54 receiver locations are simulated using the same global mesh in the previous section and a global time stepping with five different step lengths and 76 steps in total. The data are noise free, but 5% of the total field strength is assigned as the uncertainty.

The synthetic data are first inverted using GD with a reference model of a $10^{-3}$ S/m half-space. The GD inversion takes approximately 1 h: 40 s per factorization, 25 s per complete time stepping, and approximately 13 min per model update using the GN method on 12 cores of CPU. The inversion constantly uses approximately 64 GB of memory because the matrix factors and the simulated fields on the global mesh at all time steps need to be stored.

The same synthetic data set is then inverted using SD, with the techniques of adaptive looplets and adaptive R-T subsampling. Table 1 records the number of receivers used at different times and iterations. To complete an iteration for a particular regularization parameter $\beta$, the adaptive approach may need to add more receivers until the cross-validation check is satisfied. The last two times have not required additional receivers; this indicates that there is little information in the data at later times. On the other hand, as the regularization on the model structure is reduced, more

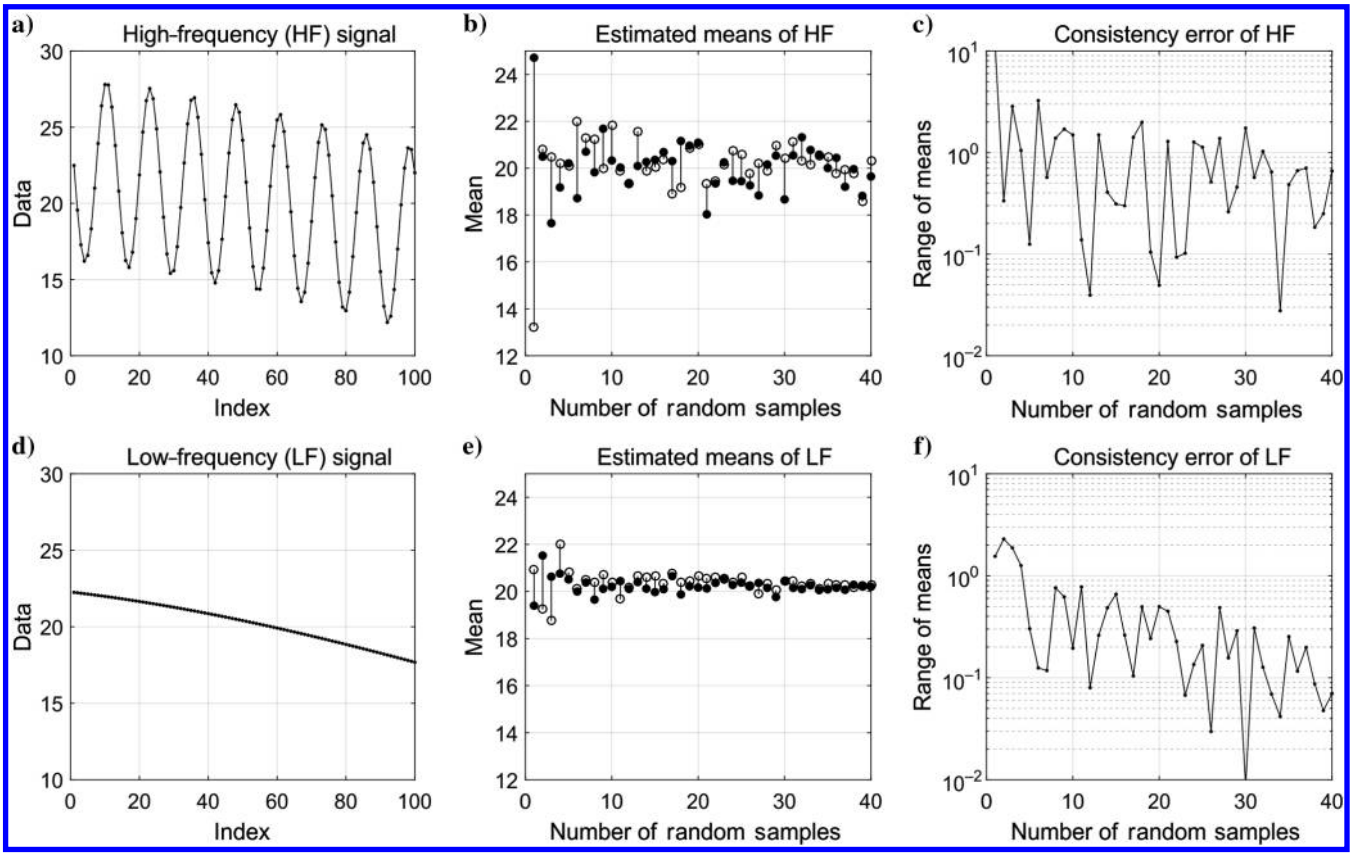


Figure 12. An illustrative example of data subsampling: adapting the number of random samples to the scale of investigation using cross-validation.

receivers at early times are demanded to construct the fine-scale structure. The randomly sampled R-T pairs used in the test subset at the last iteration are plotted in Figure 13; this shows that the early times required more receivers than the late times.

The total CPU time of SD, approximately 43 min on the same computer, is only marginally less than the time of the GD inversion (1 h). However, we make some comments to explain the advantages of our new approach. First of all, we tested SD with only 12 cores. At the last iteration, there are 418 looplet subproblems in 92 R-T pairs, which take 208 s for a complete forward modeling using 12 cores. On average, one subproblem takes 6 s on a single core. If every subproblem was assigned to an individual core, the modeling time could be reduced to a few seconds. Second, our new approach is very memory efficient because every subproblem only has a small Maxwell matrix to factorize and the sensitivities are stored on the local meshes. Such flexibility allows 3D inversion to be inexpensively carried out on desktop computers. Third, this synthetic example is relatively small. If the survey becomes larger, the poor scalability of the GD modeling quickly escalates, but SD scales much better because no forward or sensitivity modeling is carried out on the global mesh. Fourth, from the parallel computing's point of view, every looplet subproblem contributes to the final solution of modeling cumulatively, so a failed computation of a few subproblems does not destroy the final result when parallelized on a large number of processors.

