# Comparison of Time-Stepping methods for late times EM diffusion phenomenons in Geophysics

## Math 607E: Course Project

Thibaut Astic
90142150
thast@eos.ubc.ca

## Abstract

In mineral resources exploration, Time-Domain Electromagnetic (TDEM) data are acquired to image conductive structures in the subsurface. A widely used setup measures the response to a step-off electric current in an inductive loop in term either of the magnetic flux density $\mathbf{b}_z$ of its first derivative $\frac{\partial \mathbf{b}_z}{\partial t}$ , with the loop and receiver either in the air or at the surface. The receiver is generally placed at the center of the loop to avoid side effects, as described in [13]. Electromagnetic phenomenons are described by Maxwell's equations, whose forms a complex system of partial differential equations linking fields and fluxes. We use a finite volume discretization to solve these equations, because of its conservation of mass property. At $t \leq 0$, the fields are static and can be computed in various ways. We discuss how diverse initializations can lead to different accuracy behaviors later in time. At $t > 0$, Maxwell's equations are well-known to be very stiff ([3]), requiring the use of implicit time-stepping methods. Moreover, the physical system displays wave behaviors at very early time channels we do not want to compute for later diffusive time windows. We test different time-stepping methods and compare their accuracy to an exact analytic solution. Backward Euler has appeared to be very reliable and stable but can require a lot of steps to give an accurate solution at late times. We encounter some issues to do a proper convergence analysis of BDF-2. Its behavior has proven to be highly sensitive to our mesh. However it has consistently displays better accuracy at late times than Backward Euler and in a lower number of time-steps for a comparable cost. BDF-4 has been briefly tested but the lose of A-stability seems to add to the previous problems encounter in BDF-2.

## 1. Introduction

Characterizing the Earth's subsurface is one of the key questions in Earth Sciences, either to image the Earth's core, anticipating pollution migration, finding minerals resources etc. Geophysics goal is to image these structures remotely by measuring Earth's response to different fields.

One of the main group of methods uses in Geophysics are Electromagnetic methods ([13]). With this project we are going to particularly focus on Time-Domain Electromagnetic (TDEM, figure 1) fields generated by inductive sources under diffusion assumptions. Electromagnetic phenomenons are governed by Maxwell's equations and hence are sensitive to contrasts in physical properties that can be interpreted in terms of geologic structure and fluid content. The diagnostic physical property for TDEM is electrical conductivity $\sigma$ (or its inverse resistivity $\rho$). In a nutshell, once we cut-off the current in the loop, eddy currents are generated according to Faraday's law in the ground to compensate the loss in the current loop and maintain the magnetic field. These currents are attenuated due to ohm's law, producing a change in the magnetic field, which results in induced eddy currents and so on. The final result is an downward and outward diffusion of rotational currents in the ground, commonly called 'smoke rings' ([9]). We can measure these changes in the magnetic field and then interpret them in term of conductivity contrasts in the subsurface (figure 2).

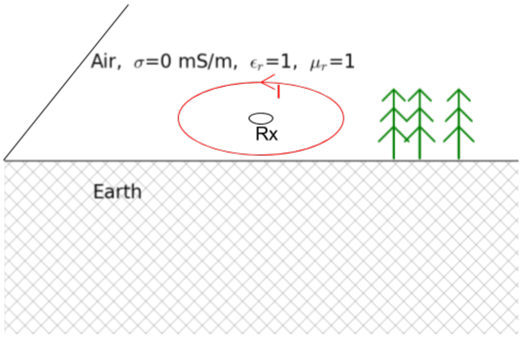

**Figure 1:** *TDEM system in a nutshell*

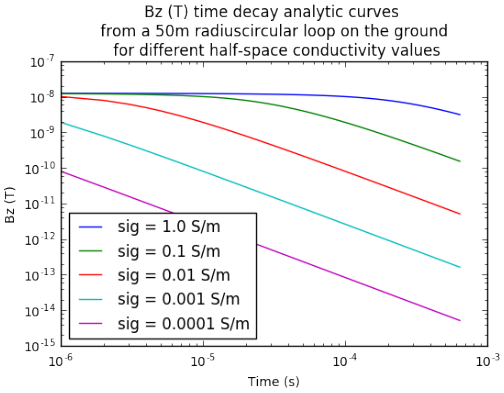

**Figure 2:** *Proof of concept: decay curves for various half-space conductivity*

Maxwell's equation forms a complex system of Partial Differential Equations. In most cases, there is no analytic solution except for very simple problems (half-space, sphere in void). To forward model the response of complex structures, numerical techniques are required. We present in this project a discretization in space of our TDEM problem using staggered grid and finite volume ([5]). Using this discretization, we have tested different time stepping methods (explicit and implicit, single and multi steps) in term of cost and accuracy to compute the late times corresponding to the diffusion phenomenons and this without computing the wave part of Maxwell's equation. We then investigate if computing late times without early diffusive times can be accurate with higher order methods.This project has been realized in Python and rely on the Python open source package SimPEG developed at UBC for the mesh and matrices operators ([1]).

