EOSC 555 – EM Forward modeling

Written by: Dominique Fournier

Student ID: 42255083

Project Report: A solution to the 3D MT forward problem

Introduction

Electromagnetic signal from natural sources has long been of interest to geoscientists. The low-frequency magnetotelluric signal has proven to be useful at imaging deep conductive structures. While the physics of electromagnetic wave propagation is well understood, its numerical implementation still remains challenging. This research project is an attempt at replicating the work done by Haber et al (1999) and Farquharson & Oldenburg (2002). I will summarize the steps required to solve the 3D forward problem and to compute impedances. The first section summarizes the EM theory needed to model the magnetotelluric response of a 3D earth. The second section presents the forward operators, as well as the implementation of adequate boundary conditions. The third section demonstrates the accuracy of the forward algorithm through a series of tests. The fourth and fifth sections analyse the solutions obtained with the algorithm.

Magnetotelluric forward problem

The theory behind the magnetotelluric experiment rests on Maxwell's four partial differential equations linking the electric and magnetic fields. In the quasi-static limits, with frequency between 1-100Hz, the contribution from the changing electric field can be neglected yielding in frequency domain:

$$i\omega\mu H = -\nabla x E \qquad (1)$$

$$\nabla x \mathrm{H} = -\sigma \mathrm{E} - \vec{s} \qquad (2)$$

For the MT problem the source **s** is outside the domain, and assumed to be a planar wave field of unknown straight. The usual strategy is to model for the ratio of electric and magnetic field instead of the field itself. The electric and magnetic fields are related by the impedance tensor. Because there are four unknown impedances and only two equations, we need to forward model the electric and magnetic field for to perpendicular polarization such as:

$$\begin{bmatrix} {E_x}^x & {E_x}^y \\ {E_y}^x & {E_y}^y \end{bmatrix} = \begin{bmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{bmatrix} \begin{bmatrix} {H_x}^x & {H_x}^y \\ {H_y}^x & {H_y}^y \end{bmatrix} \quad (3)$$

with the superscript defining the polarization of the field. Re-organising the equations to solve for the impedances yield the following four equations to be solved:

$$Z_{xx} = \frac{E_x{}^x H_y{}^y - E_x{}^y H_x{}^x}{H_y{}^y H_y{}^x - H_y{}^x H_x{}^y} \quad (4a)$$

$$Z_{xy} = \frac{E_x{}^y H_x{}^x - E_x{}^x H_x{}^y}{H_y{}^y H_y{}^x - H_y{}^x H_x{}^y} \quad (4b)$$

$$Z_{yx} = \frac{E_y{}^y H_y{}^x - E_y{}^x H_y{}^y}{H_x{}^y H_y{}^x - H_x{}^x H_y{}^y} \quad (4c)$$

$$Z_{yy} = \frac{E_y{}^x H_x{}^y - E_x{}^y H_x{}^x}{H_x{}^y H_y{}^x - H_x{}^x H_y{}^y} \quad (4d)$$

These equations are used to compute data during the inversion process, which won't be treated in this document. We can however use the impedances to compute an apparent resistivity, which can be used to test or solution.

We decompose the electric field into vector and scalar potentials as prescribed by Haber (2000):

$$\mathbf{E} = \mathbf{A} + \nabla\varphi \quad (5)$$

where **A** is a divergence free vector field and ϕ is a scalar potential function. Using (4) and re-writing (2) by substituting for H yield our new system of equation:

$$\begin{bmatrix} \nabla\mathrm{x}\nabla\mathrm{x} - i\omega\mu\sigma & -i\omega\mu\sigma\,\nabla \\ \nabla\cdot\,\sigma & \nabla\cdot\sigma\nabla \end{bmatrix} \begin{vmatrix} A \\ \varphi \end{vmatrix} = 0 \quad (6)$$

Because the vector field A is divergence free we can replace the (curl x curl) operator by the Laplacian. It turns out to be much easier to apply boundary conditions on the Laplacian then on the CURL. We need to apply Derichelet boundary conditions on A since the inducing field is set on the xy-plane (z=0).

We set the problem as a sum of a primary field associated with a uniform half-space conductivity model, and a secondary field generated by the anomalous conductivity model. The strategy is therefore to solve for the primary field, then use the information to solve for the secondary. This step is repeated for two perpendicular polarizations required by equation (3). In matrix form, the problem becomes:

$$(E_{primary}) \begin{bmatrix} \mathrm{L} - i\omega\mu\mathrm{S} & -i\omega\mu\mathrm{S}\,\nabla \\ \nabla\cdot\,\sigma & \nabla\cdot\sigma\nabla \end{bmatrix} \begin{vmatrix} A \\ \varphi \end{vmatrix} = 0 \quad (7)$$

$$(E_{secondary}) \begin{bmatrix} \mathrm{L} - i\omega\mu\mathrm{S} & -i\omega\mu\mathrm{S}\,\nabla \\ \nabla\cdot\,\mathrm{S} & \nabla\cdot\mathrm{S}\nabla \end{bmatrix} \begin{vmatrix} A \\ \varphi \end{vmatrix} = \begin{bmatrix} -i\omega\mu\Delta\mathrm{S}\ \mathrm{E_p} \\ -\nabla\cdot\Delta\mathrm{S}\ \mathrm{E}_p \end{bmatrix} \quad (8)$$

where L is the discretized Laplacian operator, S is a matrix with harmonically averaged conductivity, ∇ · is the divergence and ∇ is the gradient operator. I used a cell centered discretization with ϕ living in cell-center, E and A on cell-faces and H on edges.

