BigHeadFish Solver: Physics and Mathematical Formulation

Project Documentation

January 30, 2026

1 Introduction

The **BigHeadFish Solver** is a 3D Finite-Difference Frequency-Domain (FDFD) electromagnetic simulation engine designed for Logging-While-Drilling (LWD) applications. It solves the vector wave equation in anisotropic media using a staggered grid (Yee grid) approach with non-uniform discretization and Perfectly Matched Layers (PML) absorbing boundary conditions.

2 Governing Equations

The solver is based on the frequency-domain Maxwell's equations assuming a time-harmonic dependence of $e^{-i\omega t}$. In conductive media typical of geophysical formations, displacement currents (ϵ) are negligible compared to conduction currents (σ). The system is governed by the vector wave equation for the electric field $\mathbf{E}$:

$$\nabla \times \left(\mu^{-1} \nabla \times \mathbf{E}\right) - i\omega\hat{\sigma}\mathbf{E} = i\omega\mathbf{J}_s \tag{1}$$

Where:

- $\mathbf{E}$ is the electric field vector [V/m].
- μ is the magnetic permeability (assumed μ_0).
- $\omega = 2\pi f$ is the angular frequency.
- $\hat{\sigma}$ is the anisotropic electrical conductivity tensor [S/m].
- $\mathbf{J}_s$ is the electric source current density.

3 Material Properties: Tilted Transverse Isotropy (TTI)

The solver models geological formations as anisotropic media. The conductivity tensor $\hat{\sigma}$ in the principal axes (x', y', z') is defined as:

$$\hat{\sigma}_{principal} = \begin{pmatrix} \sigma_h & 0 & 0 \\ 0 & \sigma_h & 0 \\ 0 & 0 & \sigma_v \end{pmatrix} \tag{2}$$

where $\sigma_h = 1/\rho_h$ is the horizontal conductivity and $\sigma_v = 1/\rho_v$ is the vertical conductivity.

To transform this to the Cartesian coordinate system (x, y, z), we apply rotation matrices based on the dip angle (θ) and azimuth angle (ϕ):

$$\hat{\sigma} = R^T \hat{\sigma}_{principal} R \tag{3}$$

where R is the Euler rotation matrix.

4 Domain Truncation: Perfectly Matched Layers (PML)

To simulate an infinite domain within a finite computational box, we utilize the Complex Coordinate Stretching technique (PML). This transforms the spatial derivatives ∂_ξ into $\frac{1}{s_\xi}\partial_\xi$, where $\xi \in \{x, y, z\}$. The stretching factor s_ξ is defined as:

$$s_\xi(\xi) = 1 + i\frac{\sigma_{PML}(\xi)}{\omega\epsilon_0} \tag{4}$$

In the code, the PML profile follows a cubic ramp polynomial:

$$\sigma_{PML}(d) = \sigma_{max}\left(\frac{d}{L_{pml}}\right)^3 \tag{5}$$

where d is the distance from the inner PML interface and L_{pml} is the thickness of the PML region.

5 Numerical Discretization (FDFD)

5.1 Non-Uniform Staggered Grid

The domain is discretized using a non-uniform Yee grid. The field components are staggered:

- E_x is located at $(i + 1/2, j, k)$
- E_y is located at $(i, j + 1/2, k)$
- E_z is located at $(i, j, k + 1/2)$

This arrangement naturally satisfies the divergence conditions and simplifies the implementation of the curl operators.

5.2 Finite Difference Operators

We define two types of edge lengths for the non-uniform grid:

- **Primary Edge Length (Δl):** The distance between integer nodes (used for $\nabla \times \mathbf{E}$).
- **Dual Edge Length (Δl^*):** The distance between half-integer centers (used for $\nabla \times \mathbf{H}$).

The discretized curl-curl operator $\nabla \times \mu^{-1}\nabla \times \mathbf{E}$ for the E_x component at index (i, j, k) is approximated as:

$$\begin{aligned}[\nabla \times \nabla \times \mathbf{E}]_x \approx \quad & \frac{1}{\Delta y_j^*}\left(\frac{E_{x,j} - E_{x,j-1}}{\Delta y_{j-1}} - \frac{E_{x,j+1} - E_{x,j}}{\Delta y_j}\right) \\ & + \frac{1}{\Delta z_k^*}\left(\frac{E_{x,k} - E_{x,k-1}}{\Delta z_{k-1}} - \frac{E_{x,k+1} - E_{x,k}}{\Delta z_k}\right)\end{aligned} \tag{6}$$

(Note: Cross-terms are omitted for brevity but are handled in the general anisotropic assembler.)

5.3 Linear System Assembly

The discretization results in a sparse linear system of the form:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{7}$$

where:

- $\mathbf{A} = \mathbf{K} - i\omega\mathbf{M}$ is the system matrix.
- $\mathbf{K}$ represents the stiffness matrix (Curl-Curl operator).
- $\mathbf{M}$ represents the mass matrix (Conductivity/PML terms).
- $\mathbf{x}$ is the vector of unknown electric field components (E_x, E_y, E_z).
- $\mathbf{b}$ is the source vector $(i\omega\mathbf{J}_s)$.

6 Source Modeling

The solver supports two types of sources:

1. **Electric Dipole (Mini-Bait)**: Modeled as a current density $\mathbf{J}$ applied directly to the edge corresponding to the dipole direction.
2. **Magnetic Dipole (Gem-Bait)**: Modeled as a small current loop. A magnetic dipole $\mathbf{M}$ along the z-axis is implemented as four electric currents forming a loop in the xy-plane surrounding the source node.

7 Grid Stretching Strategy

To handle low frequencies (e.g., 500 Hz) where wavelengths are large, while maintaining high resolution near the source, we use geometric grid stretching:

$$\Delta x_{i+1} = \Delta x_i \times r \tag{8}$$

where r is the stretch ratio (e.g., 1.15). This allows the simulation domain to extend to hundreds of meters without an excessive number of grid points.