Adaptive mesh refinement for parameter identification and application to electromagnetic inverse problems

Eldad Haber
contributors: Stefan Heldmann, Jan Modersitzki, Uri Ascher, Doug Oldenburg
Motivation - Geophysics

Rocks have different electrical properties

<table>
<thead>
<tr>
<th>Rock</th>
<th>Resistivity ($\Omega/m$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clay</td>
<td>1-20</td>
</tr>
<tr>
<td>Sand wet to moist</td>
<td>20-200</td>
</tr>
<tr>
<td>Shale</td>
<td>1-500</td>
</tr>
<tr>
<td>Porous limestone</td>
<td>100-1,000</td>
</tr>
<tr>
<td>Dense limestone</td>
<td>1,000-1,000,000</td>
</tr>
<tr>
<td>Metamorphic rocks</td>
<td>50-1,000,000</td>
</tr>
<tr>
<td>Igneous rocks</td>
<td>100-1,000,000,000</td>
</tr>
<tr>
<td>Oil</td>
<td>0.1</td>
</tr>
<tr>
<td>Water</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Geophysical prospecting

Diagram showing a transmitter loop and a receiver loop, with induced eddy currents at progressively later times after turnoff.
Geophysical prospecting

TDEM-forward
Geophysical prospecting

TDEM-data
The problem

Given some measurements $d_i$ of the fields $u_i(x)$ recover the model function $m(x)$ given

$$A(m)u_i - q_i = 0 \quad i = 1, \ldots, N$$

$$d_i = Q u_i + \text{noise}$$
Solution through optimization

Solve by minimizing

\[ \min \frac{1}{2} \sum \| Qu_i - d_i \|^2 + \alpha R(|\nabla m|) \]

s.t. \[ A(m)u_i - q_i = 0 \quad i = 1, \ldots, N \]

\[ A(m) \] Maxwell operator
\[ m \] conductivity
\[ Q \] projection
\[ u \] fields
\[ R \] regularization
Geophysical prospecting - applications

Hydrology

Oil exploration

Reservoir monitoring

Mineral exploration
Part I - The forward problem
Part II - The inverse problem
Maxwell’s equations

\[ \nabla \times \mu^{-1} \nabla \times \vec{E} + \sigma \vec{E}_t = \vec{s} \text{ on } \Omega \]
\[ \vec{n} \times \vec{E} = 0 \text{ on } \partial \Omega \]

\( \vec{E} \)  electric field
\( \sigma \)  conductivity - usually jumpy
\( \mu \)  magnetic permeability
\( \vec{s} \)  source
Maxwell’s equations

\[ \nabla \times \mu^{-1} \nabla \times \vec{E} + \sigma \vec{E}_t = \vec{s} \quad \text{on } \Omega \]
\[ \vec{n} \times \vec{E} = 0 \quad \text{on } \partial \Omega \]

\( \vec{E} \)  electric field
\( \sigma \)  conductivity - usually jumpy
\( \mu \)  magnetic permeability
\( \vec{s} \)  source

Challenge: 
(Numerically) solve the system in a scalable fashion
OcTree discretization

Needed

- Easy to mesh and use
- Adjust the grid to local smoothness
- Deal with large padding (infinite domains)
- Quick assembly of the matrix
- Mimicking properties
OcTree discretization

Needed

- Easy to mesh and use
- Adjust the grid to local smoothness
- Deal with large padding (infinite domains)
- Quick assembly of the matrix
- Mimicking properties

Possibilities

- Finite elements
  Complicated geometries but large meshing time
- Finite volume
  Short meshing time but simple geometries
- Finite volume OcTrees
  Not so simple geometries and short meshing time
OcTree vs FEM discretization

OcTree discretization for Poisson like (Edwards 96, Ewing, Lazarov & Vassilevski 91, Losasso, Fedkiw & Osher 06)
OcTree discretization for Maxwell (Lipnikov, Morel & Shashkov 04, H. & Heldmann 06)
Discretization of Maxwell's equations

Use implicit time stepping method (usually BDF2)

\[ \nabla \times \mu^{-1} \nabla \times \vec{E} + \sigma \alpha \vec{E} = \vec{s} \]

