Application of compressive sensing ideas to Static Electromagnetic Geophysics

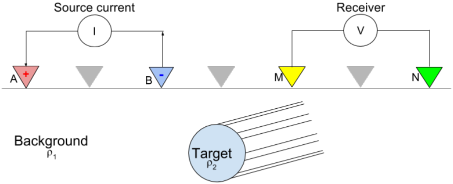
Thibaut Astic

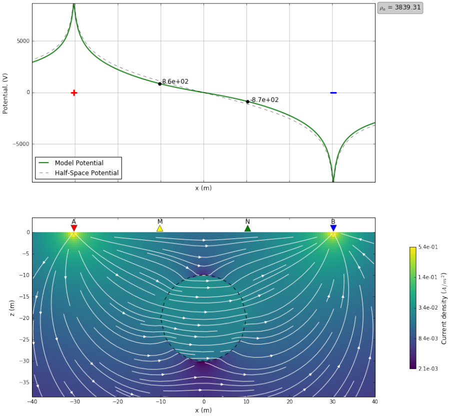
EOSC 513 2017: Project
https://github.com/thast/EOSC513

Outline

Outline

DC Resistivity Survey





Currents are injected to the earth. Theirs flow will be distorted depending upon the conductivity contrast in the earth These changes can be measurable on the surface electrodes. Interactive notebooks

DCR Physics: Maxwell's equations

conservation of charge

current density

input current

dirac delta

$$\nabla \cdot \vec{j} = I\left(\delta(r - r_{s^+}) - \delta(r - r_{s^-})\right)$$

$$= q$$

electrodes

electrical conductivity

static! usually: $-\frac{\partial \vec{b}}{\partial t}$

electric field

$$\vec{j} = \sigma \vec{e}$$

Ohm's law

$$\nabla \times \vec{e} = 0$$

Faraday's law (static)

anything

$$\nabla \times \nabla a = 0$$

vector identity

downhill

$$\vec{e} = -\nabla \phi$$

potential

source

1st order form

$$\nabla \cdot \vec{j} = q \quad \text{(a)}$$

scalar equation

$$\frac{1}{\sigma} \vec{j} = -\nabla \phi \quad \text{(b)}$$

vector equation

2nd order form

$$\nabla \cdot \sigma\left(-\nabla \phi\right) = q$$

Rowan Cockett, Lindsey J. Heagy, and Douglas W. Oldenburg (2016). "Pixels and their neighbors: Finite volume." The Leading Edge, 35(8), 703–706. doi: 10.1190/tle35080703.1

DCR: inverse Problems

- SimPEG: open-source python package for EM Geophysics
- discretize using a Finite Volume Formulation

$$\mathrm{diag}(\mathbf{v})\mathbf{D}\mathbf{M}_f(\sigma^{-1})^{-1}\mathbf{D}^\top\mathrm{diag}(\mathbf{v})\boldsymbol{\phi} = \mathbf{q}. \tag{1}$$

- Solve for $m = \ln(\sigma)$ to ensure positivity of conductivity
- Use a beta-cooling strategy for the regularizer's weight

$$f(m) = \frac{1}{2}\sum_{i=1}^{N_s} ||P_i A(m)^{-1} q_i - d_i||_2^2 + \beta R(m) \tag{2}$$

$$f(m) = \frac{1}{2}||PA(m)^{-1}Q - D||_F^2 + \beta R(m) \tag{3}$$

Usually:

$$R(m) = \frac{1}{2}||W(m - m_{ref})||_2^2 \tag{4}$$

DCR: Solving the inverse problem

- solve a linearized problem
- Inexact Gauss-Newton steps

$$f(m) = \frac{1}{2}||r(m)||_2^2 + \beta||W(m - m_{ref})||_2^2 \tag{5}$$

$$f(m + \delta m) = \frac{1}{2}||r(m) + J(m)\delta m||_2^2 + \beta||W(m + \delta m - m_{ref})||_2^2 \tag{6}$$

$$(J^T J + \beta W^T W)\delta m = -J^T r(m) - \beta W^T W(m - m_{ref}) \tag{7}$$

Update m: $m_{k+1} = m_k + \alpha\delta m$ s.t $f(m_{k+1}) < \gamma f(m_k)$

DCR: inverse Problems Results

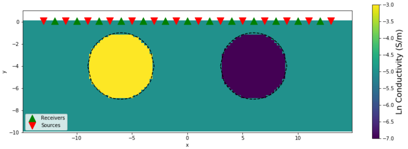


Figure: Initial model and survey

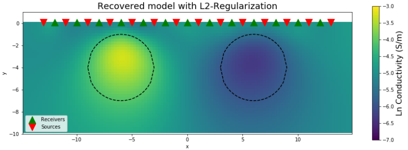


Figure: Recovered model with L2 Regularization

Problematic

- The cost of inverting DC data is mainly driven by the number of sources:
  - Can we decrease this cost? What is the price to pay?

- The diffusive nature of the physics behind DCR and the L2-Regularization are promoting smooth update
  - Can we, in some space, promote sparsity for such a problem?

