6th International Symposium on Three-Dimensional Electromagnetics
Berkeley, California, USA, March 28-30, 2017

Development of Flexible Object Oriented MATLAB Codes for 3D EM Modeling

Gary D. Egbert[1] and Maxim Yu. Smirnov[2], Maria Cherovatova[3]
[1] Oregon State University
[2] Luleå University of Technology
[3] University of Muenster

SUMMARY

We have developed a flexible set of object oriented Matlab® codes for 3D EM forward modeling, using a staggered grid finite difference/finite volume approach. The code is being used for development and testing of improved forward modeling for 3D inversion. Because the code is in Matlab®, it is easily accessible to students. Basic modules for 1D, 2D, and 3D modeling are included. Extension for more specialized cases are built on these, so far including general anisotropy (1D,2D,3D), spherical coordinates (regional and full globe), and (the original motivation) multi-resolution grids based on a quad-tree approach. The system makes it quite easy to explore different solution approaches (e.g., a multigrid approach is being tested), different boundary conditions, different sources (controlled source, tides), etc. We give example applications to diverse problems, MT, CSEM, and local and regional modeling of EM fields due to ocean tides.

Keywords: forward modeling, inversion, finite difference, software, object oriented programming

INTRODUCTION

The impetus to develop 3D forward code in Matlab® originally came out of efforts to develop new modelling approaches for the ModEM3D MT code (Egbert and Kelbert, 2012; Kelbert et al., 2014). Initial testing and debugging of complex programs is much easier in an interactive programming language, and the basic Matlab® system already has many of the tools needed for manipulating and solving linear systems, and many libraries and toolkits that support tasks that are more specialized are available. Thus, to develop a multi-resolution modelling scheme for MT that allows for higher resolution near the surface (Cherevatova et al., 2017) we decided on a strategy of prototyping in Matlab®, followed by translation to Fortroan90. By taking advantage of the Matlab® object oriented programming model, we have expanded this initial development into a flexible finite difference/finite modeling toolkit with much broader capabilities. We are using these codes for prototyping algorithms before porting to Fortran90 or C, for student projects and education, and for specialized modelling. The system supports multiple grids (standard staggered-grid, multi-resolution, octree (Haber and Heldman, 2007), in both Cartesian and spherical geometry. Multiple formulations of the forward problem (e.g., curl-curl equations, potentials, variants) are (or can) be supported, and a range of forward solution approaches have been tried (direct, Krylov space methods, multi-grid). We summarize some key aspects of our implementation here, and give a few examples of use. The paper by Cherovatova et al. (2017) provides a more detailed example of a modelling approach developed and tested within this framework.

SYSTEM OVERVIEW

Although the focus of our development (and our discussion here) is on 3D, 1D and 2D codes are also included (Fig.1). The 2D variant is particularly useful as a simplified, but still substantive, introduction for students. Both 1D and 2D codes are also useful for computing boundary data for the 3D case (Fig. 2). The basic 2D and 3D codes use staggered grids, and are for isotropic problems in standard Cartesian coordinates. A number of extensions can be built on these (now stable) base classes. For example (Fig. 1), codes allowing for general anisotropy have been developed for both 2D and 3D (as a student project). The multi-resolution (quad-tree; MR) modelling scheme presented in Cherevatova et al. (2017), has also been developed as an extension of the base 3D staggered-grid modeling code. Other extensions already developed include modelling code for regional and. global spherical coordinate cases. Combinations of these (e.g., anisotropic multi-resolution, spherical coordinate multi-resolution) can be easily implemented, and are planned.

Fig. 2 provides an overview of the main components needed for 3D modeling, and their dependency. At the lowest level, a basic tensor-product **grid** class is defined. Electric and magnetic fields defined on this grid are represented with a **vector** class, which allows for a natural representation of components on edges or faces

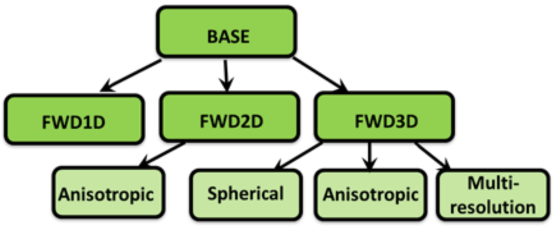

Figure 1. The code is organized into two levels: the darker green boxes denote base classes supporting 1D, 2D, and 3D forward modelling in the simplest case, with staggered tensor product grids. These are stable now, and are not intended to be modified further. The second level (ligher shade of green) includes extensions, which we show only a few examples of, including the multi-resolution and anisotropic extensions.

of the staggered grid. A **scalar** class is also defined, allowing (for example) for a scalar potential defined on nodes or cell centers. The vector and scalar classes include methods that simplify characterization of grid topology, definition of boundaries, and conversion of the field objects to simple column vectors, to allow use of Matlab sparse matrix operations and solvers. For the MR scheme of Cherevatova et al. (2017) the grid is represented as a stack of uniform staggered grids, with horizontal resolutions varying by a factor of 2. Vectors and scalars are similarly represented as stacks of the basic staggered grid objects.