View in variational form

\[
\min \int_{\Omega} \frac{1}{2\mu} |\nabla \times \vec{E}|^2 + \frac{\sigma \alpha}{2} |\vec{E}|^2 - \vec{E} \cdot \vec{s} \ dx
\]
OcTree discretization of Maxwell’s equation

- Similar to FEM we discretize the variation principle
- Unlike FEM we use finite difference and midpoint/trapezoidal method
- In 1D Varga, Fix & Strang
OcTree discretization of the curl and mass matrix

\[
(\nabla \times \vec{E})_z^2 = (\partial_y \vec{E}_x - \partial_x \vec{E}_y)^2
\]

Use only short differences and averages \(O(h^2)\)
OcTree discretization of the forward problem

Discrete approximation

\[
\int_{\Omega} \frac{1}{2\mu} |\nabla \times \vec{E}|^2 + \frac{\sigma\alpha}{2} |\vec{E}|^2 - \vec{E} \cdot \vec{s} \, dx = \\
\frac{1}{2} \mathbf{e}^\top (A + \alpha M) \mathbf{e} - \mathbf{e}^\top \mathbf{s} + \mathcal{O}(h^2)
\]

**Theorem:** Our discretization yields

\[
(\mathbf{e} - P\vec{E})^\top (A + \alpha M)(\mathbf{e} - P\vec{E}) = \mathcal{O}(h^2)
\]
Solution of the linear system

\[(A + \alpha M)e = s\]

- Can be difficult to solve due to the null space of the curl \(\nabla \times \nabla = 0 \iff AD^\top = 0\)
- Use (discrete) Helmholtz decomposition
  \[e = a + D^\top \phi\]
  \[0 = Da\] Culomb gauge condition

- Obtain a stable system
  \[
  \begin{pmatrix}
  A + D^\top D + \alpha M & \alpha MD^\top \\
  \alpha DM & DMD^\top
  \end{pmatrix}
  \begin{pmatrix}
  a \\
  \phi
  \end{pmatrix}
  =
  \begin{pmatrix}
  s \\
  Ds
  \end{pmatrix}
  \]

- Multigrid preconditioner  Ascher & Aruliah 02, H. & Ascher 01
<table>
<thead>
<tr>
<th>$N$</th>
<th>$N = 16^3$</th>
<th>$N = 32^3$</th>
<th>$N = 64^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1/\mu_{bg}$</td>
<td>$\sigma_1/\sigma_{bg}$</td>
<td>iterations</td>
<td>$\mu_1/\mu_{bg}$</td>
</tr>
<tr>
<td>$10^1$</td>
<td>$10^2$</td>
<td>9</td>
<td>$10^1$</td>
</tr>
<tr>
<td>$10^2$</td>
<td>$10^4$</td>
<td>10</td>
<td>$10^2$</td>
</tr>
<tr>
<td>$10^1$</td>
<td>$10^2$</td>
<td>11</td>
<td>$10^1$</td>
</tr>
</tbody>
</table>
Part II - The inverse problem
Inverse problem through optimization

\[
\min_{m,u} \quad \frac{1}{2} \sum_i \|Qu_i - d_i\|^2 + \alpha R(|\nabla m|) \, dx \\
\text{s.t.} \quad A(m)u_i - q_i = 0 \quad i = 1, \ldots, N
\]

Comments

- Use \( R = \int \rho(|\nabla m|) \, dx \)
- Choose \( \rho(t) = \text{Huber}(t, \theta) \) to obtain discontinuities
- Regularization parameter \( \alpha \) needs to be determined
The discrete optimization problem

Discretize then Optimize

\[
\begin{align*}
\min_{m,u} & \quad \frac{1}{2} \sum \|Qu_i - d_i\|^2 + \alpha R(|\nabla_h m|) \\
\text{s.t.} & \quad A(m)u_i - q_i = 0 \quad i = 1, \ldots, N
\end{align*}
\]

Solve using reduced space SQP

- Major cost, forward and adjoint
- Fine mesh → large scale
- Bad conditioning but well posed!
Grid sequencing

- Major idea - solve on a sequence of grids (Moře 02, Ascher & H 01, Nash & Sofer 01, Borzi & Kunish 03, Benzi Hanson & H 06 & many more)
- For smooth parameter estimation with FE (Bangerth 04, Becker, Becker Kapp 03 & Rannacher 03)
- Need very few iterations on the finest grid
- Potentially avoid local minima
Why local refinement?

