DOMAIN DECOMPOSITION AND GRADIENT DESCENT METHODS FOR THE CONDUCTIVITY INVERSE PROBLEM

Tuan Anh VU*

*PhD Student at CMAP, École Polytechnique

15$^{\text{th}}$ October 2020

Content

Content

Short lines about me

Full name:	Tuan Anh VU
	Call me Anh (⌣̈)
Nationality:	Vietnamese

Progress:

2019 - 2020	**M2 Student at Sorbonne University**
	Internship with Houssem & Marcella
10/2020	**PhD Student at École Polytechnique**
	One-shot inversion methods and domain decomposition
	Supervisors: Houssem & Marcella
	In collaboration with Lorenzo

Content

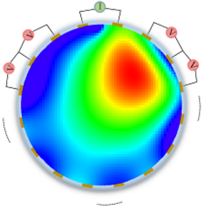
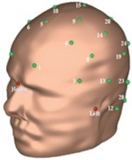
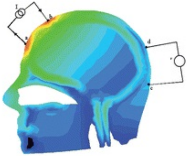
Direct problem

- Direct problem (forward problem):

$$\begin{cases} \mathbf{div}(\sigma \nabla u) &= 0, \quad \Omega \subset \mathbb{R}^d, \\ \sigma \dfrac{\partial u}{\partial \nu} &= g, \qquad \partial \Omega; \end{cases}$$

  Known: $\sigma : \Omega \to \mathbb{R}$: *the conductivity*, $g : \partial\Omega \to \mathbb{R}$: *the current*;
  Unknown: $u : \Omega \to \mathbb{R}$: *the total field*.
  We consider $\sigma \mapsto u(\sigma)$.

- Application: Electrical Impedance Tomography (EIT).

Inverse conductivity problem

Recall:

$$\begin{cases} \mathbf{div}(\sigma \nabla u) = 0, & \Omega \subset \mathbb{R}^d, \\ \sigma \dfrac{\partial u}{\partial \nu} = g, & \partial \Omega. \end{cases}$$

- Inverse conductivity problem:

  Know g, measure $f = u(\sigma)\big|_{\partial \Omega}$, find σ?

  Fact: Ill-posed problem! $\rightarrow$ Work with suitable assumptions.

- Preparation: Use *least-square method & gradient descent* to solve.
  Require: Compute the gradient of cost function at σ?

$\Longrightarrow$ Lagrangian technique!

- Lagrangian = "Cost function" + Variational formula for u;
- From Lagrangian: Each state $u(\sigma)$ has an *adjoint state* $p(\sigma)$ defined by an equation *($\rightarrow$ adjoint problem)*;
- Gradient of cost function = Formula$(u(\sigma), p(\sigma))$.

General goal

- Solve the inverse problem using *least-square method*:
  - Compute u and its adjoint p at current σ^n.
    Small problem? $\rightarrow$ Direct solver!
    Large problem? $\rightarrow$ Iterative solver: *domain decomposition*.
  - Update σ^{n+1} from σ^n using *gradient descent*.

  We do one iteration for u and one iteration for σ.

$\rightarrow$ Combine two iterations?

$\rightarrow$ Spoil: One-shot inversion method.

To-do list for internship

We work with simplified model:

Ω and its subdomains: circles co-centered at the origin ($d = 2$);

σ: piecewise constant function w.r.t the subdomains;

$g(\phi) = \cos(n\phi), n \in \mathbb{N}^*$.

- Theoretical results:
  - Calculate some analytical results (u and p, cost function and its gradient, etc);
  - Study the convergence of DDM and its speed.
- Implementation on FreeFEM:
  - Solve the inverse problem using direct solver for u and p;
  - Solve the inverse problem using DDM for u and p;
  - Compare FreeFEM computations with the analytical results.

Content

Recall problem - Apply least-square method

- Recall:

$$\begin{cases} \mathbf{div}(\sigma \nabla u) &= 0, \quad \Omega \subset \mathbb{R}^d, \\ \sigma \dfrac{\partial u}{\partial \nu} &= g, \quad \partial \Omega. \end{cases}$$

Know g, measure $f = u(\sigma)\big|_{\partial \Omega}$, find σ?

- Setting: exact conductivity σ^*; measure $f = u(\sigma^*)\big|_{\partial \Omega}$; cost function

$$J(\sigma) = \frac{1}{2} \| u(\sigma) - f \|^2_{L^2(\partial \Omega)} = \frac{1}{2} \int_{\partial \Omega} |u(\sigma) - f|^2.$$

Apply gradient descent

Strategy for gradient descent:

- Solve the forward and adjoint problems (using suitable solver):
  - State $u(\sigma)$: $\int_\Omega \sigma \nabla u(\sigma) \cdot \nabla w = \int_{\partial\Omega} gw, \forall w,$
  - Adjoint state $p(\sigma)$: $\int_\Omega \sigma \nabla p(\sigma) \cdot \nabla w = \int_{\partial\Omega} (f - u(\sigma))w, \forall w;$

**Remark: Forward and adjoint problems have the same bilinear form!*

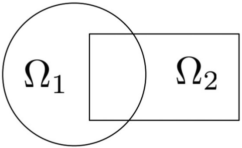
- Gradient descent: $\sigma^{n+1} = \sigma^n - \tau \nabla J(\sigma^n)$ where

$$\nabla J(\sigma) : \quad \langle \nabla J(\sigma), h \rangle = \int_\Omega h \nabla u(\sigma) \cdot \nabla p(\sigma), \forall h.$$

Content

Why domain decomposition?

