3D CSEM inversion using a Parallel Discrete Adjoint method

Oscar Peredo, Vladimir Puzyrev, Jelena Koldan, José M. Cela

{oscar.peredo, vladimir.puzyrev, jelena.koldan, josem.cela}@bsc.es

Introduction

We have developed a framework to deploy 3D CSEM inversion using massively parallel nodal finite element forward simulations based on secondary Coulomb-gauged EM potentials. The core of our implementation is based in the discrete adjoint method which builds gradients of a misfit function with respect to the electric conductivity of each nodal point of the mesh.

Discrete adjoint method

The *discrete adjoint method* [3], is a versatile and powerful technique to obtain gradients in PDE-constrained optimization problems. Formally, a PDE-constrained optimization problem (after discretization) can be formulated as:

$$\underset{(\mathbf{u},\mathbf{d})\in\mathbb{K}^{n_u}\times\mathbb{K}^{n_d}}{\text{minimize}} \quad J(\mathbf{u},\mathbf{d}) \qquad \text{subject to} \quad \mathbf{r}(\mathbf{u},\mathbf{d}) = \mathbf{0}$$

with $\mathbb{K}\in\{\mathbb{R},\mathbb{C}\}$, $J:\mathbb{K}^{n_u}\times\mathbb{K}^{n_d}\to\mathbb{R}$ cost and $\mathbf{r}:\mathbb{K}^{n_u}\times\mathbb{K}^{n_d}\to\mathbb{K}^{n_u}$ constraint functions, $\mathbf{r}(\mathbf{u},\mathbf{d}) := \mathbf{K}(\mathbf{d})\mathbf{u}-\mathbf{f}(\mathbf{d})$. Considering the implicit dependency $\mathbf{u}(\mathbf{d}) = [\mathbf{K}(\mathbf{d})]^{-1}\mathbf{f}(\mathbf{d})$ ($[\mathbf{K}(\mathbf{d})]^{-1}$ is never calculated), a new unconstrained problem is defined:

$$\underset{\mathbf{d}\in\mathbb{K}^{n_d}}{\text{minimize}} \quad j(\mathbf{d}) := J(\mathbf{u}(\mathbf{d}),\mathbf{d})$$

which can be solved using gradient-based methods. The steps to build $\nabla_{\mathbf{d}} j(\mathbf{d}_k)$ are:

1. Set initial value of $\mathbf{d}_k$;
2. $\mathbf{u}_k \leftarrow \mathbf{r}(\mathbf{u},\mathbf{d}_k) = \mathbf{0}$ (*forward problem*);
3. Calculate[a] $\nabla_{\mathbf{u}} J(\mathbf{u}_k,\mathbf{d}_k)$, $\nabla_{\mathbf{u}}\mathbf{r}(\mathbf{u}_k,\mathbf{d}_k)$;
4. $\boldsymbol{\lambda}_k \leftarrow \nabla_{\mathbf{u}}\mathbf{r}(\mathbf{u}_k,\mathbf{d}_k)^*\boldsymbol{\lambda} = \nabla_{\mathbf{u}} J(\mathbf{u}_k,\mathbf{d}_k)^*$ (*adjoint problem*);
5. Calculate[a] $\nabla_{\mathbf{d}} J(\mathbf{u}_k,\mathbf{d}_k)$, $\nabla_{\mathbf{d}}\mathbf{r}(\mathbf{u}_k,\mathbf{d}_k)$;
6. $\nabla_{\mathbf{d}} j(\mathbf{d}_k) = -\boldsymbol{\lambda}_k^*\nabla_{\mathbf{d}}\mathbf{r}(\mathbf{u}_k,\mathbf{d}_k) + \nabla_{\mathbf{d}} J(\mathbf{u}_k,\mathbf{d}_k)$;

[a] using automatic differentiation, finite differences or taking derivatives *by hand*

Forward problem

Described in [5] following ideas from [1], it is formulated in terms of $(\mathbf{A}_s,\nabla\phi_s)$ with homogeneous Dirichlet b.c. in a 3D domain Ω as:

$$\nabla^2\mathbf{A}_s + i\omega\mu_0\overline{\sigma}(\mathbf{A}_s+\nabla\phi_s) = -i\omega\mu_0\delta\overline{\sigma}(\mathbf{A}_p+\nabla\phi_p)$$
$$\nabla\cdot(i\omega\mu_0\overline{\sigma}(\mathbf{A}_s+\nabla\phi_s)) = -\nabla\cdot(i\omega\mu_0\delta\overline{\sigma}(\mathbf{A}_p+\nabla\phi_p))$$

with electrical conductivity $\overline{\sigma} := \overline{\sigma}_{base}+\delta\overline{\sigma}$, angular frequency $\omega = 2\pi f$ and magnetic permeability of free space μ_0. After finite element discretization using N nodes, the forward problem is: $\mathbf{K}(\mathbf{d})\mathbf{u} = \mathbf{f}(\mathbf{d})$ with $\mathbf{d}$ the discrete values of $\delta\overline{\sigma}$ in the domain, $\mathbf{u} = (\mathbf{A}_s^1,\mathbf{A}_s^2,\mathbf{A}_s^3,\phi_s)^T \in\mathbb{C}^{4N}$, $\mathbf{K}(\mathbf{d})\in\mathbb{C}^{4N\times 4N}$ and $\mathbf{f}(\mathbf{d})\in\mathbb{C}^{4N}$.

Adjoint problem

Cost (misfit) function:

$$J(\mathbf{u},\mathbf{d}) = \overline{(\mathbf{D}(\mathbf{u}-\mathbf{u}^{obs}))}^T\mathbf{D}(\mathbf{u}-\mathbf{u}^{obs})$$

Adjoint problem (isotropy assumed in $\overline{\sigma}$, i.e. $d = N$):

$$\mathbf{K}(\mathbf{d})^T\overline{\boldsymbol{\lambda}} = \overline{\mathbf{D}(\mathbf{u}-\mathbf{u}^{obs})}$$

Gradient:

$$\nabla_{\mathbf{d}} j(\mathbf{d}) = -2\Re\left\{\overline{\boldsymbol{\lambda}}^T\nabla_{\mathbf{d}}\mathbf{r}(\mathbf{u},\mathbf{d})\right\}$$

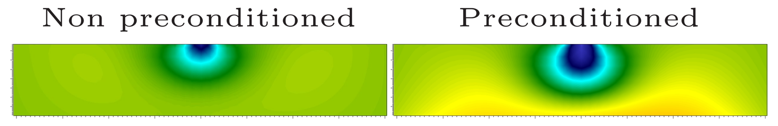
Log transformation $\gamma_i = \ln(\overline{\sigma}_{base}+d_i)$:

$$\nabla_\gamma j(\gamma) = -2\Re\left\{\overline{\boldsymbol{\lambda}}^T\nabla_{\mathbf{d}}\mathbf{r}(\mathbf{u},\mathbf{d})\right\}\cdot\nabla_\gamma\mathbf{d}(\gamma)$$

Features

Current:

- $\mathbf{D}_{ii} = \text{distance}(\text{Tx},\text{Rx}_i)^\alpha$, $\alpha > 0$.
- Depth-based Gradient preconditioner:



- Multi-Tx support: $\nabla_\gamma j(\gamma) = \sum_k \nabla_\gamma j_k(\gamma)$.

Future work:

- New preconditioners.
- Regularization in misfit function [2].
- Misfit function based in $(\mathbf{E},\mathbf{H})$ fields using moving least squares interpolation of $(\mathbf{A},\phi)$ potentials.

References

[1] E. A. Badea, M. E. Everett, G. A. Newman, and O. Biro. Finite-element analysis of controlled-source electromagnetic induction using Coulomb-gauged potentials. *Geophysics*, 66(3):786–799, 2001.

[2] M. Commer and G. A. Newman. New advances in three-dimensional controlled-source electromagnetic inversion. *Geophys. J. Int.*, 172:513–535, 2008.