- The cost of the optimization is dominated by the size of the forward problem
- For the forward problem, a factor of 10 reduction can be obtained
- For solutions with large gradients, "zoom in" on the discontinuity
Discretization of the objective function

\[ R(\|\nabla m\|) = \sum_{\text{cells}} \int_{\text{cell}} \rho(\|\nabla m\|) \, d\mathbf{x} \]

Evaluate

\[ \int_{\text{cell}} \rho(\|\nabla m\|) \, d\mathbf{x} = \int_{\text{cell}} \rho \left( \sqrt{m_{x_1}^2 + m_{x_2}^2} \right) \, d\mathbf{x} \]

\[ \frac{\partial m}{\partial x_1}(x_0) = \frac{m_2 + m_3 - 2m_1}{3h} + \mathcal{O}(h) \]
\[
\int_{\text{cell}_j} \rho(|\nabla m|) = \int_{\text{cell}_j} \rho \left( \sqrt{m_{x_1}^2 + m_{x_2}^2} \right) \, d\mathbf{x} \approx
\]

\[
V_{\text{cell}} \rho \left\{ \sqrt{\frac{1}{2} \left( \frac{m_2 + m_3 - 2m_1}{3h} \right)^2 + \frac{1}{2} \left( \frac{m_4 - m_1}{2h} \right)^2} + \text{aprx} \, (m_{x_2})^2 \right\}
\]

\[
R(\mathbf{m}) = \mathbf{v}^\top \rho \left( \sqrt{A^c_f(\nabla_{hc} \mathbf{m})^2} \right).
\]
Solving the discrete problem

Given an OcTree evaluate $m, u$ by solving the optimization problem.
Solving the discrete problem

Given an OcTree evaluate $m, u$ by solving the optimization problem.

The Euler Lagrange equations:

$$A(m)^\top \lambda_i = G_i^\top (d_i - Q_i u_i), \quad i = 1, \ldots, N$$
$$A(m) u_i = q_i, \quad i = 1, \ldots, N$$
$$\sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h^\top \Sigma(m) \nabla_h m = 0,$$
Solving the discrete problem

Work within the reduced problem
Solving the discrete problem

Work within the reduced problem

Use L-BFGS and rank 2 update Quasi-Newton methods for the solution of the problem
Solving the discrete problem

Work within the reduced problem

Use L-BFGS and rank 2 update Quasi-Newton methods for the solution of the problem

The solution can have large smooth areas. Use Adaptive Multilevel Refinement
Adaptive Multilevel Refinement (AMR)

- The cost of the optimization process is impacted by the size of the problem and initial guess.
- Adaptive multilevel refinement methods achieve a low-cost good starting guess.
- AMR reduce the size of the discrete fine grid problem.
Guidelines for Adaptive Multilevel Refinement

\[ A(m)^\top \lambda_i = Q_i^\top (d_i - Q_i u_i), \quad i = 1, \ldots, N \]
\[ A(m) u_i = q_i, \quad i = 1, \ldots, N \]
\[ \sum_{k} G(m, u_k)^\top \lambda_k + \alpha \nabla_h^\top \Sigma(m) \nabla_h m = 0, \]

We solve for \( u_j, m, \lambda_j \) on grids \( S_{u_j}, S_m, S_{\lambda_j} \).
Guidelines for Adaptive Multilevel Refinement

\[ A(m)^\top \lambda_i = Q_i^\top (d_i - Q_i u_i), \quad i = 1, \ldots, N \]
\[ A(m)u_i = q_i, \quad i = 1, \ldots, N \]
\[ \sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla h^\top \Sigma(m) \nabla h m = 0, \]

We solve for \( u_j, m, \lambda_j \) on grids \( S_{u_j}, S_m, S_{\lambda_j} \).

- If \( S_{u_j} \neq S_{\lambda_j} \) then the forward operator is not the discrete adjoint of the operator for \( \lambda_j \).
Guidelines for Adaptive Multilevel Refinement

\[ A(m)^\top \lambda_i = Q_i^\top (d_i - Q_i u_i), \quad i = 1, \ldots, N \]
\[ A(m) u_i = q_i, \quad i = 1, \ldots, N \]
\[ \sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h^\top \Sigma(m) \nabla_h m = 0, \]

We solve for \( u_j, m, \lambda_j \) on grids \( S_{u_j}, S_m, S_{\lambda_j} \).