Outline

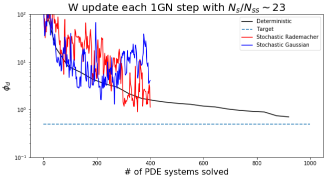
Reducing the number of Sources

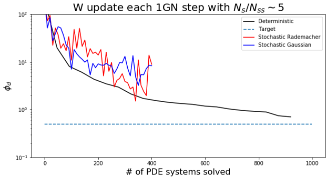
We start to try one of the basic idea of Compressive Sensing by using simultaneous sources.

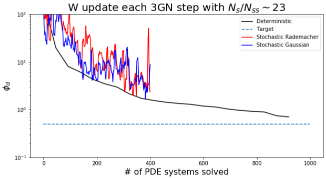
$$f(m) = \frac{1}{2}||PA(m)^{-1}QW - DW||_F^2 + \beta R(m) \tag{8}$$

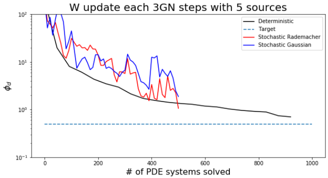
W a random gaussian matrix of size $N_s \times N_{ss}$ with $N_{ss} << N_s$
The Gauss-Newton step subproblem's size is then reduced by a factor $\frac{N_s}{N_{ss}}$
We have then experiment several strategies for updating W.

Updating W


W update each 1GN step with $N_s/N_{ss} \sim 23$
Deterministic
Target
Stochastic Rademacher
Stochastic Gaussian
ϕ_d
# of PDE systems solved


W update each 1GN step with $N_s/N_{ss} \sim 5$
Deterministic
Target
Stochastic Rademacher
Stochastic Gaussian
ϕ_d
# of PDE systems solved


W update each 3GN step with $N_s/N_{ss} \sim 23$
Deterministic
Target
Stochastic Rademacher
Stochastic Gaussian
ϕ_d
# of PDE systems solved


W update each 3GN steps with 5 sources
Deterministic
Target
Stochastic Rademacher
Stochastic Gaussian
ϕ_d
# of PDE systems solved

Outline

Formulating the GN subproblem

Original Data fitting subproblem:

$$\min_{\delta m} \frac{1}{2} ||J(m)\delta m + r(m)||_2^2 + \beta ||W(m + \delta m - m_{ref})||_2^2 \quad (9)$$

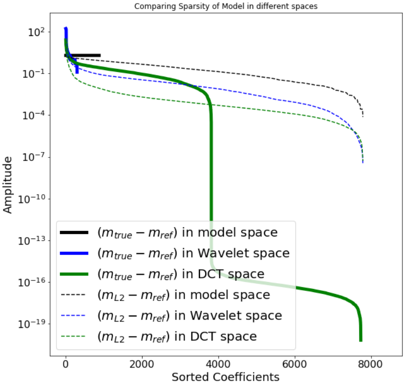
Now without the L2 penalty but a L1 constrain in some space:

$$\min ||\delta x||_1 \text{ s.t } ||J(m) S^H \delta x + r(m)||_2^2 < tol \quad (10)$$

- $\delta x = S \delta m$
- S a sparsifying matrix
- S^H the inverse transform.
- solved using SPGL1
  E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions", SIAM J. on Scientific Computing, 31(2):890-912, November 2008

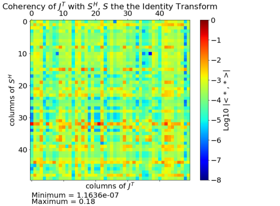
Choosing a sparsifying transform

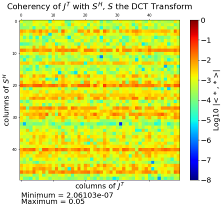
- The model to recover should be sparse in some space

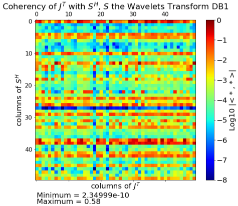


Choosing a sparsifying transform

- the columns of J^T and S^H should be incoherent
- The DC problem is a weighted Laplacian
- The eigenfunctions of the Laplacian are cosine functions

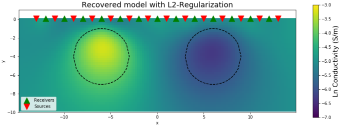


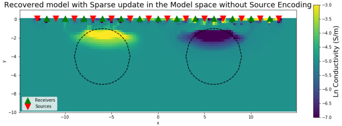


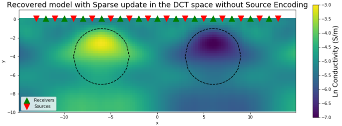


Recovered Models using sparse GN update: S = Identity or DCT

- Without Source Encoding

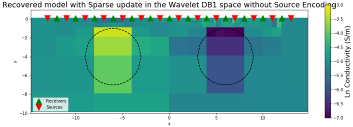


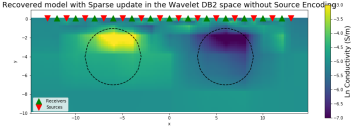


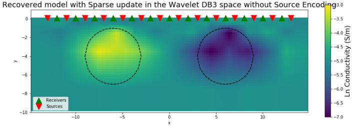


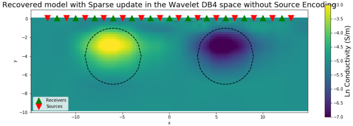
Recovered Models using sparse GN update in Wavelets space