We finally need to define boundary conditions as described by Farquharson (2002):

$$\frac{\partial \mathrm{A}^x}{\partial x} \; and \frac{\partial \mathrm{A}^y}{\partial x}\bigg|_{\partial \boldsymbol{\Omega}} = 0 \quad (9a)$$

$$\frac{\partial \mathrm{A}^x}{\partial x} \; and \frac{\partial \mathrm{A}^y}{\partial x}\bigg|_{\partial \boldsymbol{\Omega}} = 0 \quad (9b)$$

$$\frac{\partial \varphi}{\partial n}\bigg|_{\partial \boldsymbol{\Omega}} = 0 \quad (9c)$$

$$\frac{\partial \mathrm{A}^x}{\partial z} \; and \frac{\partial \mathrm{A}^y}{\partial z}\bigg|_{\partial \boldsymbol{\Omega}} = g(x, y, z) \quad (9d)$$

$$\Delta A^z|_{\partial \boldsymbol{\Omega}} = 0 \quad (9e)$$

Where **g**(x,y,z) is the inducing field. In our case **g** is a constant set at the surface changing depending on the polarization of the field. The following section describes each operator in details.

Forward operators

Solving the system of equations prescribed by (7) and (8) requires 4 operators: the Laplacian, divergence, gradient and averaging operators. Because the field is decomposed in a primary and secondary component, the Laplacian and gradient operators have two additional forms for different boundary conditions. Finally we need a curl operator to compute the magnetic fields from the electric field.

Divergence

The divergence operator is straight forward for the cell-centered discretization since no boundary conditions need to be implemented at the boundary. The operator acts from cell-faces to cell-centered. **(See Annex A).**

Gradient

The gradient operator acts from cell-centered to cell-faces. We therefore need to implement boundary conditions at the limits of the model. Equation (9c) requires that the gradient at boundaries vanishes. This is simply done by zeroing the first and last row of the operator before "kronning". **(See Annex B)**

Curl

The curl operator acts from cell-faces to cell-edges. We would therefore need to implement boundary conditions specified by (9).

I initially attempted to build equation (6) using the CURL operator, spending a fair amount of time trying to figure out Derichlet boundary conditions. Turns out that it was not necessary since I could use the Laplacian operator instead, or I can add padding cells around the model to make sure the field vanishes at the boundary. I still managed to build a BC matrix for Derichelet conditions as follow:

```
% Need to compute BC for Derichelet
e = @(n) ones(n, 1);
bcx = sparse(nx+1, 1); bcx([1,end]) = [-2 2];
bcy = sparse(ny+1, 1); bcy([1,end]) = [-2 2];
bcz = sparse(nz+1, 1); bcz([1,end]) = [-2 2];

% Get the x components of BC on edges
[Y,X,Z] = meshgrid (yf,xc,zf);
A_y = Ejf(X,Y,Z);

BCz = kron(bcz, kron( e(ny+1), e(nx)));

Hzy = BCz.*A_y(:) * dZ(1);


% Compute only face values for Dyz
[Y,X,Z] = meshgrid (yf,xc,zf);
A_z = Ekf(X,Y,Z);

BCy = kron(e(nz+1), kron( bcy, e(nx)));
Hyz = BCy.*A_z(:) * dY(1);

% Compute the total Hx component, minus sign on the Hyz like the curl.
Hx = Hzy - Hyz;
```

with the **BCz** matrix selects only the vectors fields required for each partial derivatives. This operation is repeated three times for each component, and the final boundary matrix is built by stacking them all together. **(See Annex C)**

Laplacian

The vector Laplacian is a second –order differential operator defined by a 7 point operator going from cell-face back to the cell face. **Annex D** displays the Laplacian in its algebraic and Matlab forms. For both the primary and secondary field, applying boundary conditions for the z-x and z-y faces of the model is straight forward. The x-y boundaries for the primary field are defined just like the 1D problem. Since we are not solving for the field itself but for the ratios, the upper boundary conditions can be chosen arbitrarily. I chose to set g(z=0) = 1000. At the limit of $L \rightarrow -\infty$, we should expect the field to vanish. This would require to extend the mesh beyond several skin depths. A different strategy is to use the analytical solution for the 1D case:

$$E = ce^{-i\sqrt{i\mu\omega\sigma}\,z}, \qquad E(z=0) = c \quad (10)$$

and:

$$E(z=-L) = i\sqrt{i\,\omega\mu\sigma}E - E$$

(See Annex D)

Test on operators

This section presents the tests done on the operators to verify the accuracy and stability of the forward modeling. The first test looks at the accuracy of the Laplacian operator as a function of discretization using a simple function with known second derivative. The residual between the analytical solution and the derivative is computed for the following functions:

$$\boldsymbol{A} = \left\langle \begin{array}{l} -z \cdot y \cdot e^{-5(x^2+y^2+z^2)}\ \vec{i}; \\ -x \cdot z \cdot e^{-5(x^2+y^2+z^2)} \vec{j}; \\ -x \cdot y \cdot e^{-5(x^2+y^2+z^2)} \vec{k}; \end{array} \right\rangle \quad (11)$$

$$\varphi(x, y, z) = \left(tanh\,(x) + tanh\,(y) + tanh\,(z)\right) * \ e^{-10(x^2+y^2+z^2)} \quad (12)$$

Operators	Mesh size	Residual
Laplacian	2.50e-001	1.29e-001
	1.25e-001	9.24e-002
	6.25e-002	1.10e-002
	3.13e-002	8.05e-004
	1.56e-002	9.99e-004
Divergence	2.50e-001	1.11e-002
	1.25e-001	4.22e-002
	6.25e-002	1.18e-002
	3.13e-002	3.95e-003
	1.56e-002	9.88e-004
Curl	2.50e-001	2.70e-002
	1.25e-001	4.04e-002
	6.25e-002	7.85e-003
	3.13e-002	2.77e-003
	1.56e-002	7.31e-004
Gradient	1.25e-001	1.09e-001
	6.25e-002	3.66e-002
	3.13e-002	1.12e-002
	1.56e-002	7.89e-003

Table 1: Residual as a function of mesh size.

I second test, or forward operator test, looks at the accuracy of the forward operator. I used the same test as prescribed by Haber (1999):

$$(\,Conductivity\ model\,)\ \ \gamma = tanh\left(a\left(\varepsilon + \frac{1}{4}\right)\right) - tanh\left(a\left(\varepsilon - \frac{1}{4}\right)\right) + \frac{1}{100} \quad (13)$$

$$\mathbf{E} = \left\langle -\frac{z \cdot y \cdot e^{-5(x^2+y^2+z^2)}}{\gamma(x)}\vec{\imath}; \quad \frac{-x \cdot z \cdot e^{-5(x^2+y^2+z^2)}\vec{\jmath}}{\gamma(y)} \quad ; \quad \frac{x \cdot y \cdot e^{-5(x^2+y^2+z^2)}\vec{k}}{\gamma(z)} \right\rangle \quad (14)$$

Our goal is to compute a solution for A and phi using the forward operator described in (8). I can compute a fictitious source term from:

$$J = (i\,\omega\,\mu)^{-1}\,\nabla x \nabla x\, E \;-\; \sigma E \quad (15)$$

which is then applied on the RHS of equation (8).

However, I have a function for the field **E**, but not for its potentials. I therefore computed a pseudo-analytical solution for A and phi by solving the following system:

$$\begin{bmatrix} \mathbf{speye(nfaces)} & \mathbf{GRAD} \\ \mathbf{DIV} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \varphi \end{bmatrix} = \begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix} \; (16)$$

Table 2 shows the residuals for two mesh sizes. I had to restrain myself to two steps due to limited computational power. I unfortunately did not manage to recover the same numbers as Haber (2000). Even though the **E**-fields seem to converge nicely, the potentials do not improve in O(h^2). Even more suspicious is that the residual on DIV A is much smaller than in the paper, which makes me wonder if I applied the test properly. The second row of (16) assures that DIV A is zero, which drives the solution to a very small number. Without it, there is no constraint on what A can be, which would make the pseudo-analytical solution wrong. This is still a question that has to be answered.

n^3	**∂A**	**∂ϕ**	**∂|DIV A|**	$\lvert E_a - E_c \rvert$
8	2.53e-03	5.02e-04	2.78e-17	1.16e-03
16	1.59e-03	1.33e-04	5.55e-17	7.59e-04

Table 2: Residuals between analytical and computed fields.

Experimentation

Following these two tests, I thought it would be interesting to look at the behaviour of the fields as a function of frequency and conductivity. For all the following cases, the source is considered to be a planer wave field polarized in the x and y direction. Figure 1 shows the recovered primary field for a range of background half-space conductivity models. In the absence of lateral heterogeneity the electric field preserves its polarization and simply decays at depth. As expected, the rate of decay is increasing with conductivity and frequency.

