Practical aspects of running EH3D

E. Haber*

June 4, 2001

Abstract

This document discuss of practical aspects of running the program ***EH***3D . The goal of this document is to help practitioners determine and tune parameters in the program.

1 Introduction

The program ***EH***3D is an implementation of our work [3, 4] in a Fortran90 code with a Graphical User Interface.

The program is flexible and allows the user to solve a variety of EM geophysical problems involving the following EM source types:

- Grounded sources
- Loop sources
- Plane waves
- Any secondary field (given the primary)

The program allows arbitrary conductivity, permeability and susceptibility structures as well as arbitrary source frequencies. However, this flexibility comes with a cost because for some choices of parameters care must be taken in order to be able to solve the systems which are generated and to insure physically reasonable solutions.

*Departments of Computer Science and of Earth and Oceanography Sciences, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada. (haber@cs.ubc.ca).

In this document we try to give a basic understanding of the issues involved in the solution of the wide range of problems that can be encountered in the process of solving a complex 3D electromagnetic problem.

The document is divided as follows. In Section 2 we review the equations and the discretization of the system. In Section 3 we discuss gridding issues. In Section 4 we discuss the sources and their discretization. In Section 6 we discuss issues which relate to the solution of the linear system. We summarize this document by giving some illustrating examples.

2 The fundamental equations and discretization

In this section we briefly review some of the important results of our work in [3, 4]. This is by no means a full theoretical discussion and we refer the interested reader to the papers.

Maxwell's equations in the frequency domain are

$$\boldsymbol{\nabla} \times \boldsymbol{E} - \imath\omega\mu\boldsymbol{H} = \boldsymbol{0}, \tag{1a}$$

$$\boldsymbol{\nabla} \times \boldsymbol{H} - \widehat{\sigma}\boldsymbol{E} = \boldsymbol{J}^s, \tag{1b}$$

$$\boldsymbol{n} \times \boldsymbol{H} = 0, \qquad \partial\Omega \tag{1c}$$

where μ is the magnetic permeability, σ is the conductivity, ϵ is the electrical permittivity,

$$\widehat{\sigma} = \sigma - \imath\omega\epsilon, \tag{2}$$

and $\boldsymbol{J}^s$ is a known source current density. In our work we assume that the physical properties $\mu > 0$, $\epsilon > 0$ and $\sigma > 0$ can vary with position. We also assume the **quasi-static** assumption namely that

$$\mu\epsilon\omega^2 L^2 \ll 1 \tag{3}$$

where L is a typical length scale.

To over-come numerical difficulties which arise in the quasi-static range of frequencies we have reformulated the problem in terms of the Helmholtz decomposition of the electric field

$$\boldsymbol{E} = \boldsymbol{A} + \boldsymbol{\nabla}\phi, \tag{4a}$$

$$\nabla \cdot \boldsymbol{A} = 0. \tag{4b}$$

This gives a system of equations for the vector $\boldsymbol{A}$ and the scalar ϕ

$$\begin{aligned} \boldsymbol{\nabla} \times (\mu^{-1} \boldsymbol{\nabla} \times \boldsymbol{A}) - \boldsymbol{\nabla}(\mu^{-1} \nabla \cdot \boldsymbol{A}) - \imath\omega\widehat{\sigma}(\boldsymbol{A} + \boldsymbol{\nabla}\phi) &= \imath\omega \boldsymbol{J^s}, \qquad (5a) \\ \nabla \cdot \sigma \boldsymbol{A} + \nabla \cdot \sigma \boldsymbol{\nabla}\phi &= -\nabla \cdot \boldsymbol{J^s}. \quad (5b) \end{aligned}$$

The system (5) is discretized using a finite volume method on a staggered grid. The staggered grid ensures **second order accuracy** of $\boldsymbol{E}$ and $\boldsymbol{H}$. The variables $\boldsymbol{A}$ are located at the cell's faces and ϕ is located at the cell's center. This implies that $\boldsymbol{H}$ is at the cell's edges, and $\widehat{\boldsymbol{J}}$ (and therefore $\boldsymbol{E}$) at cell's faces. The distribution of variables over the grid cell is summarized in Table 1 and plotted in Figure 1