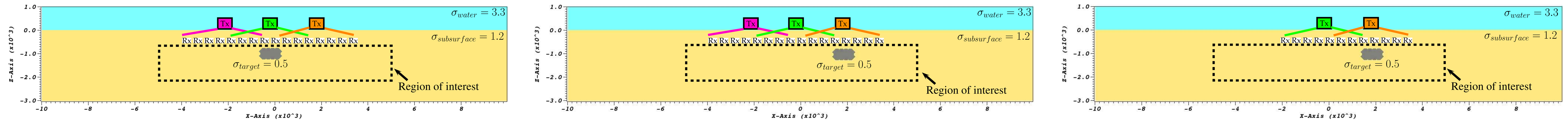
[3] M. B. Giles and N. A. Pierce. Adjoint equations in CFD: duality, boundary conditions, and solution behaviour. In *13th AIAA Computational Fluid Dynamics Conference*, number AIAA–97–1850, Snowmass Village, CO, June 1997.

[4] G. Houzeaux, M. Vázquez, R. Aubry, and J. M. Cela. A massively parallel fractional step solver for incompressible flows. *Journal of Computational Physics*, 228(17):6316 – 6332, 2009.

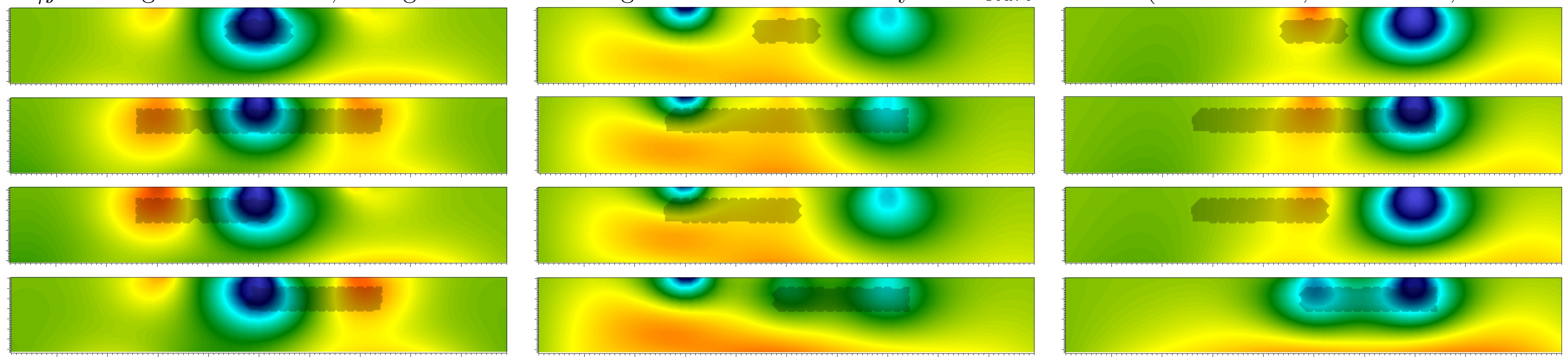
[5] V. Puzyrev, J. Koldan, J. de la Puente, G. Houzeaux, M. Vázquez, and J. M. Cela. A parallel finite-element method for 3D controlled-source electromagnetic forward modeling. *Geophys. J. Int.*, 193(2):678–693, 2013.

Does $-\nabla_\gamma j$ gives us some useful information at $f = 0.5$Hz?

Target models with anomaly at $\sigma_{target} = 0.5$ and depth 800 meters:



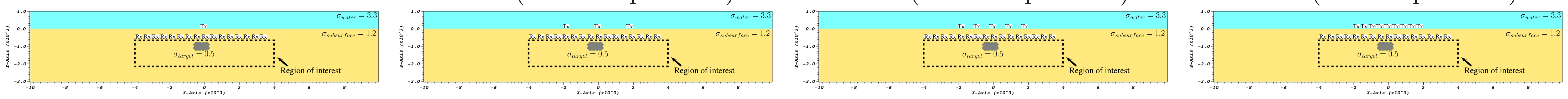
$-\nabla_\gamma j$ on region of interest, using different starting models with anomaly at $\sigma_{start} = 1.1$ (scale: , *blue* < 0, *red* > 0):



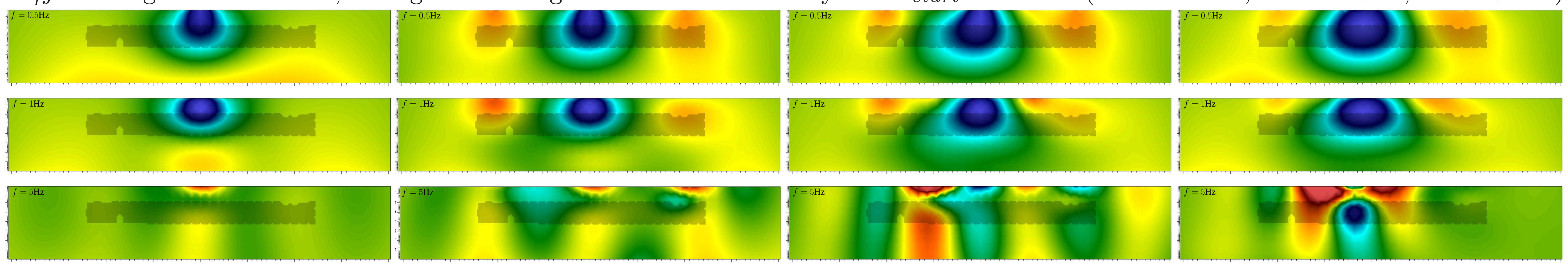
How does $-\nabla_\gamma j$ looks at different frequencies using several Tx's?

Target model with anomaly at $\sigma_{target} = 0.5$ and depth 800 meters:

1 Tx | 3 Tx (2000m separation) | 5 Tx (1000m separation) | 9 Tx (500m separation)



$-\nabla_\gamma j$ on region of interest, using a starting model with anomaly at $\sigma_{start} = 1.1$ (scale: , *blue* < 0, *red* > 0):



Forward/Adjoint solver

Alya (multi-physics parallel PDE solver [4]):

- Fortran 90 / MPI+OpenMP
- Mesh partitioning with METIS
- Sparsity: iterative methods
- Portable: runs in IBM PowerPC, Intel Xeon/SandyBridge, ...
- Big mesh → almost linear scalability
- Integration with well-known post-processors: VisIt, Paraview, GiD, ...

Gradient implementation

INPUT: $\boldsymbol{\gamma}$, $\mathbf{d} := \mathbf{d}(\boldsymbol{\gamma})$;

$\mathbf{u} \leftarrow \texttt{solve}(\mathbf{K}(\mathbf{d}),\mathbf{f}(\mathbf{d}))$[a];

$\boldsymbol{\lambda} \leftarrow \texttt{solve}(\mathbf{K}(\mathbf{d})^T,\overline{\mathbf{D}(\mathbf{u}-\mathbf{u}^{obs})})$;

for $node_i = 1:N$ **do**

  $\frac{\partial j}{\partial\gamma_i} = 0$;

  if $node_i \in$ *Region-Of-Interest* **then**

    $\texttt{assemble}\left(\frac{\partial\mathbf{K}}{\partial\mathbf{d}_i},\frac{\partial\mathbf{f}}{\partial\mathbf{d}_i}\right)$;

    $\frac{\partial\mathbf{r}}{\partial\mathbf{d}_i} = \frac{\partial\mathbf{K}}{\partial\mathbf{d}_i}\mathbf{u} - \frac{\partial\mathbf{f}}{\partial\mathbf{d}_i}$;

    $\frac{\partial j}{\partial\gamma_i} = -2\Re\left\{\overline{\boldsymbol{\lambda}}^T\frac{\partial\mathbf{r}}{\partial\mathbf{d}_i}\right\}\frac{\partial\mathbf{d}_i}{\partial\gamma_i}$;

OUTPUT: $\nabla_\gamma j(\gamma)$

[a] In red we show the operations performed in parallel.

We calculate $\nabla_\gamma j$ in each node of the domain, using two mesh sizes of 1.1M and 4.8M elements (177K and 812K nodes respectively) with MPI-only processes:

#CPUs	N	Time (hh:mm) IBM PPC 970MP 4 cores-per-node 2.3GHz	Time (hh:mm) Intel Xeon E5-2670 16 cores-per-node 2.6GHz
128	177K	00:12	00:06
256		00:07	00:04
512		00:03	00:02
128	812K	04:12	01:48
256		02:18	01:10
512		01:15	00:27

↑ cores-per-node ⟹ ↓ time. Suitable for new multiprocessor chips (> 50 cores).