- Without Source Encoding

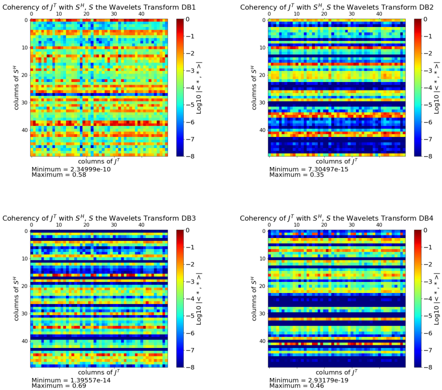








Coherency of J and Wavelet transform

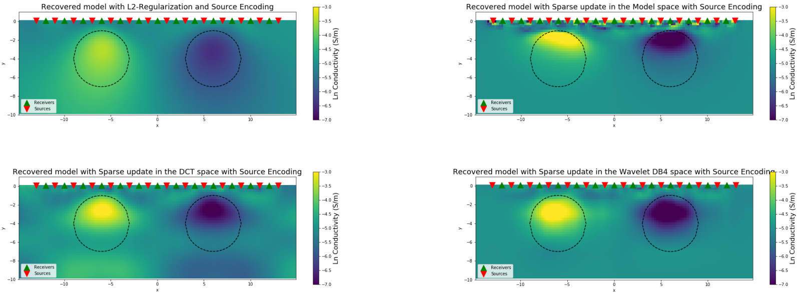


Possible explanation (hypothesis): asymptotic Incoherence

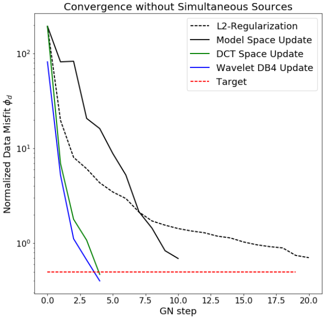
Adcock, Ben, et al. "Breaking the coherence barrier: A new theory for compressed sensing." Forum of Mathematics, Sigma. Vol. 5. Cambridge University Press, 2017.

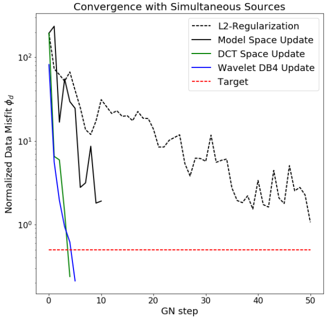
Recovered Models using sparse GN update and Simultaneous Sources

- With Source Encoding



Convergence Comparison

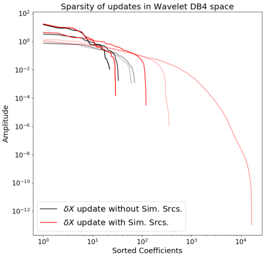


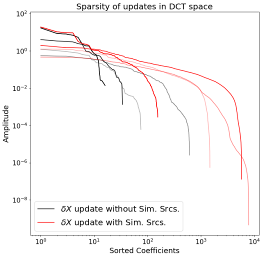


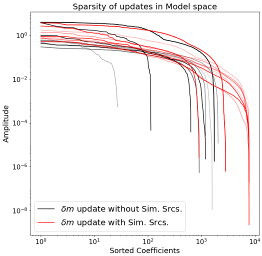
- L2 is still faster as each SPGL1 GN update is much more expansive to compute
- SPGL1 GN update seems even harder with Sim. Srcs!

A curious behaviour

- Simulateneous Sources is killing the sparsity







Outline

Discussion and Conclusion

- Simultaneous Sources
  - In terms of number of PDE solved for L2, the overall convergence rate is similar or worse
  - each step is a lot less memory-expansive, Useful if we have too many sources for our memory
  - for SPGL1, it has appeared to make things even worse, with longer time necessary to compute each GN update

Discussion and Conclusion

- Sparse GN Update
  - At first, antagonization between sparsity in model space and the diffusive behavior of the Jacobian
  - Sparse update in model space are expansive to solve, sparse in DCT space is easier but recovered model is still smooth and display periodic patterns
  - Surprisingly, Wavelets have been an efficient transform despite a somehow high maximum for coherency
  - Simultaneous Sources has been a killer both for sparsity and computation times during our experiment
  - We did not study much the influence of the tolerance for the data fitting term in SPGL1 or the strategy to set it. Considering the time required for each iteration and the few iterations required to reach the target, we have probably been quite strict.

Thank you!

All scripts and more are available on Github/thast/

Bonus: Another way to recover a compact model through prior information and vector quantization: