Speeding up the inversion of the 3D MT problem

D. Varılsüha[1]
[1]Istanbul Technical University, deniz.varilsuha@itu.edu.tr

SUMMARY

In this study, I have investigated different algorithms that can greatly reduce the inversion time of a 3D Magnetotelluric problem. These algorithms include the right choice of the partial differential equations to solve an EM problem, a hybrid finite-difference and finite-elements numerical technique, employment of multiple GPUs for the calculations, the mesh decoupling technique, and other good implementation practices. When all of them are applied in conjunction, the inversion time for a medium-sized 3D problem can be completed in around an hour.

Keywords: 3D, Magnetotellurics, GPU computing, Hybrid Numerical Technique.

Introduction

Any 3D interpretation or inversion of geophysical data is computationally challenging. Researchers who develop such software have to deal with long compute times that can vary from hours to days to obtain reasonable 3D models. In reality, an inversion tool should be rerun several times with different parameters to be sure of the final results. This may even lead to greater compute time and also it limits the possibility of one experimenting with the existing code. In this study, the 3D MT inversion problem is investigated and different algorithms are applied to the forward and inverse parts of the problem to speed up the inversion.

Methods

For the MT problem, either second-order E field or H fields obtained from Maxwell equations are solved in quasi-static approximation for the forward problem. The resulting linear system can be solved either by direct methods such as LU decomposition or iterative methods such as conjugate gradients (CG). The latter approach is better and more efficient for larger systems. However, the iterative solution's convergence might be affected by round-off errors and it may need an additional routine called divergence correction of E fields for low frequencies as the linear system gets more and more ill-conditioned when the frequency value gets closer to zero. For that reason, there are many different partial differential equations (PDE) to solve for various reasons to obtain the same fields. Varılsüha and Candansayar (2018) investigated these PDEs that can be solved iteratively in the least amount of time and the following is found to be the fastest when compared to others.

$$\nabla x \nabla x \mathbf{A} + \mu_0 \sigma (i\omega \mathbf{A} + \nabla \psi) = 0 \quad (1)$$

$$\mu_0 \nabla . \sigma \mathbf{A} + \frac{\mu_0}{i\omega} \nabla . \sigma \nabla \psi = 0. \quad (2)$$

E-fields are calculated via

$$\mathbf{E} = -i\omega \mathbf{A} - \nabla \psi \quad (3)$$

The iterative solution of the coupled system of (1) and (2) reduce the number of iterations required by half on average when compared to the classical E-field formulation.

The most common numerical techniques to solve a given problem are finite-differences (FD) and finite-elements (FE) approaches. When a model has a flat surface, it is a common approach to use simple blocks and employ the FD technique due to its simplicity. The FE technique is necessary when distorted hexahedral elements or tetrahedral elements are present. Implementation-wise, it can be a good approach to start with a structured and non-distorted mesh. Later topographic features such as hills and valleys can be added by distorting the mesh. This will result in a partially distorted mesh and the suggested technique by Varılsüha (2019) can be used to take advantage of both FE and FD techniques by employing both of them in a single mesh.

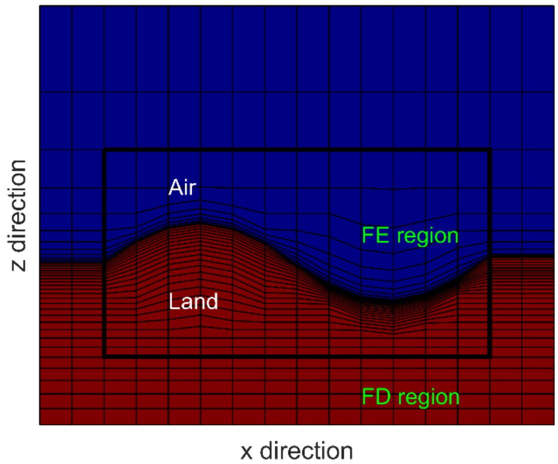

Figure 1. A depiction of the hybrid numerical technique is given for a 2D mesh. While the distorted FE region is solved with finite elements, the undistorted blocks are solved with finite-difference methods

The main advantage of this hybrid technique is its computational cost. The linear systems resulting from the pure FD technique can be solved with half of the time required by the FE technique. It also requires one-third of the memory required by the FE method. The meshes formed and used in this study have only 5 to 20 percent of their cells distorted and thus require the FE method. So, this ultimately means that the hybrid method is almost as efficient as the FD approach while being able to represent the topographic features for the model used in this study. It is also worthy to state it reduces the required memory space which could be very important when GPUs are used to solve the equations and considering their limited amount of memory space.