$$\begin{array}{cc} A^x{}_{i+\frac{1}{2},j,k} \approx A^x(x_{i+\frac{1}{2}}, y_j, z_k) & \hat{J}^x_{i+\frac{1}{2},j,k} \approx \hat{J}^x(x_{i+\frac{1}{2}}, y_j, z_k) \\ A^y{}_{i,j+\frac{1}{2},k} \approx A^y(x_i, y_{j+\frac{1}{2}}, z_k) & \hat{J}^y_{i,j+\frac{1}{2},k} \approx \hat{J}^y(x_i, y_{j+\frac{1}{2}}, z_k) \\ A^z{}_{i,j,k+\frac{1}{2}} \approx A^z(x_i, y_j, z_{k+\frac{1}{2}}) & \hat{J}^z_{i,j,k+\frac{1}{2}} \approx \hat{J}^z(x_i, y_j, z_{k+\frac{1}{2}}) \\ H^x_{i,j+\frac{1}{2},k+\frac{1}{2}} \approx H^x(x_i, y_{j+\frac{1}{2}}, z_{k+\frac{1}{2}}) & \\ H^y_{i+\frac{1}{2},j,k+\frac{1}{2}} \approx H^y(x_{i+\frac{1}{2}}, y_j, z_{k+\frac{1}{2}}) & \\ H^z_{i+\frac{1}{2},j+\frac{1}{2},k} \approx H^z(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, z_k) & \\ \phi_{i,j,k} \approx \phi(x_i, y_j, z_k) & \end{array}$$

Table 1: Summary of the discrete grid functions. Each scalar field is approximated by the grid functions at points slightly staggered in each cell $e_{i,j,k}$ of the grid.

It is important to understand where on the grid the variables are placed, especially when one is looking at these values rather than using alternative outputs, such as those obtained from output conversion utilities such as EHPOINTS.EXE, and others.

Using this discretization and utilizing a standard finite volume approach we obtain a large, sparse system for the grid functions corresponding to $\boldsymbol{A}$ and ϕ

$$K\ u\ = \begin{pmatrix} C^T M C + D^T M_c D - \imath\omega S & \imath\omega S D^T \\ -DS & DSD^T \end{pmatrix} \begin{pmatrix} \boldsymbol{A} \\ \phi \end{pmatrix} = \begin{pmatrix} b_A \\ b_\phi \end{pmatrix} = b. \qquad (6)$$

Here C corresponds to the discretization of the operator $\boldsymbol{\nabla}\times$; D likewise corresponds to the discretization of the operator $\nabla\cdot$; the diagonal matrix S

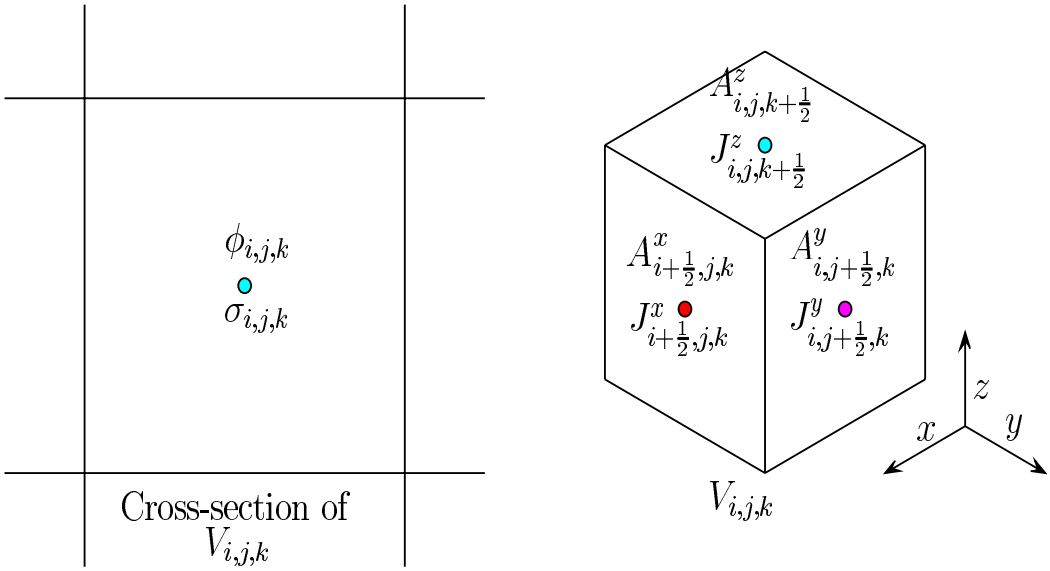

Figure 1: Staggered grid for the discretization.

results from the discretization of the operator $\widehat{\sigma}(\cdot)$; M and M_c similarly arise from the operator $\mu^{-1}(\cdot)$ at cell edges and at cell centers respectively.

Note that as long as the frequency is low enough that the diffusion number satisfies

$$d = \omega\mu\widehat{\sigma}h \ll 1$$

(where h is the grid spacing) the matrix is dominated by the diagonal blocks, that is, we have a DC type experiment with a DC type physics. This is a very important observation as it helps in the design of the grid and in the selction of solvers.

3 Mesh related issues

The program ***EH***3D uses the mesh as one of its input parameters. This means that the user is responsible for generating the mesh for the problem, which rise two fundamental questions

- What is the mesh spacing that should be used?
- Where should the mesh be terminated?

Although one may want to use a very fine grid to resolve small features in the field this may take more time and memory. Our experience so far has been that a machine of one G-Byte can run grids up to 64^3 cells.

This section does not answer these questions exactly as this is an active research area. However, we do try to give some basic concepts which can help the practitioner to design the mesh.

3.1 Mesh spacing

We start with the question of mesh spacing. In principle, as cell size gets smaller, the results (computed fields) are more accurate. However, there is a tradeoff because for a finer mesh the number of unknowns is large and the time to solve the discrete system of equation is larger. Therefore, although we would like to use a fine mesh, we might want to avoid too fine a mesh because the resulting computational time and the memory requirement may be excessively large.

As we discussed above, ***EH***3D assumes that we are working in the quasi-static region. This implies that the experiment has similar physical features to the DC experiment namely

- The fields decay toward infinity
- The fields are generally smooth on the grid and vary slowly

Thus if one has some intuition about the DC case, then similar methods can be applied to the quasi-static case. Care must be taken such that the cells are smaller than the skin-depth.

The above can be summarized by a few simple set of guidelines

1. *Mesh finely where the "action" in the solution is.*
   This simple guideline implies that in order to resolve the variations in the solution, one needs to mesh finely in regions where the solution changes the most (such as in the skin area). In many problems it is hard to know these regions *a-priori* and therefore a technique of *gradual refinement* is used. This means that the user meshes the problem on a uniform coarse grid and uses the solution on this grid to find the active regions. The user then re-meshes the problem by putting more cells in the active regions.

2. *Keep a reasonable aspect ratio.*
   Although one may want to keep the discretization fine where the action is, one needs to pay attention not to generate cells that have a large aspect ratio, (greater than 10 or so). This is especially true for the regions where the solution changes. The reason is that such large aspect ratios generate an unscaled system and as a result, the problem may be ill-conditioned and harder to solve.

3.2 Mesh termination

Geophysical problems consider the whole space as the domain but in practice the grid must be terminated at a finite length. The question of mesh termination is somewhat simpler than of grid spacing. Again we will see that the DC setting can give most of the intuition needed.

We know that the electric field decays as

$$\boldsymbol{E} \approx \frac{\exp((i\omega\widehat{\sigma})^{\frac{1}{2}}R)}{R^3}$$

We see that there are two main factors that determine the decay. First, there is the geometrical spreading term R^{-3}. This term implies that even for a very low frequency the field decays at a rate which is similar to the DC rate. As the frequency gets higher, the field is further damped by the skin depth effect.

Thus the padding and the termination of the grid should be smaller or equal to the DC case. We summarize this by the following guidelines

1. *Do not terminate the grid where there is "action".*
   This is similar to the above guideline and it implies that the boundary area needs to be an area where things do not change in any significant way.

2. *Pad using reasonable cell increments.*
   If cells that are very coarse are put close to very fine cells, the stability of the system can be compromized and convergence may be slow. Therefore, the padding should be done gradually in cells that do not have an expansion rate of more than 2.

In cases of uncertainty, one could determine the size of the domain using a similar approach to the gradual refinement we discussed in the last subsection. This is done by solving a sequence of problems where each problem is

solved on a larger domain. The solution is regarded as satisfactory when the fields do not change as the domain becomes larger.

4 Wire sources and the mesh

Sources in ***EH***3D are either wires carrying a current or elese, a primary $\boldsymbol{J}$ or $\boldsymbol{H}$ fields due to a prescribed conductivity. In this section we shortly discuss the placement of wires on the grid.

We start by a mathematical description of a unit current which flows in a wire that stretches from $[-L/2, y_0, z_0]$ to $[L/2, y_0, z_0]$ as in Figure 2

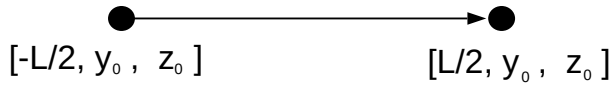

Figure 2: Wire in the x direction.

$$\boldsymbol{J^s} = \begin{pmatrix} (H(x - L/2) - H(x + x/2))\ \delta(y - y_0)\ \delta(z - z_0) \\ 0 \\ 0 \end{pmatrix} \tag{7}$$

where H the the Heaviside function and δ is the delta function. This is an exact mathematical description of the current that needed to be integrated into ***EH***3D .

As discussed in 2 ***EH***3D is a finite volume code which implies that this right hand side is integrated on a finite volume. The right hand side has dimensions of current which implies it has to be evaluated in volumes where the currents reside, that is, the volumes that include the cell's faces. Consider a cross section of the volumes where the wire lies as plotted in Figure 3. In this example, the wire crosses three cells and should be integrated through them. The integration volume between the firs two cells is shaded in the picture.

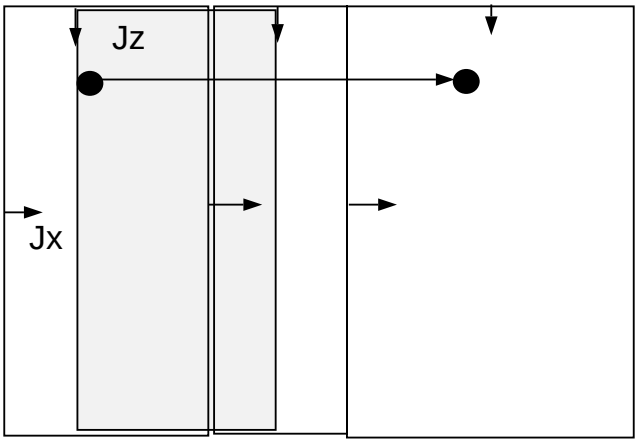

Figure 3: Integration over a wire in the x direction.

Integrating the wire in this volume gives

$$\iiint_{v_i} \boldsymbol{J^s}\, dV = \iiint_{v_i} \begin{pmatrix} (H(x-L/2) - H(x+L/2))\ \delta(y-y_0)\ \delta(z-z_0) \\ 0 \\ 0 \end{pmatrix} dx\, dy \quad (8)$$

$$= \int_{x_{left}}^{x_{right}} \begin{pmatrix} (H(x-L/2) - H(x+L/2)) \\ 0 \\ 0 \end{pmatrix} dx = \begin{pmatrix} h_x \\ 0 \\ 0 \end{pmatrix}$$

where h_x is the grid spacing. This result is then used in the code for the right hand side.

The source in the picture does not have to be aligned with the $\widehat{\boldsymbol{J}}_x$ grid. This may result in some conceptual difficulties that we now describe. Assume for a moment a different discretization plotted in Figure 4. Here the wire is at the bottom of the grid rather than near the top of it as in Figure 3. Repeating the integration we see that we obtain the *same* result as the scenario we had before, where the wire was in the top part of the cell. In fact, it is easy to see that the result remains constant for **every** wire which cross the cell perpendicularly regardless to its position *within* the cell. This refered to as the loss of resolution on a grid level since it is clear that one cannot expect to obtain resolution smaller than the grid size. If we want to better resolve the wire, we need a finer discretization for it.

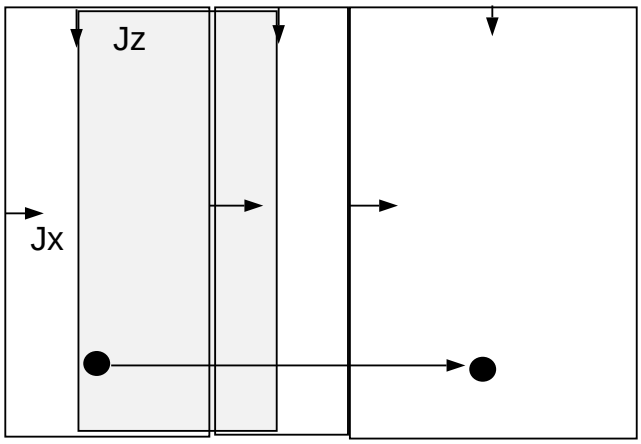

Figure 4: Integration over a wire in the x direction.

One way of thinking about this issue is in terms of an equivalent current density distributed over the whole cross section of the cell. Regardless of where the *actual* current filament is (vertically within the cell), the program produces results that are caused by a *distributed* current density that is equivalent to the individual current filament.

In order to make the user aware of this fact, ***EH***3D prints the cell center that the wire is going through. In this way, the users should be aware about the cell resolution they obtain.

One conclusion is that, as in other similar codes, results in cells that are within one or two cells of the source location should be treated as more inaccurate than results in cells that are more distant from the source.

5 Using primary fields as input rather than specific sources - Secondary fields computation

In some cases, the sources are very far from the region of interest. In these cases, one may need to discretize a huge area with relatively small cells, which may impose computational difficulties. Furthermore, the dynamic range of the field is very large. Near the source they may be of $\mathcal{O}(1)$ but they decay

by a few orders of magnitude in the region of interest. In cases of such a large dynamical range, the results may not be very accurate.