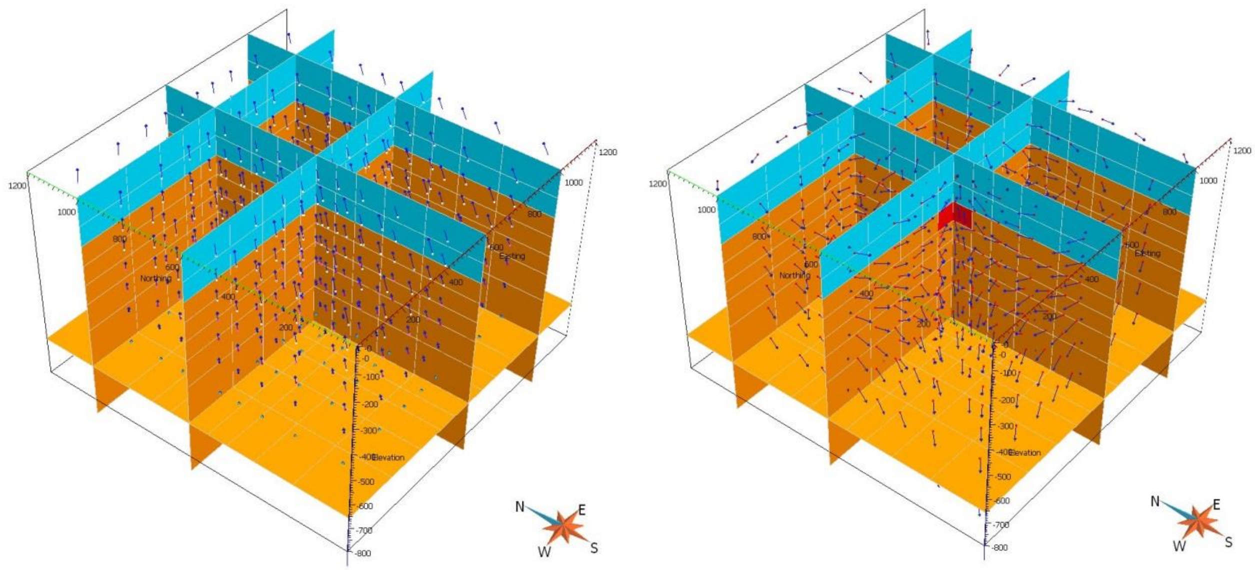
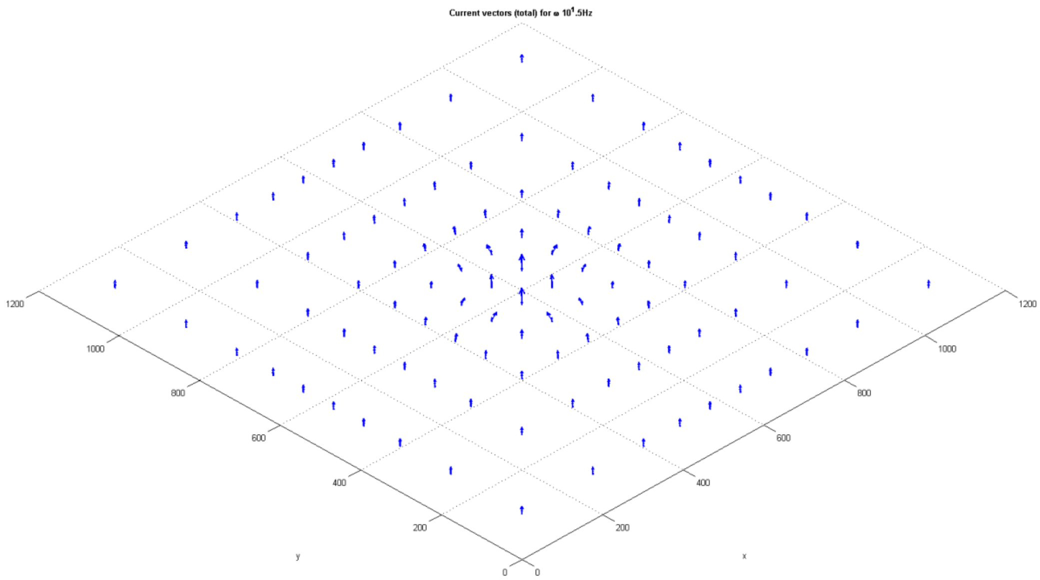


Figure 1: Solution for the primary and secondary E-field. In the absence of a 3D anomaly, the amplitude of the planar wave field decays at depth.

An even more interesting experiment looks at the behaviour of the field near a conductor. Figure 2 shows the current density around a conductive anomaly at two different frequencies. At low frequency, currents are channeled towards the extremities of the body, increasing locally the current density. This behaviour is analogous to the DC experiment. At higher frequency, the opposite can be observed - currents are forced around the conductor. This can be explained theoretically by the rapidly changing magnetic field, inducing eddy currents within the conductive body, which in turns oppose to the inducing fields.



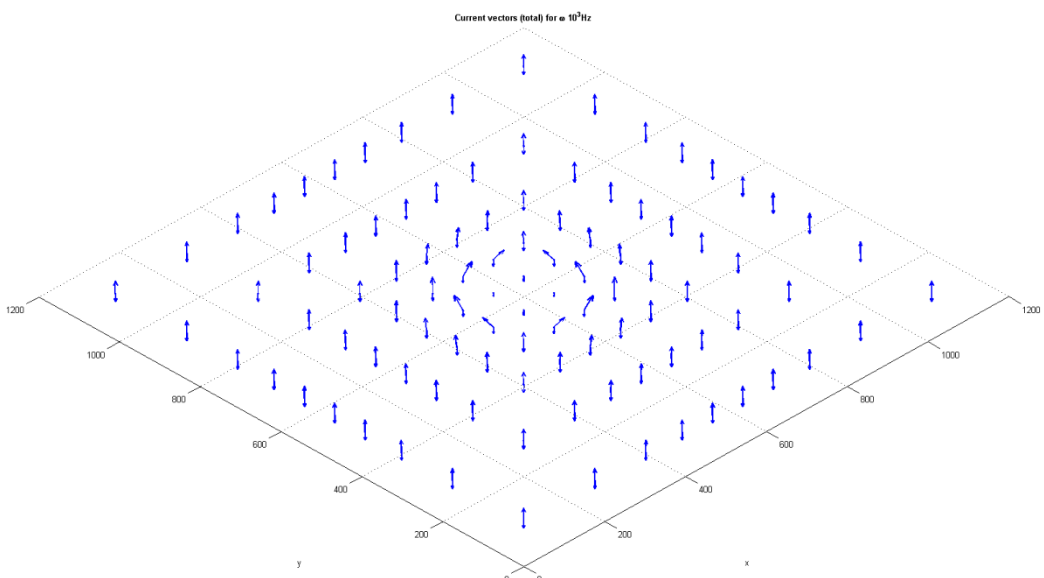


Figure 2: Solution for the current density (J) for two different frequencies. At low regime, the system behaves like the DC experiment. For high frequencies and good conductors, the back EMF caused by the changing magnetic field stops current flow.