Code and scripts to re-create most of the results generated for this project are available through Github at the address: https://github.com/thast/Math607E

## 2. Time Domain Discretization of Maxwell's Equation Using a finite volume E-B formulation

We start by writing Faraday's law and Ampere's law:

$$\frac{\partial \mathbf{b}}{\partial t} + \nabla \times \mathbf{e} = \mathbf{0} \tag{1}$$

$$\nabla \times \mu^{-1}\mathbf{b} - \sigma \mathbf{e} - \varepsilon_0 \frac{\partial \mathbf{e}}{\partial t} = \mathbf{j_s} + \mathbf{j_m} + \mathbf{j_p} \tag{2}$$

In our case we assume a magneto-quasi-static state. This means we are neglecting the electric displacement current $\varepsilon_0 \frac{\partial e}{\partial t}$ and therefore the electromagnetic waves due to the coupling of the magnetic induction with the displacement current in equation 2. We also ignore the magnetization current $j_m$ (ignoring magnetization of Earth's material) and polarization current $j_p$ (ignoring any polarization of Earth's material), only keeping our source term $j_s$. The usual diagnostic property for TDEM systems is the electric conductivity $\sigma$, as it usually varies a lot more than the magnetic permeability $\mu$, our model will assume that $\mu = \mu_0$ everywhere.

$$\frac{\partial \mathbf{b}}{\partial t} + \nabla \times \mathbf{e} = \mathbf{0} \tag{3}$$

$$\nabla \times \mu_0^{-1} \mathbf{b} - \sigma \mathbf{e} = \mathbf{j_s} \tag{4}$$

In finite volume on a staggered grid, we assign different physical quantities to the different parts of the mesh: cell-centers, nodes, edges and faces. We choose here to discretize according to the so-called E-B formulation, in opposition with the other popular duo field-flux discretization, the H-J formulation ([7]). As shown in figure 3, we put on cell-centers physical quantities that are consider constant over each cell, here our physical properties. As a flux and our data, the magnetic flux density is discretized on the faces. This forces the electric field to be discretized on the edges, as related to B through a curl operator. This is a convenient choice to discretize the wire of our source loop which is likely to be smaller that our cell size.

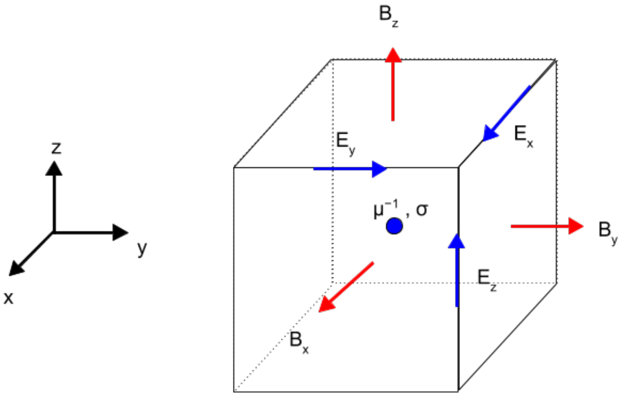

**Figure 3:** *EB Formulation: where do things live*

Reformulating equations 1 and 4 in weak form, we take the inner products of these equations with edges and faces spaces functions $\mathbf{w}$ and $\mathbf{f}$ respectively over the whole domain $\Omega$ and integrate by parts Ampere's law:

$$\int_\Omega (\nabla \times \mathbf{e}) \cdot \mathbf{f} \; dv + \int_\Omega \frac{\partial \mathbf{b}}{\partial t} \cdot f \; dv = 0 \tag{5}$$

$$\int_\Omega \mu_0^{-1} \mathbf{b} \cdot (\nabla \times \mathbf{w}) \; dv + \int_{\partial\Omega} \mu_0^{-1} \mathbf{w} \cdot (\mathbf{b} \times \hat{\mathbf{n}}) \; ds - \int_\Omega (\sigma \mathbf{e}) \cdot \mathbf{w} \; dv = \int_\Omega \mathbf{j_s} \cdot \mathbf{w} \; dv \tag{6}$$

By assuming the boundary condition for $\mathbf{b}$ to be $\mathbf{b} \times \hat{\mathbf{n}}|_{\partial\boldsymbol{\Omega}} = 0$, we cancel the boundary term. This also makes sense at a physical level as Gauss's law for magnetic flux density states that $\mathbf{b}$ is divergence free. Approximating our physical quantities and operators onto a tensor mesh, using vector identities and transposing the result, we obtain a discretize formulation in finite volume of our problem in matrix form (Note that the functions $\mathbf{f}$ and $\mathbf{w}$ nicely cancel in this form as they are arbitrary):

$$\cancel{f^T} \frac{\partial \mathbf{b}}{\partial t} + \cancel{f^T} \mathbf{Curl\, e} = 0 \tag{7}$$

