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

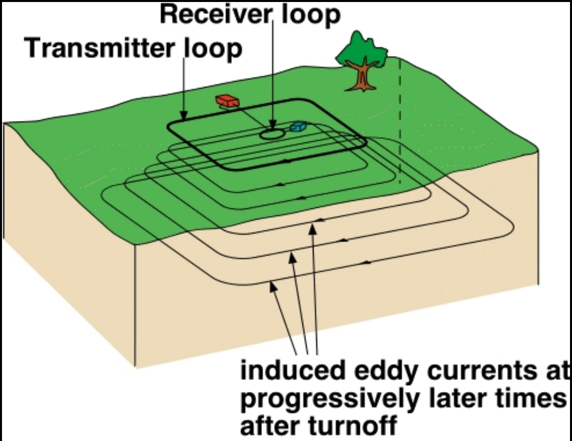
Eldad Haber
contributors: Stefan Heldmann, Jan Modersitzki,
Uri Ascher, Doug Oldenburg, Rosemary Knight

Motivation - Geophysics

Rocks have different electrical properties

Rock	Resistivity (Ω/m)
Clay	1-20
Sand wet to moist	20-200
Shale	1-500
Porous limestone	100-1,000
Dense limestone	1,000-1,000,000
Metamorphic rocks	50-1,000,000
Igneous rocks	100-1,000,000
Oil	0.1
Water	0.05

Geophysical prospecting



Geophysical prospecting

TDEM-forward

Geophysical prospecting

TDEM-data

The problem

Given some measurements $\mathbf{d}_i$ of the fields $u_i(\mathbf{x})$ recover the model function $m(\mathbf{x})$ given

$$A(m)u_i - q_i = 0 \quad i = 1, \ldots, N$$
$$\mathbf{d}_i = Qu_i + \text{noise}$$

Solution through optimization

Solve by minimizing

$$\begin{aligned} \min \quad & \frac{1}{2}\sum \|Qu_i - d_i\|^2 + \alpha R(|\nabla m|) \\ \text{s.t.} \quad & A(m)u_i - q_i = 0 \quad i = 1, \ldots, N \end{aligned}$$

$A(m)$	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

$$
\begin{aligned}
\nabla \times \mu^{-1} \nabla \times \vec{E} + \sigma \vec{E}_t &= \vec{s} \quad \text{on } \Omega \\
\vec{n} \times \vec{E} &= 0 \quad \text{on } \partial\Omega
\end{aligned}
$$

$\vec{E}$	electric field
σ	conductivity - usually jumpy
μ	magnetic permeability
$\vec{s}$	source

Maxwell's equations

$$\begin{aligned} \nabla \times \mu^{-1} \nabla \times \vec{E} + \sigma \vec{E}_t &= \vec{s} \quad \text{on } \Omega \\ \vec{n} \times \vec{E} &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

$\vec{E}$	electric field
σ	conductivity - usually jumpy
μ	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

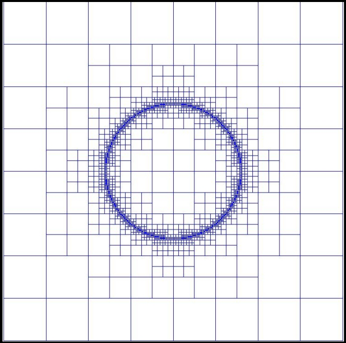
Needed

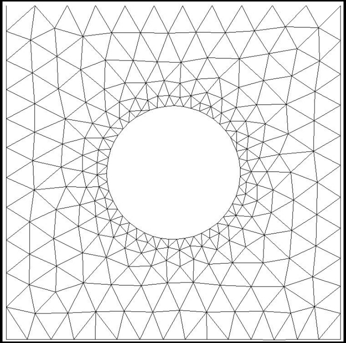
- Easy to mesh and use
- Adjust the grid to local smoothness
- Deal with large padding (infinite domains)
- Quick assembly of the matrix
- Mimicking properties

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

$$\min \int_{\Omega} \frac{1}{2\mu} |\nabla \times \vec{E}|^2 + \frac{\sigma \alpha}{2} |\vec{E}|^2 - \vec{E} \cdot \vec{s} \; dx$$

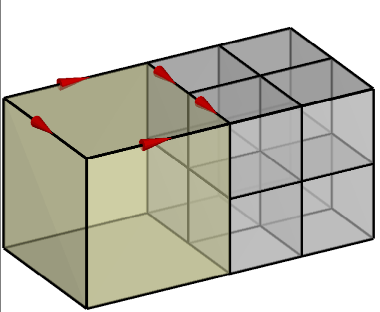
OcTree discretization of Maxwell's equation