Impedances

For the final step of the forward modeling I look into the computation of impedances. From the solution of equation (8) for to polarization I can solve equations (4 a-d). Our ultimate goal is to compare the solution of the forward with some field observations of impedance. This is usually expressed as:

$$\mathrm{Q}\ \mathrm{u} = \mathrm{d} \quad (17)$$

where **u** is a vector containing both A and phi, d are the four impedances values at every location and Q is a sparse operator linking the field to the impedance. The Q operator must do the following:

1. Compute the magnetic field (H) everywhere
2. Average the fields at the location of observations (cell center here for simplicity)
3. Select only the location of the observations
4. Compute the impedances
5. Repeat for two polarization

```
Qhx = SObsH * AVHx * CURL_f * SA / (1i*w*uo);
Qhy = SObsH * AVHy * CURL_f * SA / (1i*w*uo);

Qex = SObsEx * AVFvcz * AVC * SA;
Qey = SObsEy * AVFvcz * AVC * SA;

Zxx = @(ux, uy) ( (Qex*ux) .* (Qhy*uy) - (Qex*uy) .* (Qhx*ux) ) ./...
                ( (Qhy*uy) .* (Qhy*ux) - (Qhy*ux) .* (Qhx*uy) );

Zxy = @(ux, uy) ( (Qex*uy) .* (Qhx*ux) - (Qex*ux) .* (Qhx*uy) ) ./...
                ( (Qhy*uy) .* (Qhy*ux) - (Qhy*ux) .* (Qhx*uy) );

Zyx = @(ux, uy) ( (Qey*uy) .* (Qhy*ux) - (Qey*ux) .* (Qhy*uy) ) ./...
                ( (Qhx*uy) .* (Qhy*ux) - (Qhx*ux) .* (Qhy*uy) );


Zyy = @(ux, uy) ( (Qey*ux) .* (Qhx*uy) - (Qex*uy) .* (Qhx*ux) ) ./...
                ( (Qhx*uy) .* (Qhy*ux) - (Qhx*ux) .* (Qhy*uy) );
```

This is done using a series of in-line functions and sparse operators. The final result is a vector containing four impedance values per observation points.

I finally wanted to test this procedure. The easiest test was to compute impedance for a uniform half-space conductivity model. We know from theory that for a 1D model, the diagonal element of (3) should be null, and the off-diagonal elements should be equal. I can also compute an apparent conductivity using:

$$\boldsymbol{\sigma_{app}} = \frac{|\boldsymbol{Zxy}|^{-1}}{\boldsymbol{\omega\mu}} \qquad (\mathbf{18})$$

Table 3 displays the resulting impedances and apparent conductivity for a 1D model for σ = 1.

Z_{xx} & Z_{yy}	Z_{xy} & Z_{yx}	Computed apparent conductivity
1.0e-017 *	Impxy =	app_con =
(1,1) -0.5110 + 0.0864i	(1,1) 0.0073 + 0.0086i	0.9876
(2,1) -0.5188 + 0.1271i	(2,1) 0.0073 + 0.0086i	0.9876
(3,1) -0.4660 + 0.0702i	(3,1) 0.0073 + 0.0086i	0.9876
(4,1) -0.4016 + 0.0161i	(4,1) 0.0073 + 0.0086i	0.9876
(5,1) -0.3492 + 0.0005i	(5,1) 0.0073 + 0.0086i	0.9876
(6,1) -0.3161 + 0.0098i	(6,1) 0.0073 + 0.0086i	0.9876
(7,1) -0.3149 - 0.0201i	(7,1) 0.0073 + 0.0086i	0.9876
(8,1) -0.2729 - 0.0648i	(8,1) 0.0073 + 0.0086i	0.9876
(9,1) -0.2758 - 0.1779i	(9,1) 0.0073 + 0.0086i	0.9876
(10,1) -0.3228 - 0.2082i	(10,1) 0.0073 + 0.0086i	0.9876

Table 3: Impedances and apparent conductivity for a 1D model.

Conclusion

I manage to successfully code the forward operator and test its accuracy. I still have to understand the reason why I could not replicate Eldad's numbers for the forward operator test.

The next big step will be to code the inverse problem. I will have to accurately differentiate equations (4s), (8) and (17). Once I have an inverse problem, I would like to experiment using a compact support term in the objective function. Experimentation done on potential fields and the muon tomography problem showed great promise at recovering more compact anomalies

Special thanks to Ben Postlethwaite for his collaboration and to Nigel Phillips for letting me take time off work for this project.

References

Farquharson, C. Oldenburg, D. 2002. An algorithm for the three-dimensional inversion of magnetotelluric data. UBC-GIF.