The model recovered using SD is compared with the true model and GD inversion result in Figure 14. Both inversions recover the conductive sphere at the correct location and depth; the weakened conductivity of the sphere in the inversion models is compensated by the smeared "tails" around and below the recovered conductors; this is a result of the smoothness built into the regularization functional. This particular survey layout, involving only one transmitter and two receiver lines, also permits further nonuniqueness of the model. As a simple measure of the similarity between the GD and SG inversion models, we calculate the Euclidean distance (weighted by cell volumes) between the true model vector and

**Table 1. Synthetic inversion using SD, adaptive looplets, and adaptive receiver subsampling.**

| $\beta$ iteration | Number of receivers (time channels 1–7) | | | | | | | CPU time (s) | Cross-validation |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 182 | Fail |
| 1 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 83 | Fail |
| 1 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 133 | Pass |
| 2 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 205 | Fail |
| 2 | 8 | 4 | 4 | 4 | 4 | 2 | 2 | 158 | Pass |
| 3 | 8 | 4 | 4 | 4 | 4 | 2 | 2 | 284 | Fail |
| 3 | 16 | 4 | 8 | 8 | 4 | 2 | 2 | 266 | Pass |
| 4 | 16 | 4 | 8 | 8 | 4 | 2 | 2 | 370 | Fail |
| 4 | 16 | 8 | 16 | 16 | 8 | 2 | 2 | 481 | Fail |
| 4 | 32 | 16 | 16 | 16 | 8 | 2 | 2 | 444 | Pass |

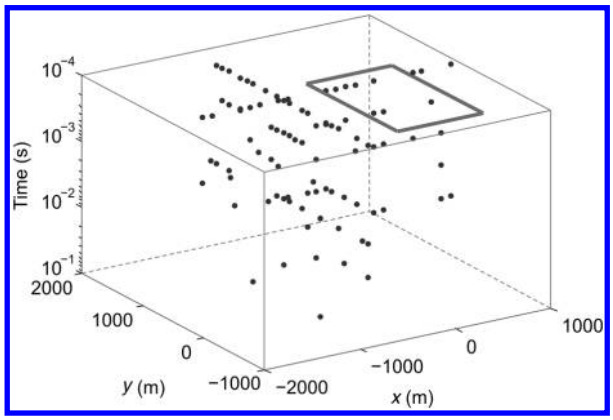


Figure 13. Location of random samples of receiver-time pairs used at the last iteration of the synthetic inversion. The dimension of time is depicted as the vertical axis, and the transmitter loop is indicated by the rectangular shape on the top.

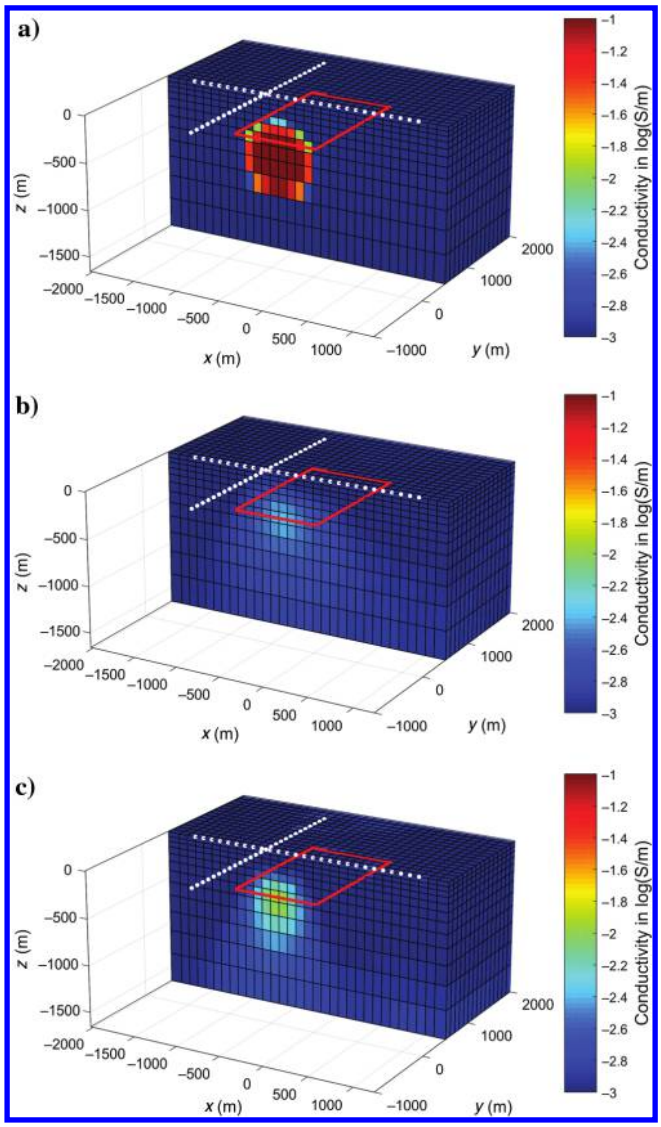


Figure 14. Synthetic inversion models: (a) the true model, (b) the GD model, and (c) the SD model. The red lines and white dots indicate the transmitter loop and receiver locations, respectively.

the inversion model vectors. The distance between the true and GD model is 1168.4, and the SD model is 1138.9. The consistent images and Euclidean distances of the two inversions show that the forward responses and sensitivities are computed in the SD subproblems as reasonably well as in the GD, and the entire framework is shown to be effective.

Finally, a quantitative model validation is carried out by forward modeling the SD inversion model with the GD parameters. The true data misfit can then be calculated without any assumptions or approximations because numerical computations are carried out exactly as in the simulation for the "true" data. For the 1134 data (54 receivers × 7 time channels × 3 components) used here, there are 780 data fit to less than the prescribed uncertainty of 5% of the observed total field strength. The overall normalized data misfit is 1.08, a value close to the misfit estimated by the test subset (1.2) and to that achieved in the GD inversion (0.8). To analyze the misfits more closely, the predicted data of the final SD inversion model are plotted as 3D H-field vectors measured on the surface in Figure 15. They can be compared with the observed data that are also shown in that figure. The distortion of the magnetic field, caused by the conductor near the cross point of the two receiver lines, is recognized and reproduced by the SD inversion model.