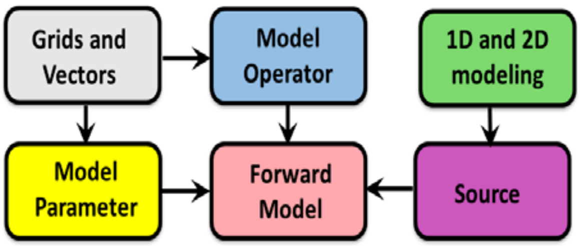

Figure 2. Main components of basic 3D modelling classes include grids and vectors, basic model operators (curl, gradient, averaging operators), model parameter and source. The source classes often use lower-dimensional modelling classes to create boundary data. All of these components feed into and support the forward modelling classes, which include the specific model equation formulation, and solvers.

Model operators start with basic curl and gradient operators, discussed in more detail below. These operators can then be used in a variety of **forward solver** classes, which (at the level of detail discussed here) combine a specific formulation of the forward problem with a solution approach. We have initially formulated the forward problem in terms of the curl-curl equation for the electric fields, and solved these either directly or with preconditioned Krylov space methods. Solver variants with and without a divergence correction have been developed. Alternative formulations are straightforward, e.g., solving a second order system for magnetic fields, or for vector and scalar potentials. With all of these problem formulations the direct and iterative solution approaches can be used. Other strategies, such as multi-grid, are also straightforward and are being tested within this general framework. Other required components include the **model parameter** (and associated mapping to equation coefficients) and the **source**. For MT the last class is used to provide boundary data, generally derived from solving a 1D or 2D problem; for CSEM this provides source terms (and for a secondary field formulation, the primary fields).

MODEL OPERATOR DETAILS

Fig. 3 provides additional detail on the **model operator**, which is generally constructed as a combination of **operator topology** and **metric elements** classes. The basic operator topology class defines curl and gradient operators on a staggered rectangular unit grid. The curl operator maps from edges to faces, and the gradient from nodes (cell corners) to edges. Both are represented as sparse matrices, with all elements ±1. Extensions of the basic staggered grid topology include the MR case (allowing for a grid with horizontal resolution that varies with depth (by factors of 2), and the global topology where lateral boundaries either connect, or collapse to a pole. The basic metric elements class provides lengths of edges, areas of faces, and volumes of cells, both for the primary and dual components on the staggered grid.

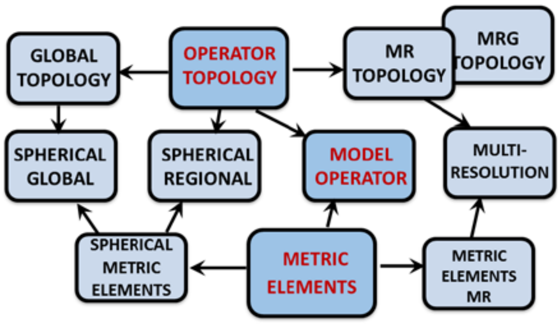

Figure 3. Details of the model operator classes. The operator is built from two components: Operator Topology, and Metric Elements. For the base case, the topology corresponds to the standard staggered grid, and metric elements are for Cartesian geometry. These classes can be extended (to the mult-resolution or global cases) or overloaded for spherical geometry.

Separate classes are provided for Cartesian and spherical coordinates. Metric elements for the MR case can be constructed from the basic staggered grid case. The different topology and metric classes can be combined to construct operators for new cases, such as regional MR in spherical coordinates.

SOME EXAMPLE USES

The multi-rresolution forward code of Cherevatova et al. (2017) provides a good example of the sort of research task for which the Matlab code has proven useful. A key issue with the multi-resolution (quadtree or octree) approach is treatment of "hanging nodes" at the interface between sub-grids of varying resolution. Two possible approaches to defining the set of active nodes

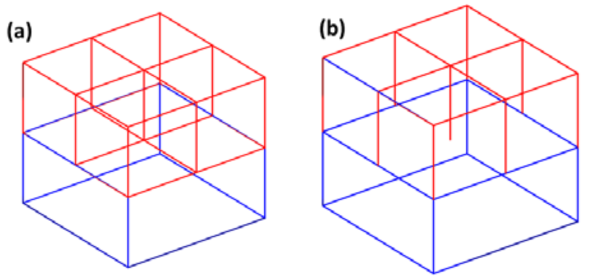

Figure 4. Two possible implementation of the grid topolgy for a multi-resolution forward modeling scheme, allowing variations (by factors of 2) in horizontal resolution with depth. For (a) the interface layer is considered part of the fine sub-grid; in (b) the interface is part of the coarse sub-grid.