Haber, E. Archer, U.M. Aruliah, D.A. Oldenburg, D. 1999. Fast simulation of 3D electromagnetic problems using potentials. Journal of Computational Physics **163**: 150-171

Chave, A.D. Jones, A.G. 2012. The magnetotelluric method: Theory and practice. Cambridge University Press

Annex A: Divergence Operator

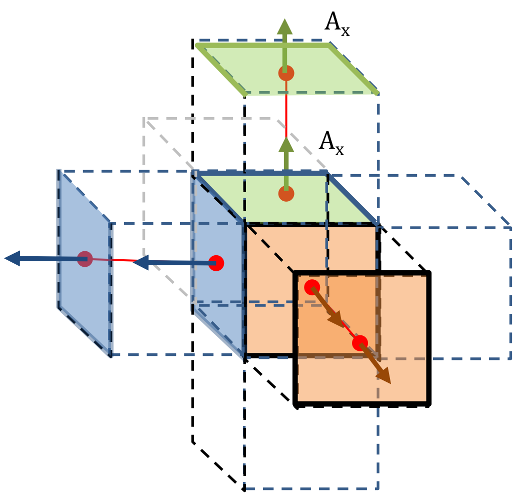
In theory…

$$\nabla \cdot \mathbf{A} = \frac{\partial A^x}{\partial x} + \frac{\partial A^y}{\partial y} + \frac{\partial A^z}{\partial z}$$

$$\frac{\partial A^x}{\partial x} = \frac{(A^x{}_{n+1} - A^x)}{h_n}$$

$$\frac{\partial A^y}{\partial y} = \frac{(A^y{}_{n+1} - A^y)}{h_n}$$

$$\frac{\partial A^z}{\partial z} = \cdots$$



In Matlab form…

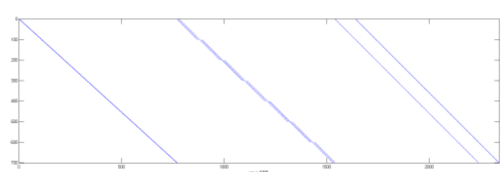
$$\frac{\partial}{\partial x} = \begin{bmatrix} 1/\sqrt{h_1} & & \cdots & 0 \\ -1/\sqrt{h_2} & 1/\sqrt{h_2} & \ddots & \vdots \\ 0 & & \cdots & -1/\sqrt{h_{end}} \end{bmatrix}$$

$$\frac{\partial A^x}{\partial x} = \mathrm{kron}\ (\ \mathrm{kron}\left(\mathrm{speye}(\mathrm{nz}),\ \mathrm{speye}(\mathrm{ny})\right), \frac{\partial}{\partial x})$$

Same idea for the 2 other partial derivatives yielding…

Divergence Operator

$$\mathbf{DIV} = [\mathbf{D_x} \quad \mathbf{D_y} \quad \mathbf{D_z}]$$



Boundary Conditions
(Primary Field)
x-y polarized

$$\frac{\partial A^x}{\partial x}\ and\ \frac{\partial A^y}{\partial x}\bigg|_{\partial\Omega} = 0$$

$$\frac{\partial A^x}{\partial y}\ and\ \frac{\partial A^y}{\partial y}\bigg|_{\partial\Omega} = 0 \qquad \Delta A^z|_{\partial\Omega} = 0$$

$$\frac{\partial A^x}{\partial z}\ and\ \frac{\partial A^y}{\partial z}\bigg|_{\partial\Omega} = g(x,y,z)$$

No boundary condition required since operates from face to center…

Annex B: Gradient Operator

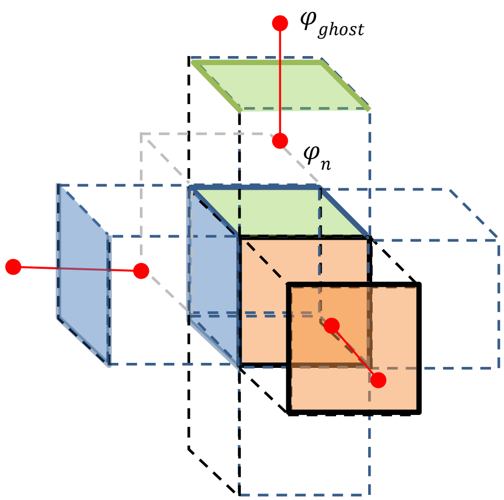
In theory…

$$\nabla\varphi = \left\langle \frac{\partial\varphi}{\partial x};\frac{\partial\varphi}{\partial y};\frac{\partial\varphi}{\partial z}\right\rangle$$

$$\frac{\partial\varphi}{\partial x} = \frac{(\varphi_{yzx-1} - \varphi_{zyx})}{h_{xmid}}$$

$$\frac{\partial\varphi}{\partial y} = \frac{(\varphi_{xzy-1} - \varphi_{xzy})}{h_{ymid}}$$

$$\frac{\partial\varphi}{\partial z} = \cdots$$



In Matlab form…

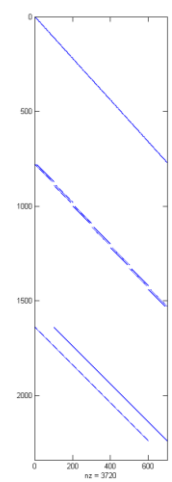
$$\frac{\partial}{\partial x} = \begin{bmatrix} 1/\sqrt{h_{mid\,x1}} & & \cdots & 0 \\ -1/\sqrt{h_{mid\,x2}} & 1/\sqrt{h_{mid\,x2}} & \ddots & \vdots \\ 0 & & \cdots & -1/\sqrt{h_{end}} \end{bmatrix}$$

$$\frac{\partial\varphi}{\partial x} = \mathrm{kron}\left(\mathrm{kron}\left(\mathrm{speye}(\mathrm{nz}),\ \mathrm{speye}(\mathrm{ny})\right), \frac{\partial}{\partial x}\right)$$

Same idea for the 2 other partial derivatives yielding…

Gradient Operator

$$\mathbf{DIV} = [\mathbf{D_x};\quad \mathbf{D_y};\quad \mathbf{D_z}]$$

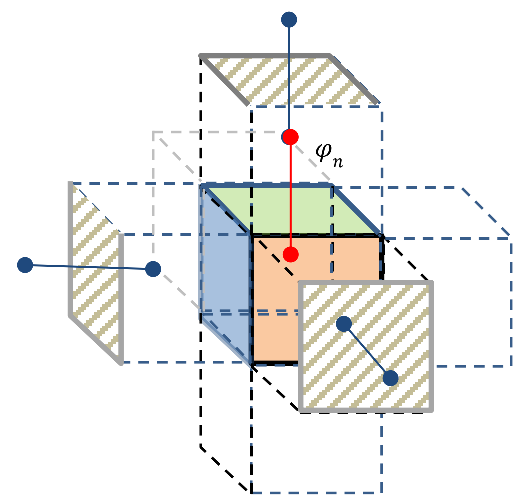


Boundary Conditions
(Primary Field)
x-y polarized

$$\left.\frac{\partial\varphi}{\partial n}\right|_{\partial\Omega} = 0$$

Boundary conditions requires that derivative operator is 0 on the edges…

$$\frac{\partial\varphi}{\partial n} = \mathrm{kron}\left(\mathrm{kron}\left((\mathrm{nz}),\ (\mathrm{ny})\right), dn([1{:}\,end] = 0)\right)$$

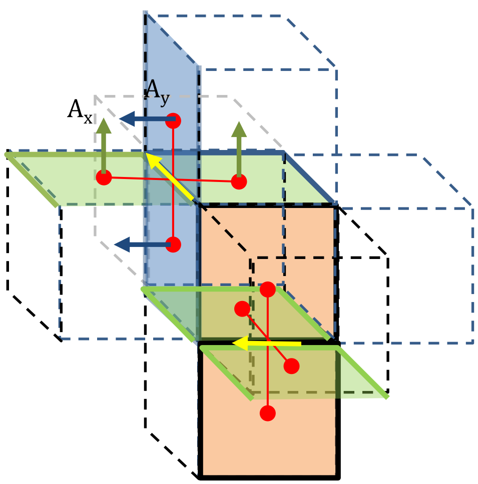


Annex C: Curl Operator

In theory…

$$\nabla \times A = \begin{bmatrix} 0 & \partial_z{}^{y} & -\partial_x{}^{z} \\ -\partial_z{}^{x} & 0 & \partial_x{}^{z} \\ \partial_y{}^{x} & -\partial_x{}^{y} & 0 \end{bmatrix}$$

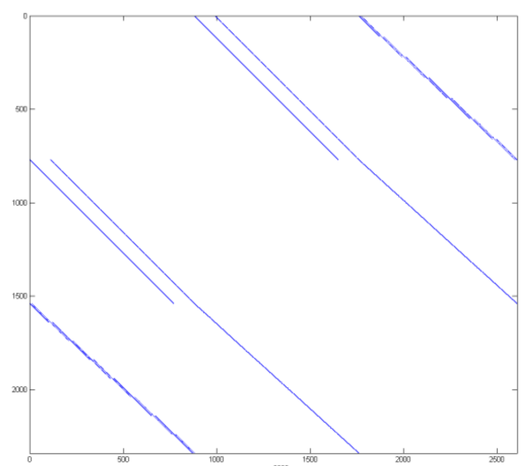
$$\nabla \times A^x = \frac{(A_{xyz-1}{}^{y} - A_{xyz}{}^{y})}{h_{zmid}} - \frac{(A_{xzy-1}{}^{z} - A_{xzy}{}^{z})}{h_{ymid}}$$



In Matlab form…

CURL Operator

$$\mathbf{CURL} = \begin{bmatrix} 0 & D_z{}^{y} & -D_x{}^{z} \\ -D_z{}^{x} & 0 & D_x{}^{z} \\ D_y{}^{x} & -D_x{}^{y} & 0 \end{bmatrix}$$



Boundary Conditions

(Primary Field)