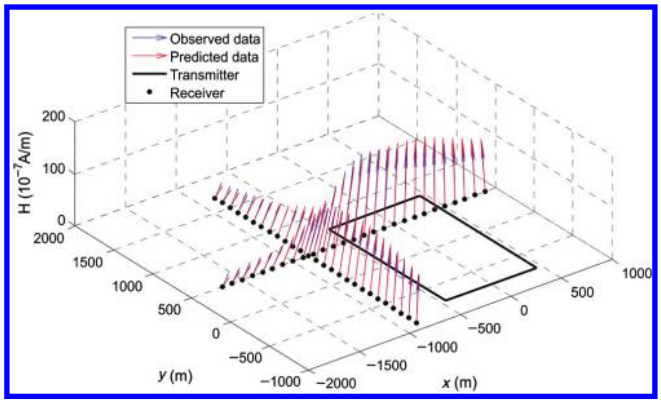


Figure 15. The synthetic observed data and predicted data for the final SD model validation (showing time at $10^{-3}$ s). The vertical axis is used as a scale for the field intensity. The solid lines and dots represent the rectangular transmitter loop and the receivers, respectively.

**Table 2. Field data inversion using SD, adaptive looplets, and adaptive receiver subsampling.**

| $\beta$ iteration | Number of receivers (time channels 1–8) | | | | | | | | CPU time (s) | Cross-validation |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 207 | Pass |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 166 | Pass |
| 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 172 | Pass |
| 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 159 | Fail |
| 4 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 100 | Pass |
| 5 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 159 | Pass |
| 6 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 173 | Fail |
| 6 | 8 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 147 | Pass |
| 7 | 8 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 222 | Pass |
| 8 | 8 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 204 | Pass |
| 9 | 8 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 206 | Fail |
| 9 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 134 | Pass |
| 10 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 234 | Pass |
| 11 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 231 | Pass |
| 12 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 246 | Pass |
| 13 | 8 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 236 | Fail |
| 13 | 8 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 224 | Fail |
| 13 | 16 | 8 | 4 | 4 | 4 | 4 | 2 | 2 | 255 | Pass |

## Field data inversion

We now apply our framework to a ground TEM data set from Lalor volcanogenic massive sulfide (VMS) deposit in central Manitoba, Canada. The deposit was discovered in 2007 within the metavolcanics, metasediments, and granitoids of the Churchill Province in the Chisel Basin portion of the Flin Flon Greenstone Belt, and it is believed to be the largest VMS ever found in this area. The Lalor deposit is characterized by discrete highly conductive ore lenses buried in a relatively resistive host. The ground TEM survey uses a large fixed transmitter loop and an advanced JESSY HTS superconducting quantum interference device (SQUID) magnetometer as the receiver. The large transmitter area and high-precision measurements at late times help to detect deep conductors and also necessitate the SD because of the scale contrast in time.

The layout of the field survey is identical to that used in our synthetic example (Figure 7). The 900 × 1500 m transmitter loop carries a step-off waveform at a base frequency of 0.5 Hz. In the inversion, we use the three-component magnetic field data measured by the SQUID at eight times: 0.87, 2.07, 4.91, 11.66, 27.69, 65.79, 156.27, and 371.22 ms at 54 receiver locations. The data set is assigned an uncertainty of 10% of the total field strength plus a noise floor of $10^{-7}$ A/m.

We first carry out the inversion using the GD. The global mesh used here is geometrically identical with the one used in the synthetic study, except for details of the coordinate system. The global time discretization models all the times from 0.87 to 371.22 ms using four different time step lengths and 74 time steps in total. The computational costs of this inversion are almost the same as those of the synthetic inversion in the previous section, so the time and memory required for one GN step are similar. Starting from a $10^{-3}$ S/m half-space model, the target normalized misfit of one is achieved after six model updates in approximately 80 min on the same 12-core computer.

In the SD inversion, there are 432 R-T pairs (54 receivers × 8 time channels) available for selection. Table 2 records the addition of receivers at different times as the inversion proceeds. Similar to the synthetic example, this inversion finds that the most R-T pairs are needed at the early times and at those times that are significantly affected by the anomaly. Throughout the inversion, a total number of 402 R-T pairs have been used in the inversion, with no more than 44 in any single iteration. With the same inversion parameters, the target data misfit is achieved after 11 model updates in 42 min using 12 cores and approximately 2.8 GB memory is required at the last iteration.

The final conductivity models recovered using GD (iteration 6) and the SD (iteration 11) are compared on two perpendicular cross

sections in the $x$- and $y$-directions in Figure 16. Both inversions provide very similar images of the deep major conductor. They also have similar model norms when measured quantitatively. There are some differences in the small-scale structure near the surface. This is understood because the near-surface features have a relatively minor effect on the overall data compared with the large conductor at depth. The SD inversion captures the primary anomaly first by using fewer samples, and it gradually adds more small-scale features as the adaptive subsampling adds more receivers at early times. When the SD inversion was left to continue, we found that the model at iteration 13 had near-surface structures similar to those from the GD inversion.

Last, the model recovered by the SD inversion is again validated by a forward modeling using the GD parameters. It provided a normalized data misfit of 0.86, which was below the target one achieved by the GD inversion. The field data and the predicted data from the SD inversion model are plotted as 3D H-field vectors measured on the surface. This shows that the most important anomalies due to the major conductor in the data are reasonably reproduced (Figure 17).

## DISCUSSION

### Adaptiveness to the scale of investigation

Further insight can be gained by comparing the synthetic and field data inversions. The two data sets, even though they are collected using the same survey layout, are associated with different scales of investigation; the synthetic model has a shallower target conductor and earlier times, whereas the field example has a deeper target and uses later times. It is difficult for the GD inversion, working on the scales set by the global mesh and global time discretization, to adapt to the appropriate scales of investigation. However, the new framework is able to recognize the difference and only uses 194 looplets in 44 R-T pairs in the last iteration of the field data inversion. This is about half the number used in the synthetic inversion, and it is clear evidence that the adaptive schemes are effective at using as few subproblems as possible.