- If \( S_{u_j} \neq S_{\lambda_j} \) then the forward operator is not the discrete adjoint of the operator for \( \lambda_j \).

Set the grids \( S_{u_j} = S_{\lambda_j} \).
Guidelines for Adaptive Multilevel Refinement

\[ A(m)^T \lambda_i = Q_i^T (d_i - Q_i u_i), \quad i = 1, \ldots, N \]
\[ A(m) u_i = q_i, \quad i = 1, \ldots, N \]
\[ \sum_k G(m, u_k)^T \lambda_k + \alpha \nabla_h^T \Sigma(m) \nabla_h m = 0, \]

We solve for \( u_j, m, \lambda_j \) on grids \( S_{u_j}, S_m, S_{\lambda_j} \).

- If \( S_{u_j} \neq S_{\lambda_j} \) then the forward operator is not the discrete adjoint of the operator for \( \lambda_j \).

Set the grids \( S_{u_j} = S_{\lambda_j} \)

- If \( S_m \) is finer than \( S_{u_j} \) then homogenization is needed.
Guidelines for Adaptive Multilevel Refinement

\[ A(m)^\top \lambda_i = Q_i^\top (d_i - Q_i u_i), \quad i = 1, \ldots, N \]
\[ A(m)u_i = q_i, \quad i = 1, \ldots, N \]
\[ \sum_{k} G(m, u_k)^\top \lambda_k + \alpha \nabla_h^\top \Sigma(m) \nabla_h m = 0, \]

We solve for \( u_j, m, \lambda_j \) on grids \( S_{u_j}, S_m, S_{\lambda_j} \).

- If \( S_{u_j} \neq S_{\lambda_j} \) then the forward operator is not the discrete adjoint of the operator for \( \lambda_j \).

Set the grids \( S_{u_j} = S_{\lambda_j} \)

- If \( S_m \) is finer than \( S_{u_j} \) then homogenization is needed.

Set \( S_m \subseteq S_{u_j} \).
Refinement criteria for $m$

We minimize

$$\frac{1}{2} \sum \|Qu_i - d_i\|^2 + \alpha R(m)$$

Based on the evaluation of the integral $R(m)$

The controversy: we may need to refine $m$ in regions where $u$ changes very slowly
Initializing and refining the grid for $u$

Rule of thumb: the $u$ grid must be fine enough to represent the data $d$.

If the $u$ grid is too coarse then $d = QA(m)^{-1}q$ is biased by numerical errors $\rightarrow$ large errors in $m$. 
Numerical Experiments

The model and experiment
Reconstruction

True model

$16^3$ U-grid

$32^3$ OT grid

$64^3$ OT grid
<table>
<thead>
<tr>
<th>Level</th>
<th>$u$ grid</th>
<th>$m$ grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>5656</td>
<td>4096</td>
</tr>
<tr>
<td>L2</td>
<td>10356</td>
<td>6014</td>
</tr>
<tr>
<td>L3</td>
<td>21342</td>
<td>16199</td>
</tr>
</tbody>
</table>

The number of unknowns on the finest grid is roughly 12 times smaller than the number of unknowns we would obtain by using the full $64^3$ grid.
Conclusions

- By using AMR we can gain a factor in computational time.
- Allows to deal with complex geometries and infinite domains.
- AMR requires special discretization.
- Different grids for $m$, $u$, $\lambda$.
- Can we learn about regularization effects (bias) using AMR?
Conclusions

- By using AMR we can gain a factor in computational time
Conclusions

- By using AMR we can gain a factor in computational time
- Allows to deal with complex geometries and infinite domains
Conclusions

- By using AMR we can gain a factor in computational time
- Allows to deal with complex geometries and infinite domains
- AMR requires special discretization
Conclusions

- By using AMR we can gain a factor in computational time
- Allows to deal with complex geometries and infinite domains
- AMR requires special discretization
- Different grids for $m$ and $u, \lambda$
Conclusions

- By using AMR we can gain a factor in computational time
- Allows to deal with complex geometries and infinite domains
- AMR requires special discretization
- Different grids for $m$ and $u, \lambda$
- Can we learn about regularization effects (bias) using AMR?