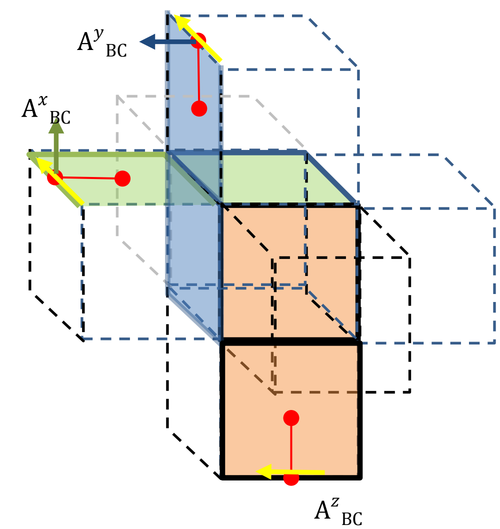
x-y polarized

$$\frac{\partial A^x}{\partial x} \; and \; \frac{\partial A^y}{\partial x}\bigg|_{\partial\Omega} = 0$$

$$\frac{\partial A^x}{\partial y} \; and \; \frac{\partial A^y}{\partial y}\bigg|_{\partial\Omega} = 0$$

$$\frac{\partial A^x}{\partial z} \; and \; \frac{\partial A^y}{\partial z}\bigg|_{\partial\Omega} = g(x, y, z)$$

Tricky to apply boundary matrix since we need to put value at the right location… Much simpler if the field vanishes at the boundary.



Annex D: Laplace Operator

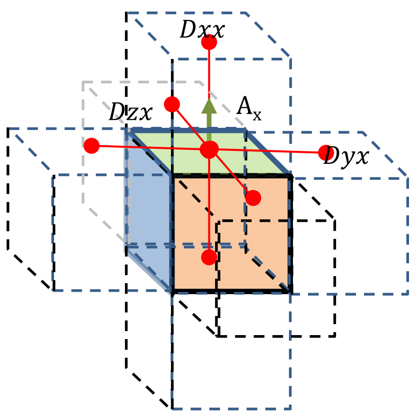
Vector Laplacian

$$\Delta A_x = \frac{\partial^2 A^x}{\partial x^2} + \frac{\partial^2 A^x}{\partial y^2} + \frac{\partial^2 A^x}{\partial z^2}$$

$$\frac{\partial^2 A_x}{\partial x^2} = \frac{\frac{(A^x{}_{n+1} - A^x)}{h_{n-1}} - \frac{(A^x - A^x{}_{n-1})}{h_n}}{h_{mid}}$$

$$\frac{\partial^2 A_x}{\partial y^2} = \frac{\frac{(A^x{}_{n+1} - A^x)}{h_{mid}} - \frac{(A^x - A^x{}_{n-1})}{h_{mid}}}{h}$$

$$\frac{\partial^2 A_x}{\partial z^2} = \ \dots$$



In Matlab form…

$$\frac{\partial}{\partial x} = \begin{bmatrix} 1/\sqrt{h_1} & & \cdots & 0 \\ -1/\sqrt{h_2} & 1/\sqrt{h_2} & \ddots & \vdots \\ 0 & & \cdots & -1/\sqrt{h_{end}} \end{bmatrix}$$

$$\frac{\partial^2 A^x}{\partial x^2} = \mathrm{kron}\ (\ \mathrm{kron}\left(\mathrm{speye}(\mathrm{nz}),\ \mathrm{speye}(\mathrm{ny})\right), \frac{\partial^2}{\partial x^2})$$

Same idea for the 8 other partial derivatives yielding…

Laplacian Operator

$$\mathbf{L} = \begin{matrix} \frac{\partial^2 Ax}{\partial x^2} + \frac{\partial^2 Ax}{\partial y^2} + \frac{\partial^2 Ax}{\partial z^2} & 0 & 0 \\ 0 & \frac{\partial^2 Ay}{\partial x^2} + \frac{\partial^2 Ay}{\partial y^2} + \frac{\partial^2 Ay}{\partial z^2} & 0 \\ 0 & 0 & \frac{\partial^2 Az}{\partial x^2} + \frac{\partial^2 Az}{\partial y^2} + \frac{\partial^2 Az}{\partial z^2} \end{matrix}$$

Boundary Conditions
(Primary Field)
x-y polarized

$$\frac{\partial A^x}{\partial x}\ and\ \frac{\partial A^y}{\partial x}\bigg|_{\partial\Omega} = 0$$

$$\frac{\partial A^x}{\partial y}\ and\ \frac{\partial A^y}{\partial y}\bigg|_{\partial\Omega} = 0 \qquad \Delta A^z|_{\partial\Omega} = 0$$

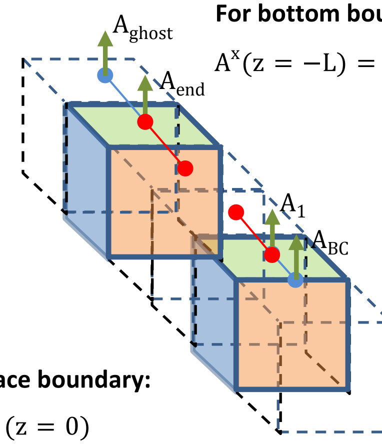
$$\frac{\partial A^x}{\partial z}\ and\ \frac{\partial A^y}{\partial z}\bigg|_{\partial\Omega} = g(x,y,z)$$

Defining g(x,y,z) as the 1D problem for the x and y components:

$$A^x(z) = e^{i\sqrt{i\,\omega\mu\sigma}}$$

For bottom boundary:

$$A^x(z = -L) = i\sqrt{i\,\omega\mu\sigma}A^x - A^x$$



For surface boundary:

$$A^x{}_{BC}\ (z = 0)$$