- Similar to FEM we discretize the variation principle

- Unlike FEM we use finite difference and midpoint/trapezoidal method
- In 1D Varga 62

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 $\mathcal{O}(h^2)$

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

$$(A + \alpha M)\mathbf{e} = \mathbf{s}$$

- Can be difficult to solve due to the null space of the **curl** $\nabla \times \nabla = 0 \leftrightarrow AD^\top = 0$
- Use (discrete) Helmholtz decomposition

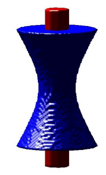
$$\begin{aligned} \mathbf{e} &= \mathbf{a} + D^\top \phi \\ 0 &= D\mathbf{a} \quad \text{Culomb gauge condition} \end{aligned}$$

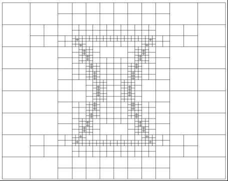
- Obtain a stable system

$$\begin{pmatrix} A + D^\top D + \alpha M & \alpha M D^\top \\ \alpha D M & D M D^\top \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \phi \end{pmatrix} = \begin{pmatrix} \mathbf{s} \\ D\mathbf{s} \end{pmatrix}$$

- Multigrid preconditioner Ascher & Aruliah 02, H. & Ascher 01

Example





$N = 16^3$		
μ_1/μ_{bg}	σ_1/σ_{bg}	iterations
10^1	10^2	9
10^2	10^4	10

$N = 32^3$		
μ_1/μ_{bg}	σ_1/σ_{bg}	iterations
10^1	10^2	10
10^2	10^4	12

$N = 64^3$		
μ_1/μ_{bg}	σ_1/σ_{bg}	iterations
10^1	10^2	11
10^2	10^4	14

Part II - The inverse problem

Inverse problem through optimization

$$\min_{m,u} \quad \frac{1}{2}\sum_i \|Qu_i - \mathbf{d}_i\|^2 + \alpha R(|\nabla m|)\, d\mathbf{x}$$
$$\text{s.t.} \quad A(m)u_i - q_i = 0 \qquad i = 1, \ldots, N$$

Comments

- Use $R = \int \rho(|\nabla m|)dx$
- Choose $\rho(t) = \mathrm{Huber}(t, \theta)$ to obtain discontinuities
- Regularization parameter α needs to be determined

The discrete optimization problem

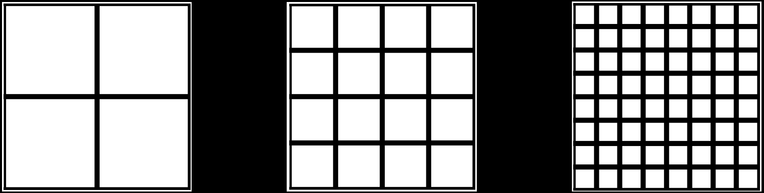
Discretize then Optimize

$$\min_{m,u} \quad \frac{1}{2}\sum \|Qu_i - \mathbf{d}_i\|^2 + \alpha R(|\nabla_h m|)$$
$$\text{s.t.} \quad A(m)u_i - q_i = 0 \qquad i = 1, \ldots, N$$

Solve using reduced space SQP

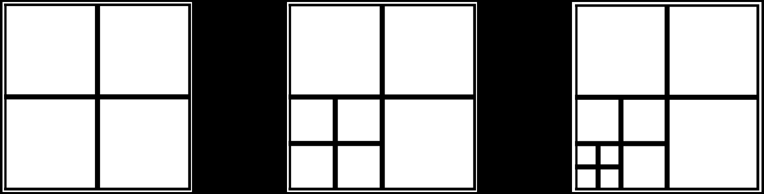
- Major cost, forward and adjoint
- Fine mesh $\rightarrow$ large scale
- Bad conditioning but well posed!