$$\cancel{w^T} \mathbf{Curl}^{\mathbf{T}} \mathbf{M}^{\mathbf{f}}_{\mu^{-1}} \mathbf{b} - \cancel{w^T} \mathbf{M}^{\mathbf{e}}_{\sigma} \mathbf{e} = \cancel{w^T} \mathbf{M_e j_s} \tag{8}$$

where the matricies $\mathbf{M^f_{\mu^{-1}}}$, $\mathbf{M^e_\sigma}$ and $\mathbf{M_e}$ are defined as (with the appropriate averaging matrices $\mathbf{A}$ between the different parts of the mesh):

$$\mathbf{M^f_{\mu^{-1}}} = \mathbf{diag}[\mathbf{A^T_{fc}}(\mathbf{v} \odot \boldsymbol{\mu_0^{-1}})];\ \mathbf{M^e_\sigma} = \mathbf{diag}[\mathbf{A^T_{ec}}(\mathbf{v} \odot \boldsymbol{\sigma})];\ \mathbf{M_e} = \mathbf{diag}[\mathbf{A^T_{ec}}\mathbf{v}] \tag{9}$$

We now have the choice to either express our problem in term of the field $\mathbf{e}$ or either in term of the flux $\mathbf{b}$. We choose here to solve for $\mathbf{b}$, as it is easy to initialize thanks to Biot-Savart's law as we will see in the next section. Expressing $\mathbf{e}$ in term of $\mathbf{b}$ we get:

$$\mathbf{e} = M_\sigma^{e-1} Curl^T M^f_{\mu^{-1}} \mathbf{b} - Curl M_\sigma^{e-1} M_e \mathbf{j_s} \tag{10}$$

And finally we end with a single linear partial differential equation for $\mathbf{b}$:

$$\frac{\partial \mathbf{b}}{\partial t} = -Curl\ M_\sigma^{e-1} Curl^T M^f_{\mu^{-1}} \mathbf{b} - M_\sigma^{e-1} M_e \mathbf{j_s} \tag{11}$$

In our code, the tensor mesh and associated matrices are generated thanks to SimPEG ([1]).

The most common type of data for this kind of system is $\frac{\partial \mathbf{b}_z}{\partial t}$ and more rarely $\mathbf{b}_z$. However it is pretty straightforward to see that, once the PDE system solved for $\mathbf{b}$, $\frac{\partial \mathbf{b}_z}{\partial t}$ can be easily obtained by computing $\mathbf{e}$ using the equation 10 and then taking the Curl of $\mathbf{e}$ gives us $-\frac{\partial \mathbf{b}_z}{\partial t}$ by Faraday's law (equation 1). We have choose here to focus on the magnetic flux density $\mathbf{b}$ to compare our time-stepping methods, as analytic solutions exists both for $\frac{\partial \mathbf{b}_z}{\partial t}$ and $\mathbf{b}_z$, in the intend to speed up a bit the already computationally expensive process involved in the experiments.

## 3. Initializing the magnetic flux density b

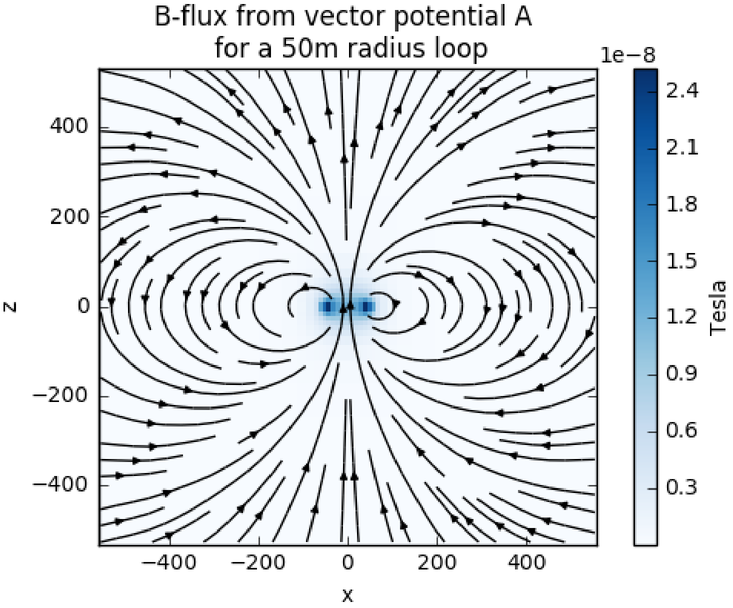

**Figure 4:** *Static B flux*

We now need to define our initial condition for $\mathbf{b}$. Here we are mostly interested to the response to a step-off function. We can then assume a magneto-static case at $t = 0$ (see figure 4). Looking at the Ampere's equation (2), the magneto-static field generated by an inductive loop can be written as:

$$\nabla \times \mu_0^{-1}\mathbf{b} = \mathbf{j_s} \tag{12}$$

The initialization for an inductive source is not dependent of the conductivity structures of the ground. Note this is not true for a galvanic source as in this case we would need first to solve the DC problem to get our source current term [5]. Also note that this would not be true either if we consider contrasts of magnetic permeability as this time we would need to formulate the problem using a fictitious source [5].