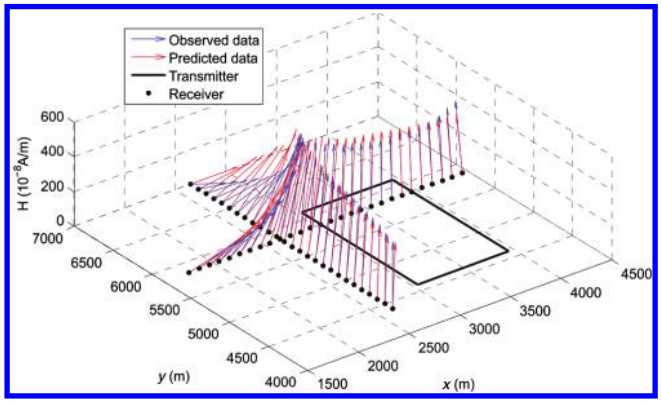


Figure 17. The observed Lalor SQUID data and predicted data for the final SD model validation (showing time at 4.91 ms). The vertical axis is used as a scale for the field intensity. The solid lines and dots represent the rectangular transmitter loop and the receivers, respectively.

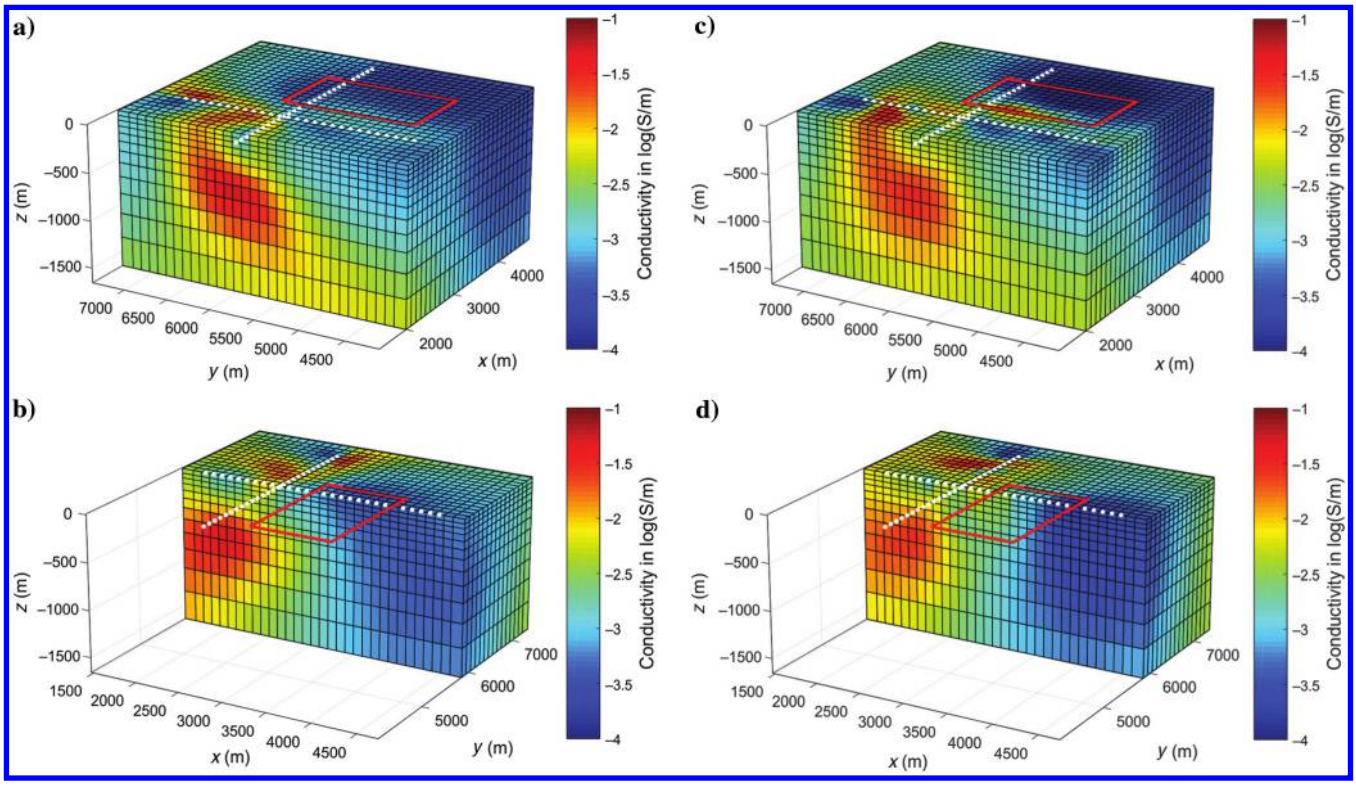


Figure 16. Lalor SQUID field data inversion models: (a) the GD model cut in the $y$-direction, (b) the GD model cut in the $x$-direction, (c) the SD model cut in the $y$-direction, and (d) the SD model cut in the $x$-direction. The red lines and white dots indicate the transmitter loop and receiver locations, respectively.

## Smart sampling

The workflow introduced in this paper involves random sampling at two steps: First, we sample looplets inside the transmitter loop to model a particular R-T pair, and then, we sample receivers confined along two lines at select time channels independently. Although our procedure has worked well so far, this is an area where we believe other sampling methods can be an improvement, for example, a spatial-temporal sampling approach. In the most generic case, a CSEM survey can be decomposed into many subproblems characterized by three parameters: (1) the dipole source location, (2) dipole receiver location, and (3) time. For inductive source surveys, the data volume spanned by those three parameters is smooth, and thus it can be subsampled. Once enough random samples in this data volume are obtained, the full structure of the data volume is restored through interpolation. This method can lead to fewer subproblems because it uses the correlation between space and time, and it can also sample with a nonuniform distribution that pays more attention to the regions with more anomalies. For example, if the training and test subsets in our adaptive subsampling provide very different estimates of the misfit at the same region, it may indicate that more data should be sampled near that place in the next iteration.

Also, the downsampling and the numerical integration within a large transmitter can be treated as a signal compression problem. Advanced techniques developed for numerical analysis, such as generalized Gaussian quadratures using exponential decomposition (Beylkin and Monzón, 2002), may be applied to optimize the SD framework.

## Practical considerations when forming subproblems

We have shown that the difficulty of dealing with a large number of subproblems can be mitigated by an adaptive subsampling approach in which we only need to model the most "informative" part of the entire data set. However, there are certain circumstances in which the data cannot be significantly downsampled, for example, in forward modeling requesting tight data spacing or if the data contain high-frequency noise. Atomic decomposition in these cases is likely to be inefficient, so it might be necessary to package more atomic problems into one subproblem such as that used in the airborne inversion. In addition, considering the realistic variation of existing computing devices, it is usually worthwhile to explore different grouping options available when forming the subproblems. Eventually, this boils down to the trade-off between the size of equation 1 and the number of times it must be solved. The conventional approach will have the fewest systems to solve but their sizes can be extremely large, whereas atomic decomposition will have the smallest systems but results in a large number of subsystems. Ideally, we want to generate subproblems that contain as many atomic blocks as possible without significantly increasing the size of equation 1. It is difficult to precisely formulate this optimization process because too many factors are involved; however, we realize that a workable decomposition can be designed if the following aspects are considered:

1) Capability of a single core: It is important to know how the computational time for solving a subproblem varies with the size of the Maxwell system equation 1 and the amount of storage required. This provides insight regarding the largest size of subproblem that might be used efficiently.
2) "Distance" between the atomic blocks: Atomic building blocks that are close to each other in time or space can often be combined into one subproblem without significant growth of equation 1. When including more atomic blocks in the subproblems, one needs to evaluate how the increase in size of the subproblem compares with the savings from having fewer subproblems. Expanding the size of the subproblem stops once the grouping becomes uneconomical.
3) Communication time: The SD models subproblems independently, so the communication time is minimal. However, a large number of subproblems can still result in notable communication, and this encourages more atomic blocks in one subproblem. This will vary between platforms.
4) The total number of cores with sufficient memory: The scale of parallelization on a computer is crucial to the implementation of the decomposition technique because it sets hard constraints on the number of subproblems that each core must handle.
5) The total time required for forward modeling: This time can be approximately formulated as $T_f = T_s \times (N_{\text{sub}}/N_p) + T_{\text{com}}$, where $N_{\text{sub}}$ is the number of subproblems, $T_s$ is the time to solve one subproblem, $N_p$ is the number of cores, and $T_{\text{com}}$ is the communication time. The previous items in this list need to be adjusted, so that this time is as small as possible.

## CONCLUSIONS

Despite many advancements in the algorithms, 3D CSEM inversions still have a fundamental difficulty of scalability that generates computational complexities and prevents the inversions from being carried out efficiently. A large amount of computing is associated with modeling multiple scales of investigation with one universal design, for example, having a global mesh that has small cells near the transmitter for early times and keeps these same cells for modeling later times or using a fixed mesh with a sufficiently large domain that can handle diffusion distances associated with early- and late-time data. We identify three computational complexities, namely, (1) space complexity, (2) time complexity, and (3) optimization complexity. To handle the space and time complexities, we propose a generic SD framework to effectively reduce the scale contrast by modeling each datum separately. We define an atomic building block to be a solution of Maxwell's equations for a single datum at a particular receiver location, time, or frequency due to an electric or magnetic dipole source. This problem is solved with a locally optimized mesh and time-stepping scheme. An atomic problem might be used directly for data simulation, for example, in airborne EM data with dipole sources and receivers. However, if the transmitter is spatially large, simulating a field datum can be achieved by representing the transmitter as a summation of dipoles and evaluating the sum of the responses from atomic problems. Upon decomposition, each subproblem is a small, self-contained EM modeling problem that does not need information from other subproblems. The subproblems can thus be solved with massive parallelization, with scalability that is better than other approaches that produce more interdependent subproblems.

An atomic decomposition can potentially lead to an inordinate number of subproblems, so we offer two solutions: First, one may be able to combine multiple atomic building blocks into one subproblem that can still be solved quickly on a single core. Second, for

inversion, further reduction of the optimization complexity can be achieved by downsampling the data used in the inversion. We develop an adaptive random subsampling procedure that matches the sampling rate to the scale of investigation throughout the inversion. Our approach is designed to minimize the number of subproblems in two ways: (1) It recognizes the wavelengths of signals in space and time and samples accordingly. This leads to using more data at early times than at late times and (2) it uses fewer samples at the early iterations of inversion when only large-scale features are sought. Thus, the number of subproblems needed increases as the regularization parameter in the inversion algorithm decreases.

The local mesh, local time discretization, and adaptive random subsampling are the three major components in the SD framework. However, they may be of different relative importance for different types of EM survey. The improvement of efficiency made by using SD will depend on how much overcomputing (computational complexity or scale contrast) is in conventional GD methods. For example, a towed EM survey enjoys significant improvement by using a local mesh because the contrast between the smallest diffusion scale and the size of survey can be very large. Because the challenging airborne inversion has been previously treated, in this paper, we use a large-loop TEM survey as a more complicated example since the large transmitter loop needs to be represented by many magnetic dipole sources, referred to as looplets. The minimum number of looplets required depends on the conductivity model, the receiver offset, and most importantly upon the time; the looplets are selected by another adaptive refinement algorithm. The entire framework of SD, along with the adaptive data subsampling and adaptive looplet refinement, is tested using a synthetic example. Our approach produces the same results as a standard method, but with reduced time and memory usage. We used only 12 cores when solving 418 subproblems, and hence more speed up can be anticipated by increasing the number of cores or processors. If each subproblem had a designated core, a complete 3D forward modeling would take only a few seconds and the complete inversion of the large-loop TEM data set would have been approximately 5 min, a time comparable with the 1D and geometric inversions. Last, a field data set collected using the SQUID magnetometer at Lalor massive sulfide deposit was inverted using SD. This example showed that our approach can be practically implemented to carry out an inversion in reduced time and memory without a sacrifice in quality.

The primary goal of this paper is to design a practical methodology for carrying out 3D CSEM inversions efficiently in a computing environment in which many processors are available. Even at the time of writing this paper, the potential to use 1000 cores is readily achievable. This number is likely to increase exponentially with time. We have outlined a framework that is able to convert a cumbersome 3D problem to a large number of smaller and independent subproblems suitable for parallel computing. We note that many of the procedures in the framework presented here (e.g., sampling strategies) were simple implementations, and the parallelization was implemented in a relatively small computing environment. These components can be further investigated and improved, and that will result in additional computational efficiency.