One way to overcome this problem is by transforming the problem to calculate secondary fields [5]. This requires the calculation of a *primary field* assosiated with the *background conductivity*. ***EH***3D allow the user to define the primary fields as the values of the current $\boldsymbol{J}$ over the grid (note again the staggered grid). The code requires both the primary fields and the background conductivity for the calculation of the secondary fields. Care must be taken such that the input primary fields uses the same conventions as ***EH***3D .

6 The linear solver

After the mesh has been specified and the right hand side calculated, the matrix system (6) is generated and the fields are solved using an iterative method.

In general we write the system as

$$Ku = b$$

where K is the system b is the right hand side and u contains the fields. The matrix K is sparse, that is, although it is very large (a few millions for a 64^3 grid), it has only a very small number of non-zero elements. Such matrix systems can be solved using iterative methods and our implementation uses BiCGstab [1] as the iterative solver. However, in most applications, many iterations are needed unless the matrix is preconditioned. The term preconditioning means that the original system is changed to

$$M^{-1}Ku = M^{-1}b$$

where M is a preconditioning matrix.

Obviously, if $M = K$ then the problem is solved in one iteration however, calculating the inverse of K is a computational challenge that requires huge amount of computational time. Therefore, one tries to obtain a good and sparse approximation to the inverse of K. This is an ongoing research topic which arise in most computational science.

In our implementation we have used two types preconditioners. First, as mentioned before, the system is DC like for low frequencies. We therefore use

an approximation to the DC case as a preconditioner. The approximation is based on the LU decomposition of the diagonal blocks of K. If we set the frequency to 0 and the tolerance for the LU to 0, the problem will be solved in 1 iteration. However, this requires a very large size of memory and therefore we use an Incomplete LU (ILU) factorization which does not give as good of a preconditioner but can still converge in a small number of steps.

The ILU preconditioner is based on the idea that the problem is DC like. However, as the frequency becomes higher, the problem does not behave as a DC type problem. In this case the ILU is not a good preconditioner and we use the SSOR [2] as a preconditioner.

To finalize we give the following set of guidelines

- *For low frequency use the ILU preconditioner*
- *For higher frequencies (where the ILU is not very effective) use SSOR*
- *If there is no memory limitation use ILU with low tolerance (lower than* 10^{-2}*)*
- *If you are limited by memory use either ILU with a high tolerance or SSOR*
- *As a rule of thumb, failure is possible when any cell has an aspect ratio* > 10*, although padding cells should be less of a problem because fields in padding cells should be small.*

Another important issue for the solver is the convergence. The following question has no simple answer: *"How do we know that we have converged and how to set the tolerance of the solver?"*. To illustrate we consider an example.

$$\begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix}$$

Of course we need not use iterative methods for this system and the solution is

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

However, for this example, consider we use an iterative solver and assume that at iteration k we have the approximate solution

$$\begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \end{pmatrix} = \begin{pmatrix} 0.9999 \\ 0.1 \end{pmatrix}$$

Simple calculation shows that

$$r = \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \end{pmatrix} - \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} = - \begin{pmatrix} 0.0001 \\ 0.9\,\epsilon \end{pmatrix}$$

and the square norm of the residual is

$$\|r\|^2 = 0.0001^2 + (0.9\,\epsilon)^2 = 10^{-8} + 0.81\epsilon^2$$

For ϵ small (say $1e-3$) the norm of the residual looks small and we may think we have converged. However, this is obviously not true for the second component as it is very far from the actual solution.

For this reason, the choice of norms which represents the accuracy of the solution is not an easy task. The norm used in ***EH***3D is the two norm described above which works best if the matrix is scaled. This assumption is not correct for very large aspect ratios and therefore results may be biased. This is one of the reasons that care should be taken in the discretization to select grids with good aspect ratios.

References

[1] R. Barrett, M. Berry, T. Chan, J. Demmeland, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1994.

[2] G. Golub and C. V. Loan, *Matrix Computations*, Johns Hopkins University Press, 1983.

[3] E. Haber and U. Ascher, *Fast finite volume modeling of 3d electromagnetic problems with highly discontinuous coefficients*, to appear in SISC, (1999).

[4] E. Haber, U. Ascher, D. Aruliah, and D. Oldenburg, *Fast simulation of 3D electromagnetic using potentials*, J. Comput. Phys., (2000). To appear.

[5] R. Wehage and E. Haug, *Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems*, J. of Mechanical Design, 104 (1982), pp. 247–255.