Several options are now possible to solve the magneto-static problem for an inductive source. First there exists analytic solutions for $\mathbf{b}$ for a circular loop both for the magneto-static case and for the response to a step-off for an half-space. One might thinks this would be a wise choice here, since our goal is to implement different time-stepping methods, we need to compare them with an exact solution. However, if the exact continuous solution for $\mathbf{b}$ is divergence free as required, this is unlikely to be true for the discretized version on our domain. This will cause some major accuracy issues as presented in figure 5, where we computed the response to a step-off of an half-space with our two types of initialization (script: TDEM_initialization.py). To address that problem, since $\nabla \cdot \mathbf{b} = 0$ by Gauss's law, we can express it in terms of a vector potential $\mathbf{a}$:

$$\mathbf{b} = \nabla \times \mathbf{a} \tag{13}$$

The vector potential $\mathbf{a}$ is now a quantity that lives on the edges. It exists, as for $\mathbf{b}$, an analytic solution for a circular loop. We can then compute it on every edges and use our discretized operator Curl to obtain a divergence-free discretized magnetic flux density $\mathbf{b}$. However our initialization is no longer independent of our mesh and will have a quadratic convergence compared to the size of the cells (figures 6 & 7), even if we use an analytic formulation for $\mathbf{a}$ (Script: Test_convergence_Initialization.py in the repository).

For any type of loop or even current distribution, it is also possible to initialize using an equivalent of the Biot-Savart's law for the vector potential $\mathbf{a}$. This has also been implemented and will be useful for more general cases, as for example for square loops that are widely used on field.

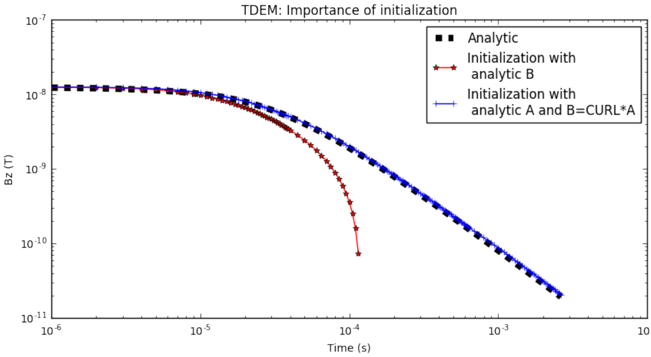

**Figure 5:** *Accuracy with* $\boldsymbol{a}$ *or* $\boldsymbol{b}$ *initialization for an half-space, all others parameters similar otherwise*

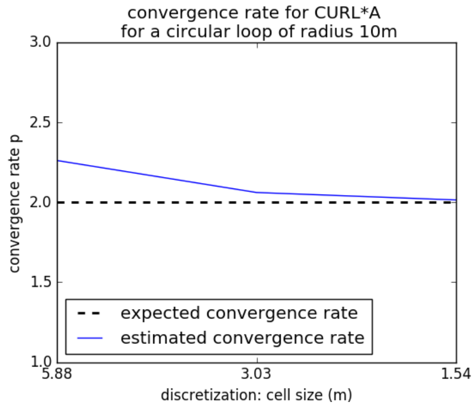

**Figure 6:** *Initialization: Approximated convergence rate*

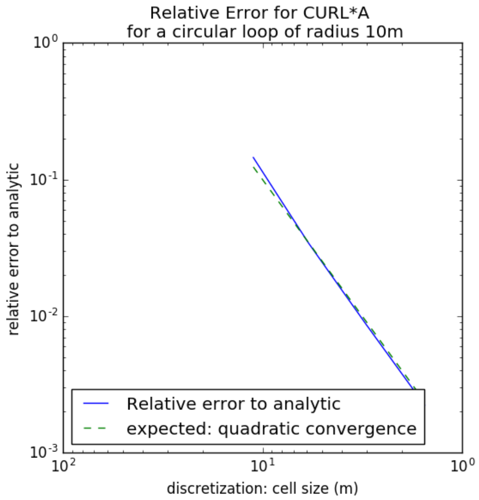

**Figure 7:** *Initialization: Relative error for a 10m radius loop*

# 4. Meshing

A crucial part of our problem (and still an open question for many aspects of it) is: how do we mesh? We are facing here several imperatives. First, we should discretize enough the loop and its interior such as our initial value is the best possible. Second, as the later we go in time, the further away the 'smoke rings' are going and so should expand our mesh. Our cells should also be small enough to account update between our smaller expected time steps (at earliest times). Finally, our mesh should include the air as part of the model to avoid boundary conditions issues. This will be the case even when we refer to our model as an half-space (meant as an half-space homogeneous Earth).