## ACKNOWLEDGMENTS

The authors thank the sponsoring companies of UBC-Geophysical Inversion Facility and NSERC for the funding. R. Shekhtman provided assistance to the software programming. The field SQUID data at Lalor are courtesy of Hudbay Minerals Inc. and Discovery International Geophysics Inc. D. Woods provided technical information about the SQUID survey at Lalor. We also thank M. Commer and two anonymous reviewers for their expert comments that improved this manuscript and suggestions for further development.

## APPENDIX A

## FAST DISCRETE MODEL CONVERSION BETWEEN ARBITRARY 3D MESHES

The representation of the global model on a local mesh is crucial in SD. Here, we document an intersecting-volume weighted material averaging method that implements fast model conversion between two meshes embedded in different coordinate systems using the proxy of a point cloud. We assume that the following information is known: the original model vector $\mathbf{m}_1$ on mesh 1 in coordinate system 1, the destination mesh 2 in coordinate system 2, and the coordinate transform parameters, which, in case of two Cartesian systems, is a $3 \times 3$ rotation matrix $\mathbf{M}$ and a translation vector $\mathbf{t}$, so that for a point $\mathbf{p}_1 = (x_1, y_1, z_1)^\top$ in system 1 and its co-ordinate $\mathbf{p}_2 = (x_2, y_2, z_2)^\top$ in system 2, we have

$$\mathbf{p}_2 = \mathbf{M}\mathbf{p}_1 + \mathbf{t}. \tag{A-1}$$

We also require that a point can be located in a mesh in terms of a cell index in the model vector; there are no restrictions on the structure of the meshes. The goal is to find an averaging matrix $\mathbf{A}$, with which we can get the model $\mathbf{m}_2$ on the destination mesh by applying the following averaging operation:

$$\mathbf{m}_2 = \mathbf{A}\mathbf{m}_1. \tag{A-2}$$

An entry $a_{ij}$ in $\mathbf{A}$ represents the weight, essentially the normalized intersecting volume, when calculating the relative contribution of cell $j$ in $\mathbf{m}_1$ to cell $i$ in $\mathbf{m}_2$. Summing elements of $\mathbf{A}$ along rows should yield a vector of all ones:

$$\sum_{j=1}^{N_{\mathbf{m}_1}} a_{ij} = \mathbf{1}. \tag{A-3}$$

Matrix $\mathbf{A}$ is usually sparse because a cell in $\mathbf{m}_2$ may intersect only a few cells in $\mathbf{m}_1$. If inactive cells are involved, the corresponding rows and columns of $\mathbf{A}$ must be removed.

The point cloud approach first initiates $N_p$ random points in every cell in $\mathbf{m}_2$; then, the points are transformed to system 1 and their cell indices in $\mathbf{m}_1$ are calculated; finally, $\mathbf{A}$ is assembled by counting the number of points: If a point from the $i$th cell in $\mathbf{m}_2$ is found in the $j$th cell in $\mathbf{m}_1$, then $a_{ij}$ increments by unity. Each row of $\mathbf{A}$ is normalized by its sum before $\mathbf{A}$ is applied to $\mathbf{m}_1$. When $N_p$ approaches infinity, the estimated intersecting volumes are identical to the results computed using the rigorous analytic geometry; however, we have noticed that $N_p \approx 50$ is adequate to quickly produce a good converted model for the purpose of EM modeling. The point cloud approach is especially useful when using a rigorous method is expensive for complex meshes.

## APPENDIX B

## INVERSION ALGORITHM USING RANDOM AND ADAPTIVE DATA SUBSAMPLING

We design Algorithm 1 to effectively minimize the number of receiver-time pairs needed in an inversion by adapting the number of receivers to the scale of investigation at different times. In Algorithm 1, $\mathbf{m}$ is the current model, $\delta\mathbf{m}$ is the model update, $\beta$ is the regularization parameter, $\mu$ is the scalar tuning the misfit reduction criterion in the cross-validation, $tol$ is the prescribed stop criterion for the data misfit, $N_t$ is the number of time channels, and $N_i$ is the number of receivers chosen for the $i$th time; the selection is random, independent, and renewed at every iteration. The computing results of the test subset from the previous iteration, including the forward data and the designed local meshes, are recycled for the training subset at the next iteration, so the computations done for the test subset have additional use after the cross-validation, and $N_i$ is always doubled when adding more receiver samples.

**Algorithm 1. Generic inversion workflow for the subsampling of receiver-time pairs.**

Initialization:

Select $N_i$ receivers at $i$th time channel for training subset $\mathbf{S}_i^{\text{train}}(i = 1, \ldots, N_t)$

Compute forward responses $F(\mathbf{m})_i^{\text{train}}$ at $\mathbf{S}_i^{\text{train}}(i = 1, \ldots, N_t)$

Compute sensitivity $\mathbf{J}(\mathbf{m})_i^{\text{train}}$ at $\mathbf{S}_i^{\text{train}}(i = 1, \ldots, N_t)$

Empty test subset $\mathbf{S}_i^{\text{test}}(i = 1, \ldots, N_t)$

**Repeat**

Solve for $\delta\mathbf{m}$ using $\beta, \mathbf{m}, F(\mathbf{m})_i^{\text{train}}$ and $\mathbf{J}(\mathbf{m})_i^{\text{train}}$ at $\mathbf{S}_i^{\text{train}}(i = 1, \ldots, N_t)$

**for** $i = 1, \ldots, N_t$ **do**

**if** $\mathbf{S}_i^{\text{test}}$ is empty **then**

Select $N_i$ receivers at $i$th delay time for test subset $\mathbf{S}_i^{\text{test}}$

Compute forward responses $F(\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}$

Compute data misfit $\phi(\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}$

**end if**

**end for**

Compute forward responses $F(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}(i = 1, \ldots, N_t)$

Compute data misfit $\phi(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}(i = 1, \ldots, N_t)$

**if** $\phi(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}} \le tol$ **or** $\phi(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}} < \mu \cdot \phi(\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}(i = 1, \ldots, N_t)$ **then**

Compute sensitivity $\mathbf{J}(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}(i = 1, \ldots, N_t)$

$\mathbf{S}_i^{\text{train}} \leftarrow \mathbf{S}_i^{\text{test}}(i = 1, \ldots, N_t)$