It is the most straightforward approach to use a single mesh for all data, frequencies, and receivers. It is also a common approach to use multiple smaller meshes for different receivers to reduce computational overburden, especially for airborne EM and time domain EM studies. It can be possible to employ this technique for the frequency domain MT problem. Varılsüha (2020) showed that different meshes can be created for different frequency groups since higher and lower frequencies have different penetration depths. In this study, the same technique is employed and smaller meshes are later mapped to the parameter mesh with a mapping matrix. Figure 2 depicts a picture of a parameter mesh and two local meshes for a high-frequency group and a low-frequency group. By using different smaller meshes for different frequencies, as opposed to using a single large mesh, it was possible to at least halve the total number of cells for the forward modeling meshes used in this study. This ultimately means halving the time for the forward modeling and thus inversion time.

High performance and parallel computing can also reduce computational time. It is common to use multi-core CPUs or multiple CPUs in a workstation to divide up the work using libraries such as MPI or OpenMP. In the last decade, GPU computing has become popular and the platform itself has shown to have the potential to speed up numerical computations. It is also possible to plug in multiple GPUs to a single computer to speed-up calculations even more. This could be also an important feature because one doesn't need to deal with large and complicated shared or distributed memory clusters since it is easy to plug in more GPUs to a single computer and it is easy to configure the new hardware. For these reasons, in this work, I have used 8 GPUs (Nvidia RTX 3090) along with a 10-core CPU (Intel 10900K) to do the numerical crunching using Matlab. The experimentation showed that using 8-GPUs in parallel speed-up the computations by 8-fold when compared with parallel computing using the multi-core CPU.

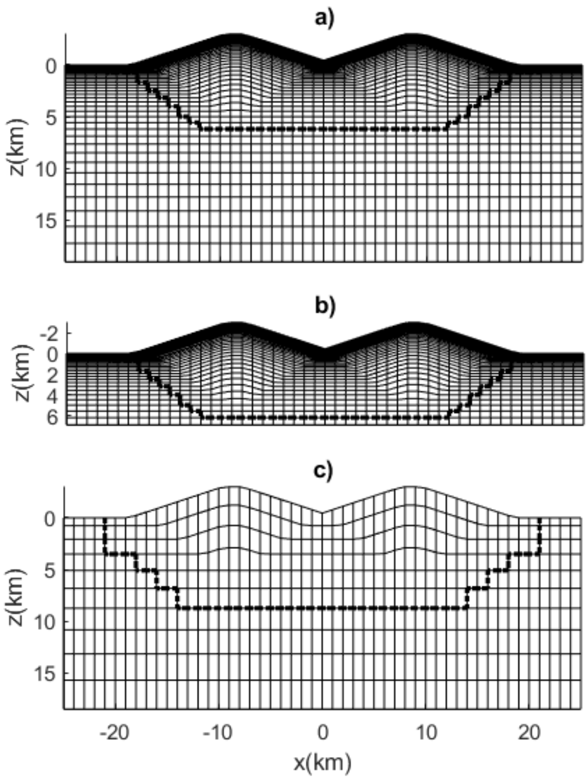

Figure 2. a) The parameter mesh and the forward modeling mesh without the mesh decoupling technique. b) mesh for the highest frequency group and c) mesh for the lowest frequency group when the mesh decoupling technique is employed. The dashed line shows the barrier between FE and FD regions for the hybrid numerical technique.

A regular consumer-grade CPU could have a core count between 1 and 64. On the other hand, a consumer-grade GPU might have a core count of 10000. In addition to that, GPUs have their dedicated memory and the memory bandwidth on the GPU can exceed the bandwidth between CPU and RAM by 10 to 20 times. This is an important detail because the linear algebra for sparse matrices is often memory-bounded as opposed to being compute-intensive. This means higher memory bandwidth of a system can greatly reduce the computation times. For that reason, multiple GPUs are employed to solve the linear system iteratively using a Conjugate gradient (CG) method in this study to achieve significant speed-ups. However, to make the algorithm converge to a solution in a fast manner, the CG system must be preconditioned using a preconditioner matrix. Usually, the preconditioner matrix is decomposed into two triangular matrices using the incomplete lower-upper algorithm (ILU(0)) and it is applied to a given vector in every step of the iterative solver by forward and backward sweeping. This operation can be challenging for GPUs because sometimes it can be difficult to find parallelism for a given preconditioner matrix. My experimentation also confirmed that 80 to 90 percent of GPU time is spent doing the triangular matrix solutions. I also noticed that solving larger linear systems makes the GPU more efficient, however, this ultimately means larger models and longer inversion times. To make the computations more efficient, I came up with an implementation detail that resulted in a 15-20 percent decrease in forward modeling time. For the 3D MT problem, a linear system such as $Ax = b$ is solved for two polarizations. Instead of solving it twice for different RHS vectors, I've found that solving the following is more efficient,

$$\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \qquad (4)$$

where x_1 and x_2 are the solution for two polarizations. This technique is implemented in the inversion algorithm fully.