These contradictory imperatives, small cell sizes on a large scales, have lead several authors to use specialized meshes such as Octree Meshes ([4]) for large scale problems or cylindrical meshes ([6]) to deal with layered Earth models or centered anomalies. For simplicity, we have here decided to only use a regular tensor mesh whose cells expand the further away from the core. This will nonetheless limit some of our abilities to test the mesh as the problem can become computationally expensive very quickly.

Telford and al. ([12]) derive some formulas such as the speed of diffusion or diffusion distance (equivalent to the skin depth concept for plane wave) that can help to make decisions depending on the problem.

# 5. Time Stepping

## 5.1 Quick stability analysis and stiffness

Our initializations done, we can now begin to compute the response in time to the step-off. As the term $\mathbf{j}_s$ disappears, we now have:

$$\frac{\partial \mathbf{b}}{\partial t} = -Curl\ M_{\sigma}^{e-1} Curl^T M_{\mu^{-1}}^{f} \mathbf{b} \tag{14}$$

$$\frac{\partial \mathbf{b}}{\partial t} = L\mathbf{b} \tag{15}$$

Because $\nabla \cdot \mathbf{b} = 0$, equation 14 is very similar to the heat equation. In a full homogeneous space, $L = \sigma^{-1}\mu^{-1}\Delta$ and we would be able to write:

$$\sigma\mu \frac{\partial \mathbf{b}}{\partial t} = \Delta \mathbf{b} \tag{16}$$

A quick absolute stability analysis in 1D assuming an uniform cell size $h$ gives that we need a time-step $k$ such that:

$$k \leq \frac{\sigma \mu h^2}{2} \tag{17}$$

However, for most ground material, $\sigma\mu \simeq 10^{-8}$ (and $\sigma\mu \simeq 10^{-15}$ for the air),which make our problem very stiff. In our case, for diffusion phenomenons we are usually interested to a wide time window between $10^{-6}s$ and $10^{-1}s$ and our space discretization is in the order of $\simeq 10m$. Moreover, for $t \leq 10^{-7}$, our magneto-quasi-static assumption will not hold anymore and we would need to consider the wave behavior of Maxwell's equations. This make explicit method inadequate for most practical applications in our case. We thus decide to focus on implicit methods.

Our matrix operator $\mathbf{L}$ can be considered as a model-weighted Laplacian. As all our physical properties are positive, we are assure that all $\mathbf{L}$'s eigenvalues are non-positive. This makes the use of A-stable implicit methods even more recommended.

We particularly focus here on the family of BDF methods, as they have a very stiff decay and thus damp quickly the high frequencies of the error due to the magneto-quasi-static approximation ([3]).

## 5.2 Backward Euler

The first of all these implicit methods is the well-known and widely used Backward Euler.

$$\mathbf{b^{n+1}} = (I - kL)^{-1}\mathbf{b^{n}} \tag{18}$$

One drawback of implicit methods is they require to solve a linear system. By chance, the matrix is symmetric positive definite, which open a lot of possibilities for the choice of solver. We choose to use a direct solver designed to handle large sparse matrices, Pardiso ([8],[10],[11]), as we care here about the precision of our solution. It also keeps into memory the factorization, which speeds up the process as long as we keep the same time step. A conjugate gradient approach would have been another option from the side of iterative optimization methods.

One advantage of Backward Euler, and all the family of BDF methods, is that it damps the high frequencies. Also as this method is A-stable, we do not have to worry about stability.

This method has proven to be relatively resilient in term of meshing. Unadapted meshes did not seem to affect it as much as higher-order methods (see **5.3 BDF-2** section). We ran a convergence analysis with a loop at the surface of an half-space of conductivity $10^{-1}S/m$. The initial value is computed with a relative error of 0.37% and we are focusing here on the accuracy for the mid-to-late time $10^{-4}s$, depending on how many time steps we are using to reach it (Script: Test_convergence_BackwardEuler_TS.py). The convergence to the analytic solution is linear as expected (figure 8). However as we can see on figure 9, with a regular time-step size, we still does not have an 99% accuracy with 50 time steps. To diminish the number of time-steps, we can increase the time-step size at later times but we then have to re-factorize a new matrix to solve our linear system and similar accuracy is not ensured.