$F(\mathbf{m})_i^{\text{train}} \leftarrow F(\mathbf{m})_i^{\text{test}}(i = 1, \ldots, N_t)$

$\mathbf{J}(\mathbf{m})_i^{\text{train}} \leftarrow \mathbf{J}(\mathbf{m})_i^{\text{test}}(i = 1, \ldots, N_t)$

Empty $\mathbf{S}_i^{\text{test}}, F(\mathbf{m})_i^{\text{test}}, \mathbf{J}(\mathbf{m})_i^{\text{test}}(i = 1, \ldots, N_t)$

Reduce $\beta$

$\mathbf{m} \leftarrow \mathbf{m} + \delta\mathbf{m}$

**Else**

**for** $i = 1, \ldots, N_t$ **do**

**if** $\phi(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}} > tol$ **and** $\phi(\mathbf{m} + \delta\mathbf{m})_i^{\text{test}} \ge \mu \cdot \phi(\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}$ **then**

Compute sensitivity $\mathbf{J}(\mathbf{m})_i^{\text{test}}$ at $\mathbf{S}_i^{\text{test}}$

$\mathbf{S}_i^{\text{train}} \leftarrow \mathbf{S}_i^{\text{train}} \cup \mathbf{S}_i^{\text{test}}$

$F(\mathbf{m})_i^{\text{train}} \leftarrow F(\mathbf{m})_i^{\text{train}} \cup F(\mathbf{m})_i^{\text{test}}$

$\mathbf{J}(\mathbf{m})_i^{\text{train}} \leftarrow \mathbf{J}(\mathbf{m})_i^{\text{train}} \cup \mathbf{J}(\mathbf{m})_i^{\text{test}}$

$N_i = N_i \times 2$

Empty $\mathbf{S}_i^{\text{test}}, F(\mathbf{m})_i^{\text{test}}, \mathbf{J}(\mathbf{m})_i^{\text{test}}$

**end if**

**end for**

**end if**

**until** $\phi(\mathbf{m})_i^{\text{test}} \le tol(i = 1, \ldots, N_t)$.

## REFERENCES


Alumbaugh, D. L., G. A. Newman, L. Prevost, and J. N. Shadid, 1996, Three-dimensional wideband electromagnetic modeling on massively parallel computers: Radio Science, **31**, 1–23, doi: 10.1029/95RS02815.

Asvadurov, S., V. Druskin, and S. Moskow, 2007, Optimal grids for anisotropic problems: Electronic Transactions on Numerical Analysis, **26**, 55–81.

Aurenhammer, F., 1991, Voronoi diagrams – A survey of a fundamental geometric data structure: ACM Computing Surveys (CSUR), **23**, 345–405, doi: 10.1145/116873.116880.

Badea, E. A., M. E. Everett, G. A. Newman, and O. Biro, 2001, Finite-element analysis of controlled-source electromagnetic induction using Coulomb-gauged potentials: Geophysics, **66**, 786–799, doi: 10.1190/1.1444968.

Beylkin, G., and L. Monzón, 2002, On generalized Gaussian quadratures for exponentials and their applications: Applied and Computational Harmonic Analysis, **12**, 332–373, doi: 10.1006/acha.2002.0380.

Börner, R.-U., O. G. Ernst, and K. Spitzer, 2008, Fast 3-D simulation of transient electromagnetic fields by model reduction in the frequency domain using Krylov subspace projection: Geophysical Journal International, **173**, 766–780, doi: 10.1111/j.1365-246X.2008.037.

Commer, M., and G. Newman, 2004, A parallel finite-difference approach for 3D transient electromagnetic modeling with galvanic sources: Geophysics, **69**, 1192–1202, doi: 10.1190/1.1801936.

Commer, M., and G. A. Newman, 2006a, An accelerated time domain finite difference simulation scheme for three-dimensional transient electromagnetic modeling using geometric multigrid concepts: Radio Science, **41**, RS3007.

Commer, M., and G. A. Newman, 2006b, Large scale 3D EM inversion using optimized simulation grids nonconformal to the model space: 76th Annual International Meeting, SEG, Expanded Abstracts, 760–764.

Commer, M., and G. A. Newman, 2008, New advances in three-dimensional controlled-source electromagnetic inversion: Geophysical Journal International, **172**, 513–535, doi: 10.1111/j.1365-246X.2007.0366.

Commer, M., G. A. Newman, J. J. Carazzone, T. A. Dickens, K. E. Green, L. A. Wahrmund, D. E. Willen, and J. Shiu, 2008, Massively parallel electrical-conductivity imaging of hydrocarbons using the IBM Blue Gene/L supercomputer: IBM Journal of Research and Development, **52**, 93–103, doi: 10.1147/rd.521.0093.

Cox, L. H., G. A. Wilson, and M. S. Zhdanov, 2010, 3D inversion of airborne electromagnetic data using a moving footprint: Exploration Geophysics, **41**, 250–259, doi: 10.1071/EG10003.

Cox, L. H., G. A. Wilson, and M. S. Zhdanov, 2012, 3D inversion of airborne electromagnetic data: Geophysics, **77**, no. 4, WB59–WB69, doi: 10.1190/geo2011-0370.1.

Du Fort, E. C., and S. P. Frankel, 1953, Stability conditions in the numerical treatment of parabolic differential equations: Mathematical Tables and Other Aids to Computation, **7**, 135–152, doi: 10.2307/2002754.

Grayver, A. V., R. Streich, and O. Ritter, 2013, Three-dimensional parallel distributed inversion of CSEM data using a direct forward solver: Geophysical Journal International, **193**, 1432–1446, doi: 10.1093/gji/ggt055.

Haber, E., U. Ascher, D. Aruliah, and D. Oldenburg, 2000, Fast simulation of 3D electromagnetic problems using potentials: Journal of Computational Physics, **163**, 150–171, doi: 10.1006/jcph.2000.6545.

Haber, E., and U. M. Ascher, 2001, Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients: SIAM Journal on Scientific Computing, **22**, 1943–1961, doi: 10.1137/S1064827599360741.

Haber, E., D. W. Oldenburg, and R. Shekhtman, 2007, Inversion of time domain three-dimensional electromagnetic data: Geophysical Journal International, **171**, 550–564, doi: 10.1190/geo2012-0131.1.