To test all the techniques presented in this work all at once, a previously published dataset that would require a medium-sized model is chosen. This dataset is also inverted by other people so it was possible to compare the inversion results to test the accuracy of this algorithm. The dataset named two mountain model's data is first created and used by Usui (2015) and I obtained it by contacting the author. It has 40 receiver points and 16 frequencies. Every receiver has 4 components of the impedance tensor and 2 components of Magnetic Transfer Function (MTF) for all frequencies. The data also has additional noise and real-valued distortion components to make it harder to invert. In this study, I inverted the data to obtain the original model and the distortion tensors as parameters. In Figure 3 the true model that the data is obtained from is given.

The inversion is carried out by minimizing the following objective functional by the L-BFGS algorithm.

$$\phi(\mathbf{m_\sigma}, \mathbf{m_C}) = \phi_d(\mathbf{m_\sigma}, \mathbf{m_C}) + \lambda\phi_m(\mathbf{m_\sigma}) + \kappa\phi_C(\mathbf{m_C}) \qquad (5)$$

where ϕ_d , ϕ_m and ϕ_C are data misfit, model roughness, and distortion intensity. λ and κ are trade-off parameters that would be adjusted as the inversion is carried out. $\mathbf{m_\sigma}$ and $\mathbf{m_C}$ are model parameters for the conductivity and the distortion tensor that are being obtained in inversion.

In Figure 4, it can be seen the inversion results are side by side with the true model. In this inversion, the impedance tensor (**Z**) and the MTF vector (**W**) are used as data. In addition to those, the distortion tensor for both **Z** and **W** is estimated as a parameter to obtain better models. Distorted tensors could be expressed as

$$\mathbf{Z}_D = \mathbf{C}_h\mathbf{Z} \qquad (6)$$

$$\mathbf{W}_D = (\mathbf{W} + \mathbf{C}_z\mathbf{Z}) \qquad (7)$$

$\mathbf{Z}_D$ and $\mathbf{W}_D$ are the distorted forms of data. $\mathbf{C}_h$ and $\mathbf{C}_z$ are the horizontal and vertical distortion tensor and vectors respectively. In this study, they are estimated as a parameter to obtain much correct inversion.

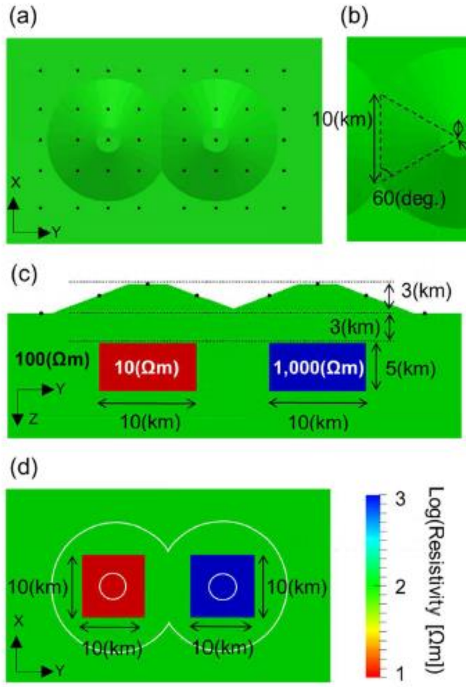

Figure 3. The figure is taken from the work by Usui, 2015. a) and b) shows the surface topography in detail. c) and d) show the exact spot and sizes of conductive and resistive anomalies. In this study, x and y directions are reversed and named y and x directions respectively.

Results

It took 45 inversion steps to obtain the results given in Figure 4. The location of both conductive and resistive anomalies are found correctly however the resistivity value of the resistive structure is estimated at 300 Ωm which is far away from its true value. However, considering the nature of the MT method and its insensitivity to resistive structures, these findings can be considered normal. The same inversion is also carried out by using only the impedance data (while estimating the distortion tensor also) and a very similar model is obtained from it. However, it took 52 steps to complete the inversion and this ultimately means that the joint inversion solves the problem in a lesser amount of time. It should be noted that the joint inversion of different data types can potentially reduce the computation times significantly.