Grid sequencing



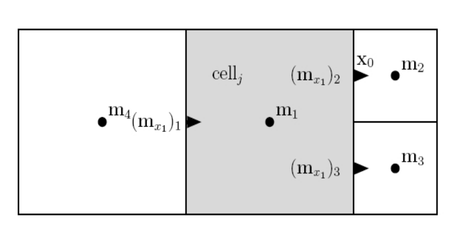
- Major idea - solve on a sequence of grids (Morè 02, Ascher & H 01, Nash & Sofer 01, Borzi & Kunish 03, Benzi Hanson & H 06 & many more)
- For smooth parameter estimation with FE (Bangerth 04, Becker, Becker Kapp 03 & Rannacher 03)
- Need very few iterations on the finest grid
- Potentially avoid local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

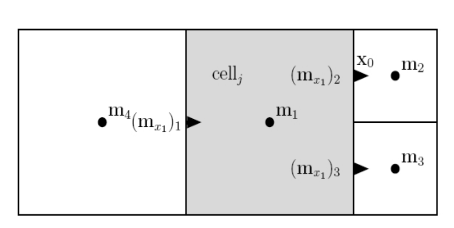


$$R(|\nabla m|) = \sum_{\text{cells}} \int_{\text{cell}} \rho(|\nabla m|)\, d\mathbf{x}$$

Evaluate

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

$$\frac{\partial m}{\partial x_1}(\mathbf{x}_0) = \frac{m_2 + m_3 - 2m_1}{3h} + \mathcal{O}(h)$$



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

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

$$R(\mathbf{m}) = \mathbf{v}^\top \rho\left(\sqrt{\mathbf{A}_f^c(\nabla_{hc}\mathbf{m})^2}\right).$$

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:

$$
\begin{aligned}
&A(m)^\top \lambda_i = Q_i^\top (\mathbf{d}_i - Q_i u_i), \quad i = 1, \dots, N \\
&A(m) u_i = q_i, \quad i = 1, \dots, N \\
&\sum_k G(m, u_k)^\top \lambda_k + \alpha \, \nabla_h{}^\top \, \Sigma(m) \, \nabla_h \, m = \mathbf{0},
\end{aligned}
$$

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

$$A(m)^\top \lambda_i = Q_i^\top(\mathbf{d}_i - Q_i u_i), \quad i = 1, \ldots, N$$
$$A(m) u_i = q_i, \quad i = 1, \ldots, N$$
$$\sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h{}^\top \Sigma(m) \nabla_h m = \mathbf{0},$$

We solve for u_j, m, λ_j on grids $\mathcal{S}_{u_j}$, $\mathcal{S}_m$, $\mathcal{S}_{\lambda_j}$.

Guidelines for Adaptive Multilevel Refinement

$$A(m)^\top \lambda_i = Q_i^\top (\mathbf{d}_i - Q_i u_i), \quad i = 1, \ldots, N$$
$$A(m) u_i = q_i, \quad i = 1, \ldots, N$$
$$\sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h{}^\top \Sigma(m) \nabla_h m = \mathbf{0},$$

We solve for u_j, m, λ_j on grids $\mathcal{S}_{u_j}$, $\mathcal{S}_m$, $\mathcal{S}_{\lambda_j}$.

- If $\mathcal{S}_{u_j} \neq \mathcal{S}_{\lambda_j}$ then the forward operator is not the discrete adjoint of the operator for λ_j.

Guidelines for Adaptive Multilevel Refinement

$$A(m)^\top \lambda_i = Q_i^\top (\mathbf{d}_i - Q_i u_i), \quad i = 1, \ldots, N$$
$$A(m) u_i = q_i, \quad i = 1, \ldots, N$$
$$\sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h{}^\top \Sigma(m) \nabla_h m = \mathbf{0},$$

We solve for u_j, m, λ_j on grids $\mathcal{S}_{u_j}$, $\mathcal{S}_m$, $\mathcal{S}_{\lambda_j}$.

- If $\mathcal{S}_{u_j} \neq \mathcal{S}_{\lambda_j}$ then the forward operator is not the discrete adjoint of the operator for λ_j.

Set the grids $\mathcal{S}_{u_j} = \mathcal{S}_{\lambda_j}$

Guidelines for Adaptive Multilevel Refinement

$$A(m)^\top \lambda_i = Q_i^\top (\mathbf{d}_i - Q_i u_i), \quad i = 1, \dots, N$$
$$A(m) u_i = q_i, \quad i = 1, \dots, N$$
$$\sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h{}^\top \Sigma(m) \nabla_h m = \mathbf{0},$$

We solve for u_j, m, λ_j on grids $\mathcal{S}_{u_j}$, $\mathcal{S}_m$, $\mathcal{S}_{\lambda_j}$.

- If $\mathcal{S}_{u_j} \neq \mathcal{S}_{\lambda_j}$ then the forward operator is not the discrete adjoint of the operator for λ_j.