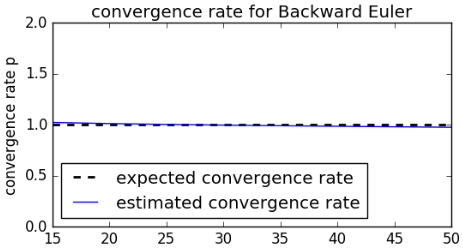

**Figure 8:** *BE: Approximated convergence rate*

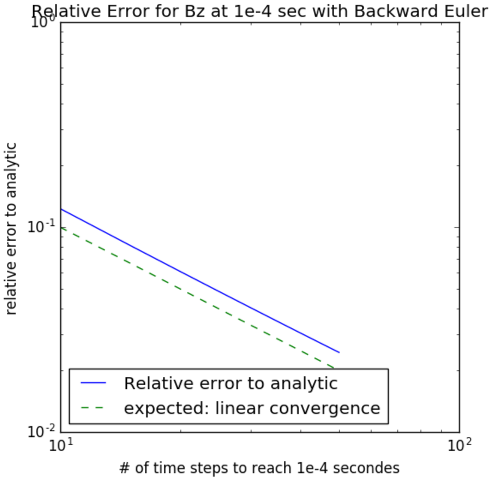

**Figure 9:** *BE: Relative error*

## 5.3 BDF-2

We then use a BDF-2 scheme. This scheme first seemed appropriate as it has the same ability as backward Euler to damp high frequency modes and is an implicit scheme. It is using a polynomial interpolation between the different calculated states, which seemed reasonable as we are only observing the secondary field after the step-off (no source), which are known to be very smooth (see Maxwell's equations 1 & 2 without source term).

$$b^{n+1} = (I - \frac{2}{3}kL)^{-1}(\frac{4}{3}b^n - \frac{1}{3}b^{n-1}) \tag{19}$$

We admit here our incapacity to perform a proper convergence analysis of this scheme (script: Test_convergence_BDF2_TS.py). As it can be seen in figure 10, our scheme as here converge a lot more faster than expected. This has proven to be highly sensitive to our mesh, with sometimes convergence a bit less than expected for what we guess were unadapted meshes. However in any cases we have observed, it has consistently displays better accuracy at late times than Backward Euler and in a lower number of time-steps for a comparable cost (compute the factorization of a similar matrix and then solve a right-hand-side for each time-step). For the same problem than the one solved previously with Backward Euler, we see here our relative error has damped to less than 1% in less than 10 steps (> 50 for Backward Euler) and up to a similar relative error than our initial state in about 20 steps.

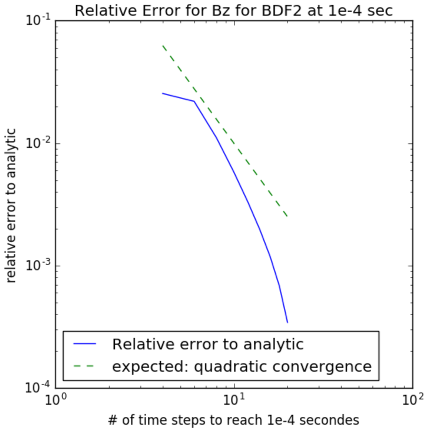

**Figure 10:** *Relative error for BDF2*

Our best assumptions so far to explain our incapacity to compute a rate of convergence of 2 for BDF-2 are:

- As it is using a polynomial interpolation, the points where we evaluate it are important (our mesh)
- We are converging too fast to a similar error than our initial state, not giving the time for the algorithm to stabilize
- our intuition that the secondary fields are smoothed enough to be approximated by a polynomial is wrong

Another huge drawback of BDF-2 is that it cannot be used in a multiple steps sizes schemes as the coefficients assume a regular sampling in time (at least not without modifications to the coefficients, which would lead to Modified Divided Differences ([2])). This can be problematic as usually data are recorded over several orders of magnitude of time (from $10^{-6}s$ to $10^{-1}s$). We would then recommend it as a fast way to get to late time and then switch to a Backward Euler scheme with various time-steps sizes.

## 5.4 BDF-4

We only briefly tackled this method due to the challenges faced obtaining a proper convergence analysis of BDF-2. From the few tests we made, we generally obtained a very accurate result, better than BDF-2, in only 4 of 5 steps for late time. However it displays a tendency to lose accuracy as we increased the number of steps. Our best assumption is that, as this implicit method is no longer A-stable, we might have hit the instability region at some of the iterations.

# 6. Synthetic Examples

## 6.1 Half-Space

Figure 11 and 12 shows the computed response of an half-space of $\sigma = 10^{-1}s$ to a circular loop of $50m$ radius for both Backward Euler and BDF-2 and the obtained relative errors. Backward Euler is using an increasing time step starting at $10^{-6}s$ and updated every 100 steps, while BDF-2 is using a constant step size of $10^{-5}s$ (script: TDEM_HalfSpace_example.py) For late time, BDF-2 is actually cheaper than Backward Euler, as we require to factorize a single matrix (for a single time step), whereas backward Euler need to re-factorize a new matrix each time we increase the time-step. We see that rapidly BDF-2 catch up and overcome Backward Euler for accuracy, even if it started later and with longer time steps. The loss of accuracy at later times for both methods can be explained by the size of the mesh as our eddy currents start being close to our boundaries. The bumps in Backward Euler are explained by the varying time-steps. However the 'double picks' seen in BDF-2 for mid-to-late times ($\simeq 10^{-4}s$)is not completely understood but seems related to our increasing cell-sizes in our mesh (too fast or too soon increase?).