The model given in Figure 4 had 116K parameters and 50 31 and 75 cells in x, y, and z directions respectively. Using all techniques described in this study helped to reduce the total inversion time to an hour. Even though the inversion code requires special hardware such as GPUs to run, the algorithms described here can be coded for any architecture and run on any hardware. These results also prove that the same techniques could be applied to other EM problems or other elliptic problems to achieve similar results.

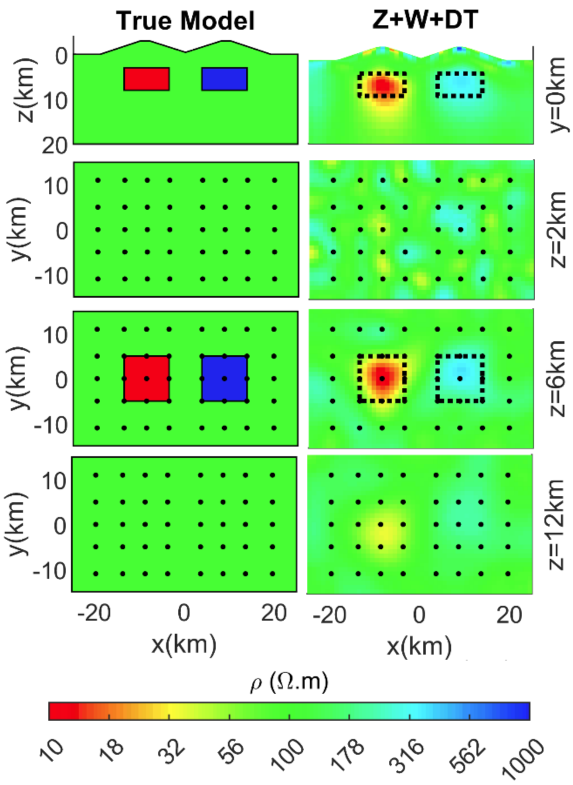

Figure 4. The true model and the model obtained by the joint inversion of impedance tensor and MTF vector. The dashed lines indicate the position of true anomalies. The inversion is performed by jointly inverting the impedance (**Z**) and MTF (**W**) data while estimating the distortion tensor (**DT**)

Discussion

When dealing with anisotropic forward modeling for an EM problem, the Lebedev grids can be employed especially when an FD grid is in mind with E fields defined on the edges. This not only helps with the numerical discretization of the anisotropic PDE but also helps to reduce the size of the mesh. The Lebedev grid technique requires 3 more meshes shifted to the original mesh so the total computational time seems to get quadrupled however at the same time it significantly increases the accuracy of the forward solution. This can also mean that a coarser mesh can be adapted and the same accuracy of a single mesh could be achieved. The important detail about the Lebedev grids is that they will generate 4 independent meshes for the isotropic case instead of one. When coupled with the idea given in Equation 4, it will make the forward solution on a GPU even more efficient. I have experimented with this idea and a model with no topography, and I've seen a further increase in efficiency. However, due to the topography of the model used in this study, it is not considered here.

Conclusions

There is no single answer when it comes to speeding up a geophysical inverse problem. In this study, I show that various techniques must be implemented together to achieve significant results. Simple parallelization techniques or better/faster hardware alone may not be enough. A correct choice of the PDE, hybrid numerical techniques, the mesh decoupling approaches also play a great role to achieve exceptional timings because every technique shown in this study has the potential to reduce the computational time significantly however when all of them are employed together the final speed-up is the multiplication of the speed-ups of all of them. This study concludes that these techniques can reduce the total inversion time by almost 2 orders of magnitude.

References

Usui Y (2015) 3-D inversion of magnetotelluric data using unstructured tetrahedral elements: applicability to data affected by topography, Geophysical Journal International 202, 828-849.

Varılsüha D (2019) Fast Three Dimensional Inversion of Magnetotelluric data based on Finite Difference and Finite Element Hybrid Forward Modeling, Ph.D. thesis, Ankara University, Ankara.

Varılsüha D (2020) 3D inversion of magnetotelluric data by using a hybrid forward-modeling approach and mesh decoupling, Geophysics 85, E191-E205.

Varılsüha D, Candansayar ME (2018) 3D magnetotelluric modeling by using finite-difference method: Comparison study of different forward modeling approaches, Geophysics 83, WB51-WB60.