Numerical Simulation Methods in Geophysics: Report II

1. MGPY+MGIN

thomas.guenther@geophysik.tu-freiberg.de

TU BERGAKADEMIE FREIBERG
TUBAF
Die Ressourcenuniversität.
Seit 1765.

Helmholtz equations for H

The magnetic field in the frequency domain is governed by the Helmholtz equations (no displacement currents, no sources)

$$\boldsymbol{\nabla} \times \sigma^{-1} \boldsymbol{\nabla} \times \mathbf{H} + \imath\omega\mu\mathbf{H} = 0 \tag{1}$$

2D E/H polarization: $-\boldsymbol{\nabla} \cdot \boldsymbol{\nabla} H_x + \imath\omega\sigma\mu H_x = 0$

discretized by FE with $\mathbf{A} + \imath\omega\mathbf{M}_\sigma = \mathbf{b}$ ($\mathbf{A}$-stiffness, $\mathbf{M}$-mass)

$\mu = \mu_0$, upper boundary ($\mathbf{b}$): $H_x(z = 0) = 1$ (Dirichlet BC)

Plane wave problem and solution

Task: simulate magnetic field in the Earth for a plane wave excitation

$$\mathbf{H}(z,\omega) = \begin{bmatrix} H_x^0 \\ 0 \\ 0 \end{bmatrix} e^{\imath\omega t}, \quad z < 0$$

For homogeneous $\sigma = \sigma_0$, the solution is (see Theory EM)

$$H_x(z) = H_x^0 e^{-\imath k z} = H_x^0 e^{-\sqrt{2\omega\mu\sigma_0}(1+\imath)z} \tag{2}$$

Secondary field for EM

$$-\boldsymbol{\nabla}^2(\mathbf{H}_0 + \mathbf{H}_a) + \imath\omega\mu(\sigma_0 + \delta\sigma)(\mathbf{H}_0 + \mathbf{H}_a) = 0$$

$$-\boldsymbol{\nabla}^2\mathbf{H}_0 + \imath\omega\mu\sigma_0\mathbf{H}_0 = 0$$

$$-\boldsymbol{\nabla}^2\mathbf{H}_a + \imath\omega\mu\sigma\mathbf{H}_a = -\imath\omega\mu\delta\sigma\mathbf{H}_0$$

discretized by

$$\mathbf{A}\mathbf{H}_a + \imath\omega\mathbf{M}_\sigma\mathbf{H}_a = -\imath\omega\mathbf{M}_{\delta\sigma}\mathbf{H}_0$$

Implementation with pyGIMLi solver module

```
import pygimli.solver as ps

A = ps.createStiffnessMatrix(mesh)
M = ps.createMassMatrix(mesh, sigma*mu)
b = np.zeros(mesh.nodeCount())
ps.assembleDirichletBC(A, [[grid.boundary(1), 1]], b)
M.cleanRow(0)  # first row
# or (otherwise homogeneous Neumann BC)
bc = dict(Dirichlet={-1: 1.0, -2: 0.0})  # general
ps.assembleBC(bc, mesh, A,  b)
```

Solve complex systems with pyGIMLi

directly using complex matrix

```
C = pg.core.CSparseMatrix(A.vecColPtr(), A.vecRowIdx(),
                          pg.core.toComplex(A.vecVals(), M.vecVals()*w))
c = pg.core.toComplex(b, b*0)
u = ps.linSolve(C, c)
```

or transform into real system

$$\begin{pmatrix} A & -\omega M \\ \omega M & A \end{pmatrix} \begin{pmatrix} u_r \\ u_i \end{pmatrix} = \begin{pmatrix} b_r \\ b_i \end{pmatrix}$$

```
ndof = mesh.nodeCount()
B = pg.BlockMatrix()
B.Aid = B.addMatrix(A)
B.Mid = B.addMatrix(M)
B.addMatrixEntry(B.Aid, 0, 0)
B.addMatrixEntry(B.Aid, ndof, ndof)
B.addMatrixEntry(B.Mid, 0, ndof, scale=-w)
B.addMatrixEntry(B.Mid, ndof, 0, scale=w)
```

Report II - 2D plane-wave EM

1. Start with a period of $T = 0.1$s and compute skin depth
2. Create a subsurface 2D model with some bodies
3. Compute the primary magnetic field analytically and numerically for a homogeneous conductivity
4. Visualize both and compare numerical & analytical solutions
5. Investigate the influence of model discretization
6. Change the period and rerun, compare
7. Change the conductivity and rerun, compare

Report II - secondary field

1. Compute the field with a conductive anomaly
2. Compute the secondary field
3. Visualize the secondary and total fields
4. Replace the good conductor with a bad conductor and rerun
5. Discuss the boundary conditions for total and secondary fields