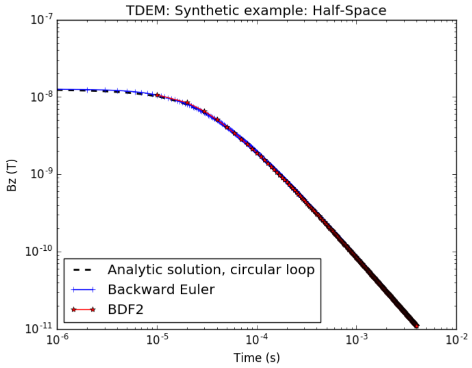

**Figure 11:** *Half-Space Response: BE and BDF-2*

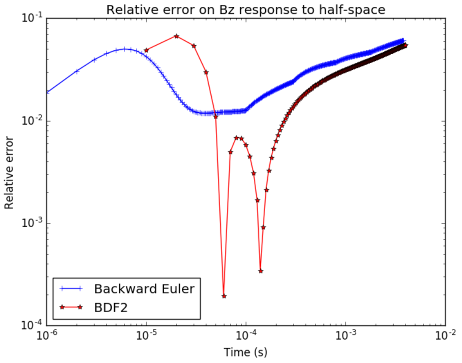

**Figure 12:** *Half-Space Response: BE and BDF-2 relative error*

## 6.2 Layered Earth

Finally, as a bonus to prove the abilities of our current code, consider a 2 layers Earth model (figure 13, the air is included for $z > 0m$) of different conductivities with a square inductive loop at the surface (figure 14). Initialization is performed using a Biot-Savart's law for the vector potential $\mathbf{a}$ and computing the Curl of it to obtain the magnetic flux density $\mathbf{b}_0 = Curl\ \mathbf{a}$ at $t = 0$.

We then use a Backward Euler scheme to compute the model response to a step-off (see figure 15) with a multi-step-sizes strategy. In dashed, we compares our solution to the analytic for 2 homogeneous Earth with the same conductivity of each layer respectively induced by a circular inductive loop of same magnetic dipole moment as our square loop. We see very well that at early times we are only seeing the first layer and at later time we get closer and closer to the response of an homogeneous Earth composed only of the second layer.

Figure 16 show the magnetic flux density at the last computed time. We can recognize well the anticipated diffusive 'smoke ring'.

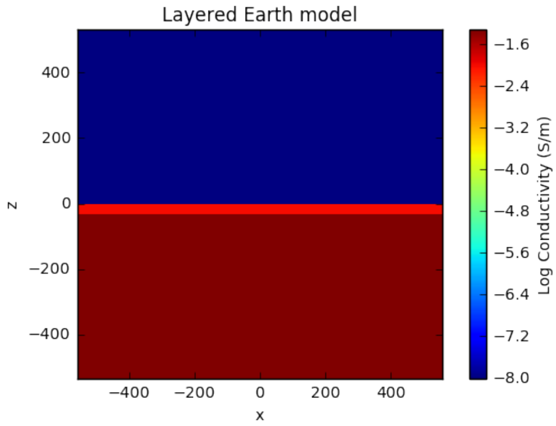

**Figure 13:** *Layered Earth Model*

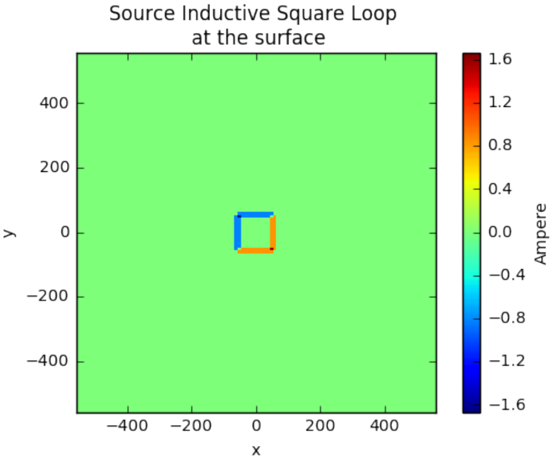

**Figure 14:** *B-flux at the final time*

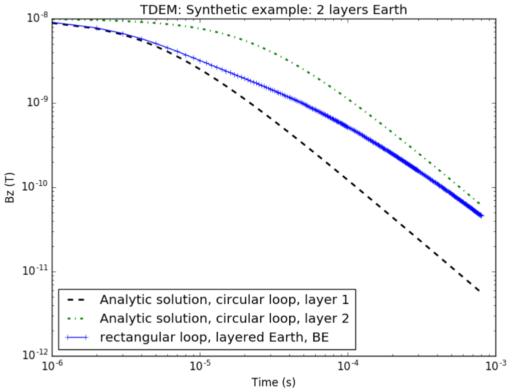

**Figure 15:** *2-Layers Earth Response: Backward Euler*

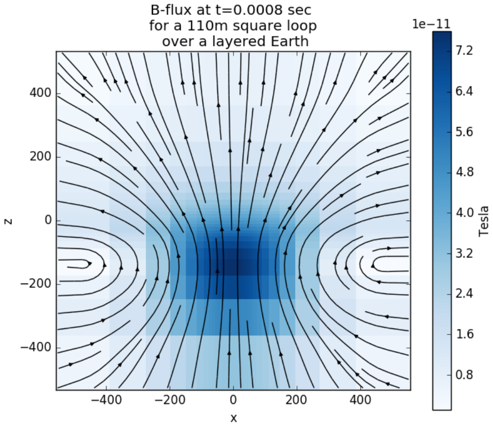

**Figure 16:** *B-flux at the final time*

# 7. Discussion and conclusion

We have built here a code that is able to forward model electromagnetic fields in time in response to a step-off for any discretized subsurface and survey configurations. Despite not addressed here, our code should also be able to deal with waveforms, as we are able to form our vector $\mathbf{j}_s$ and to include it in our matrices operators.