Haber, E., and C. Schwarzbach, 2014, Parallel inversion of large-scale airborne time-domain electromagnetic data with multiple octree meshes: Inverse Problems, **30**, 055011, doi: 10.1088/0266-5611/30/5/055011.
Herrmann, F. J., 2010, Randomized sampling and sparsity: Getting more information from fewer samples: Geophysics, **75**, no. 6, WB173–WB187, doi: 10.1190/1.3506147.
Hestenes, M. R., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, 409–436, doi: 10.6028/jres.049.044.
Hohmann, G. W., 1975, Three-dimensional induced polarization and electromagnetic modeling: Geophysics, **40**, 309–324, doi: 10.1190/1.1440527.
Holtham, E., and D. W. Oldenburg, 2012, Large-scale inversion of ZTEM data: Geophysics, **77**, no. 4, WB37–WB45, doi: 10.1190/geo2011-0367.1.
Jahandari, H., and C. Farquharson, 2015, Finite-volume modelling of geophysical electromagnetic data on unstructured grids using potentials: Geophysical Journal International, **202**, 1859–1876, doi: 10.1093/gji/ggv257.
Key, K., and J. Ovall, 2011, A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling: Geophysical Journal International, **186**, 137–154, doi: 10.1111/j.1365-246X.2011.05025.x.
Moskow, S., V. Druskin, T. Habashy, P. Lee, and S. Davydycheva, 1999, A finite difference scheme for elliptic equations with rough coefficients using a Cartesian grid nonconforming to interfaces: SIAM Journal on Numerical Analysis, **36**, 442–464, doi: 10.1137/S0036142997318541.
Nabighian, M. N., and J. C. Macnae, 1991, Time domain electromagnetic prospecting methods, *in* M. N. Nabighian, Electromagnetic methods in applied geophysics: SEG, 427–520.
Newman, G. A., and D. L. Alumbaugh, 1997, Three-dimensional massively parallel electromagnetic inversion — I. Theory: Geophysical Journal International, **128**, 345–354, doi: 10.1111/j.1365-246X.1997.tb01559.x.
Newman, G. A., and M. Commer, 2005, New advances in three dimensional transient electromagnetic inversion: Geophysical Journal International, **160**, 5–32, doi: 10.1111/j.1365-246X.2004.0246.
Newman, G. A., G. W. Hohmann, and W. L. Anderson, 1986, Transient electromagnetic response of a three-dimensional body in a layered earth: Geophysics, **51**, 1608–1627, doi: 10.1190/1.1442212.
Oldenburg, D. W., E. Haber, and R. Shekhtman, 2013, Three-dimensional inversion of multisource time domain electromagnetic data: Geophysics, **78**, no. 1, E47–E57, doi: 10.1190/geo2012-0131.1.
Pridmore, D., G. Hohmann, S. Ward, and W. Sill, 1981, An investigation of finite-element modeling for electrical and electromagnetic data in three dimensions: Geophysics, **46**, 1009–1024, doi: 10.1190/1.1441239.
SanFilipo, W. A., and G. W. Hohmann, 1985, Integral equation solution for the transient electromagnetic response of a three-dimensional body in a conductive half-space: Geophysics, **50**, 798–809, doi: 10.1190/1.1441954.
Schwarzbach, C., and E. Haber, 2013, Finite element based inversion for time-harmonic electromagnetic problems: Geophysical Journal International, **193**, 615–634, doi: 10.1093/gji/ggt006.
Um, E. S., J. M. Harris, and D. L. Alumbaugh, 2010, 3D time-domain simulation of electromagnetic diffusion phenomena: A finite-element electric-field approach: Geophysics, **75**, no. 4, F115–F126, doi: 10.1190/1.3473694.
van den Eshof, J., and M. Hochbruck, 2006, Preconditioning Lanczos approximations to the matrix exponential: SIAM Journal on Scientific Computing, **27**, 1438–1457, doi: 10.1137/040605461.
Wang, G. L., C. Torres-Verdn, J. M. Salazar, and B. Voss, 2009, Fast 2D inversion of large borehole EM induction data sets with an efficient Fréchet-derivative approximation: Geophysics, **74**, no. 1, E75–E91, doi: 10.1190/1.3033213.
Wang, T., and G. W. Hohmann, 1993, A finite-difference, time-domain solution for three-dimensional electromagnetic modeling: Geophysics, **58**, 797–809, doi: 10.1190/1.1443465.
Wang, T., M. Oristaglio, A. Tripp, and G. Hohmann, 1994, Inversion of diffusive transient electromagnetic data by a conjugate-gradient method: Radio science, **29**, 1143–1156, doi: 10.1029/94RS00617.
Xie, G., J. Li, E. L. Majer, D. Zuo, and M. L. Oristaglio, 2000, 3-D electromagnetic modeling and nonlinear inversion: Geophysics, **65**, 804–822, doi: 10.1190/1.1444779.
Yang, D., and D. W. Oldenburg, 2012a, Practical 3D inversion of large airborne time domain electromagnetic data sets: 22nd International Geophysical Conference and Exhibition, ASEG Extended Abstracts, 1–4.
Yang, D., and D. W. Oldenburg, 2012b, Three-dimensional inversion of airborne time-domain electromagnetic data with applications to a porphyry deposit: Geophysics, **77**, no. 2, B23–B34, doi: 10.1190/geo2011-0194.1.
Yang, D., D. W. Oldenburg, and E. Haber, 2014, 3-D inversion of airborne electromagnetic data parallelized and accelerated by local mesh and adaptive soundings: Geophysical Journal International, **196**, 1942–1507.
Zaslavsky, M., V. Druskin, and A. Abubakar, 2013, Large-scale Gauss-Newton inversion of transient controlled-source electromagnetic measurement data using the model reduction framework: Geophysics, **78**, no. 4, E161–E171, doi: 10.1190/geo2012-0257.1.
Zaslavsky, M., V. Druskin, and L. Knizhnerman, 2011, Solution of 3D time-domain electromagnetic problems using optimal subspace projection: Geophysics, **76**, no. 6, F339–F351, doi: 10.1190/geo2011-0088.1.