on the interface between fine and coarse grids are shown in Fig. 4. In the first case (comparable to the approach taken by Haber and Heldman (2007)), fine-grid edges on the horizontal interface are considered active. In the second, the interface is considered part of the coarse grid, with no active edges ("degrees of freedom") defined. As discussed in Cherevatova et al. (2017) the second approach turns out to have substantial advantages – in particular simple definitions of curl (which require interpolation to "missing edges" on the interface) lead naturally to a symmetric curl-curl operator, and more accurate and efficient solutions. Both approaches are readily implemented within our framework, requiring only minor modifications to grids, and operator topology classes.

A second example is provide by experiments we have done with alternative formulations of the forward problem. The operator $\nabla\times\nabla$ is singular, making the quasi-static equations nearly singular (at low frequencies, or in the air). One solution is to periodically apply a divergence correction, explicitly enforcing continuity of currents in the Earth, and electric fields in the air. An alternative approach is to modify the curl-curl equation by adding $\nabla\nabla\cdot$ to the equations in the air, and $\sigma\nabla\nabla\cdot\sigma$ in the Earth. These modifications effectively enforce appropriate divergence-free conditions. Testing this scheme and comparing to alternative approaches such as the divergence correction required less than one day, using our object oriented matlab codes.

A final example is provided by development of a code for modelling EM fields due to motional induction by ocean tides, a student project. Results from modelling the M2 tide at global and regional scales are shown in Fig. 5. Both global and regional spherical coordinate modelling codes were used, with sources derived from a modern data-constrained model of ocean tidal current (Egbert and Erofeeva, 2002). The global solution was used to provide boundary data for the regional solution, which is for an area off the northwest coast of the US, using a conductivity model derived from an array of MT data.

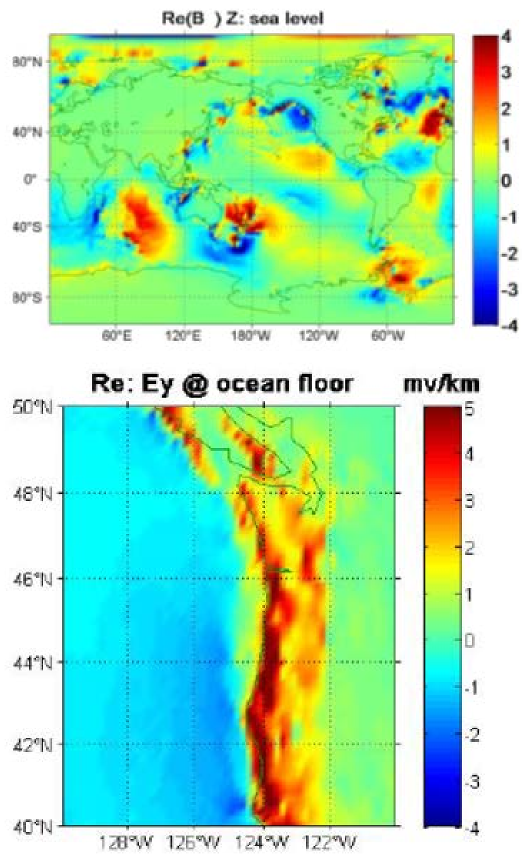

Figure 5. Tidal EM fields obtained by 3D modelling of with tidal motional induction sources. The upper panel gives magnetic fields at the surface, derived from a global spherical solution. The lower panel gives electric fields at the seafloor/land surface derived from a regional spherical coordinate solution, with boundary data derived from the global solution.

ACKNOWLEDGMENTS

This work has been partially supported by NSF grant EAR-125496 TO GDE.

REFERENCES

Cherevatova, M., Egbert, G.D., Smirnov, M. Yu., & M. Becken (2017). 3D MT and CSEM Modelling with Multi-Resolution Grid. 6^{th} International Symposium on Three-Dimensional Electromagnetics, Berkeley, CA.

Haber, E., and Heldmann, S. (2007). An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients, Journal of Computational Physics, **223**, 2, 783-796.

Egbert. G.D. & S. Yu. Erofeeva (2002). Efficient inverse modeling of barotropic ocean tides, *J. Atmos. Ocean. Tech.*, 19, 183-204.

Egbert, G. D. and Kelbert, A. (2012). Computational Recipes for Electromagnetic Inverse Problems, *Geophys. J. Int.*, **189**: 251–267. doi: 10.1111/j.1365-246X.2011.05347.x

Kelbert, A., N. Meqbel, G. D. Egbert, and K. Tandon (2014). ModEM: A modular system for inversion of electromagnetic geophysical data, *Comput. Geosci.*, *66*, 40–53, doi:10.1016/j.cageo.2014.01.010.

.