Initialization with a vector potential has been a key ingredient to the success of the code, as it ensures a divergence-free starting point.

Backward Euler is still the most widely used method in the domain for TDEM problems. It has proven, again, its reliability. BDF-2, despite failing to obtain a proper convergence analysis of our code, has given consistently more accurate results in less time steps and for a similar or even lower cost.

TDEM is a very stiff problem, with a lot of contradictory imperatives for the mesh that make the problem computationally expensive very quickly. Our time-steps, especially BDF-2, has shown unexpected behaviors associated with what we would guess being unadapted meshes for the problem. For further studies, it would be interesting to apply the same approach with a more efficient mesh in order to span more rigorously the space of variables (conductivity, cell size, mesh size...) and determine the key ingredients to an problem-adapted mesh that ensure a good behavior of the time-stepping methods.

Forward modeling is an important step of geophysics when it comes to invert data to recover a model. It is important to note that Backward Euler is usually easier to use as part of an inversion process. For late time interest, BDF-2 can be use in a strategic way combined to Backward Euler. It is in our opinion a very effective way to get close to the time windows of interest in few steps and we can then switch to a more appropriate time step size using Backward Euler.

# References

[1] Cockett, Rowan, Seogi Kang, Lindsey J. Heagy, Adam Pidlisecky, and Douglas W. Oldenburg (2015) "SimPEG: An Open Source Framework for Simulation and Gradient Based Parameter Estimation in Geophysical Applications." *Computers and Geosciences*, September 2015. doi:10.1016/j.cageo.2015.09.015.

[2] Krogh Fred T. (1974) "Changing stepsize in the integration of differential equations using modified divided differences" *Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations: 19,20 October 1972, The University of Texas at Austin* pages="22-71, isbn= 978-3-540-37911-9, doi=10.1007/BFb0066584, url=http://dx.doi.org/10.1007/BFb0066584

[3] Haber, E., U. Ascher, D. W. Oldenburg (2004). "Inversion of 3D electromagnetic data in frequency and time using an inexact all-at-once approach" *Geophysics*, vol. 69, 1216-1228.

[4] Haber, E., Heldmann S. (2007) "An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients" *Journal of Computational Physics*, Volume 223 Issue 2, May, 2007, Pages 783-796

[5] Haber E.(2014). "Computational Methods in Geophysical Electrodynamics"

[6] Heagy J. lindsey, Rowan Cockett, Seogi Kang, Gudni K. Rosenkjaer, Douglas W. Oldenburg (2016) "A framework for simulation and inversion in electromagnetics" *Geophysics*

[7] Heagy J. lindsey http://simpeg.xyz/journal/dad772ba49ad31ddf3a3f7944511682b

[8] Luisier M. , O. Schenk et.al. (2013) Fast Methods for Computing Selected Elements of the Green's Function in Massively Parallel Nanoelectronic Device Simulations Euro-Par 2013, LNCS 8097, F. Wolf, B. Mohr, and D. an Ney (Eds.), Springer-Verlag Berlin Heidelberg, pp. 533544, 2013.

[9] Nabighian, M., and Macnae, J. (1991) "Time domain electromagnetic prospecting methods, in Electromagnetic Methods in Applied Geophysics" Volume 2: Application, Part A, chapter six,M. N. Nabighian (ed.), Society of Exploration Geophysicists, Tulsa, OK.

[10] Schenk O., M. Bollhfer, and R. Rmer, On large-scale diagonalization techniques for the Anderson model of localization. Featured SIGEST paper in the SIAM Review selected "on the basis of its exceptional interest to the entire SIAM community". SIAM Review 50 (2008), pp. 91-112.

[11] Schenk O., A. Wchter, and M. Hagemann, Matching-based Preprocessing Algorithms to the Solution of Saddle-Point Problems in Large-Scale Nonconvex Interior-Point Optimization. Journal of Computational Optimization and Applications, pp. 321-341, Volume 36, Numbers 2-3 / April, 2007.

[12] Telford, W.M., Geldart, L.P. and Sheriff, R.E. (1990) Applied Geophysics, 2nd Edition: Cambridge University Press, 1990.

[13] S. H. Ward, C. W. Hohmann (1988). "Electromagnetic Theory for Geophysical Applications", in Nabighian, M. N., Ed., Electromagnetic Methods in Applied Geophysics: Society of Exploration Geophysics.