  Set the grids $\mathcal{S}_{u_j} = \mathcal{S}_{\lambda_j}$

- If $\mathcal{S}_m$ is finer than $\mathcal{S}_{u_j}$ then homogenization is needed.

Guidelines for Adaptive Multilevel Refinement

$$A(m)^\top \lambda_i = Q_i^\top (\mathbf{d}_i - Q_i u_i), \quad i = 1, \dots, N$$
$$A(m) u_i = q_i, \quad i = 1, \dots, N$$
$$\sum_k G(m, u_k)^\top \lambda_k + \alpha \nabla_h{}^\top \Sigma(m) \nabla_h m = \mathbf{0},$$

We solve for u_j, m, λ_j on grids $\mathcal{S}_{u_j}$, $\mathcal{S}_m$, $\mathcal{S}_{\lambda_j}$.

- If $\mathcal{S}_{u_j} \neq \mathcal{S}_{\lambda_j}$ then the forward operator is not the discrete adjoint of the operator for λ_j.

  Set the grids $\mathcal{S}_{u_j} = \mathcal{S}_{\lambda_j}$

- If $\mathcal{S}_m$ is finer than $\mathcal{S}_{u_j}$ then homogenization is needed.

  Set $\mathcal{S}_m \subseteq \mathcal{S}_{u_j}$.

Refinement criteria for m

We minimize

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

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

The controversy: we may need to refine m in regions where u changes very slowly

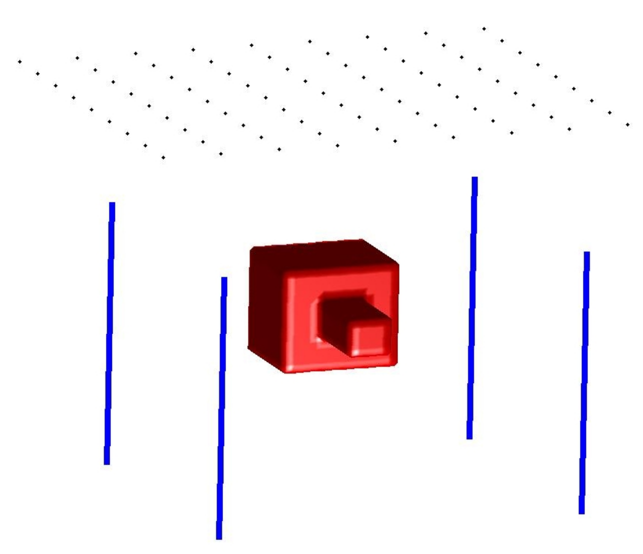
Initializing and refining the grid for u

Rule of thumb: *the u grid must be fine enough to represent the data $\mathbf{d}$*

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

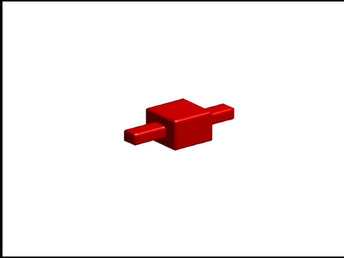
Numerical Experiments

The model and experiment

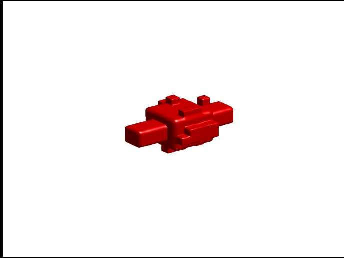


Reconstruction

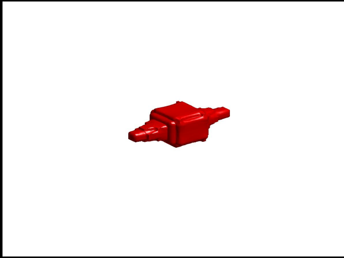
True model



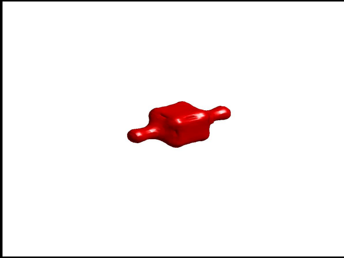
16^3 U-grid



32^3 OT grid



64^3 OT grid



Grids

Level	u grid	m grid
L1	5656	4096
L2	10356	6014
L3	21342	16199

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

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, λ

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, λ
- Can we learn about regularization effects (bias) using AMR?