- Small forward problem? $\rightarrow$ Direct solver: robust but much memory!
  Large forward problem? $\rightarrow$ Iterative solver!
  We choose *domain decomposition methods (DDM)*: divide large domains into small subdomains & propose suitable subproblems for them.



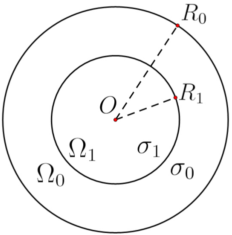
- Advantages of DDM:
  - Apply direct solver on each subproblem;
  - Exchange information between subdomains and iteratively update the solution.
  - Parallel computing.

DDM for one-inner-circle case (1/5) - Model

- "One-inner-circle case": For $0 < R_1 < R_0$,

$$\Omega = B_{R_0}, \Omega_0 = \{R_1 < r < R_0\} =: \mathrm{ann}(R_1, R_0) \text{ and } \Omega_1 = B_{R_1}.$$

- Piecewise constant function $\sigma = \sigma_0.\mathbb{1}_{\Omega_0} + \sigma_1.\mathbb{1}_{\Omega_1} > 0$.



- Forward problem:

$$\begin{cases} \mathbf{div}(\sigma \nabla u) & = & 0, & B_{R_0}, \\ \sigma \dfrac{\partial u}{\partial \nu} & = & g(\phi), & \partial B_{R_0}. \end{cases}$$

DDM for one-inner-circle case (2/5) - A choice of DDM

Non-overlapping DDM: initial u_0^0 and u_1^0;

$$\begin{cases} \mathbf{div}(\sigma_0 \nabla u_0^{k+1}) & = & 0, & \Omega_0 = \mathrm{ann}(R_1, R_0), \\ \sigma_0 \dfrac{\partial\, u_0^{k+1}}{\partial\, \nu} & = & g(\phi), & \partial\, B_{R_0}, \\ \left(\sigma_0 \dfrac{\partial}{\partial\, \nu} + \alpha\right) u_0^{k+1} & = & \left(\sigma_1 \dfrac{\partial}{\partial\, \nu} + \alpha\right) u_1^k, & \partial\, B_{R_1}; \end{cases}$$

$$\begin{cases} \mathbf{div}(\sigma_1 \nabla u_1^{k+1}) & = & 0, & \Omega_1 = B_{R_1}, \\ \left(\sigma_1 \dfrac{\partial}{\partial\, \nu} + \alpha\right) u_1^{k+1} & = & \left(\sigma_0 \dfrac{\partial}{\partial\, \nu} + \alpha\right) u_0^k, & \partial\, B_{R_1} \end{cases}$$

where $\alpha > 0$ is a constant (Robin parameter).

**Remark: ν is outward-pointing unit normal vector to Ω_j in the related problem, $j = 1, 2$.*

DDM for one-inner-circle case (3/5) - Convergence

- Consider the errors

$$e_j^k = u_j^k - u\big|_{\Omega_j}, j = 1, 2; k \geq 0.$$

- Use Fourier series to show that

$$e_0^k = a_0^{(k)} + \sum_{n\in\mathbb{Z}\backslash\{0\}} a_n^{(k)} \left(r^{|n|} + R_0^{2n} r^{-|n|}\right) \mathrm{e}^{\mathrm{i}n\phi};$$

$$e_1^k = b_0^{(k)} + \sum_{n\in\mathbb{Z}\backslash\{0\}} b_n^{(k)} r^{|n|} \mathrm{e}^{\mathrm{i}n\phi}$$

where

$$n \neq 0, k \geq 1: \quad a_n^{(k+1)} = \tilde{\rho}(|n|;\alpha) a_n^{(k-1)} \quad \text{(same for } b\text{)};$$
$$n = 0, k \geq 1: \quad a_0^{(k+1)} = b_0^{(k)} = a_0^{(k-1)} \quad \text{(same for } b\text{)}.$$

$\implies$ "Geometric sequence" when $n \neq 0$!

- Show that $|\tilde{\rho}(|n|;\alpha)| < 1, \forall n \neq 0, \forall \alpha > 0$; suitable $u_j^0 \implies$ Convergence!

DDM for one-inner-circle case (4/5) - Convergence factor

**Remark 1: $\tilde{\rho}(|n|;\alpha)$ connects $(k+1)$-th term and $(k-1)$-th term.*

**Remark 2: Simplify notation $|n|, n \in \mathbb{Z}\backslash\{0\} \to n \in \mathbb{N}^*$.*

For $n \in \mathbb{N}^*, \alpha > 0$: Convergence factor

$$\rho(n;\alpha) = \sqrt{|\tilde{\rho}(n;\alpha)|} < 1$$

where

- $\tilde{\rho}(n;\alpha) = \dfrac{\alpha - p(n)}{\alpha + p(n)} \cdot \dfrac{\alpha - q(n)}{\alpha + q(n)}$,
- $p(n) = \dfrac{\sigma_1 n}{R_1} > 0$,
- $q(n) = \dfrac{\sigma_0 n}{R_1} \cdot \dfrac{R_0^{2n} R_1^{-n} - R_1^n}{R_0^{2n} R_1^{-n} + R_1^n} = \dfrac{\sigma_0 n}{R_1} \cdot \dfrac{(R_0/R_1)^{2n} - 1}{(R_0/R_1)^{2n} + 1} > 0 \quad (R_0 > R_1 > 0)$.

DDM for one-inner-circle case (5/5) - Speed

- Fact: Solution is approximated up to certain frequency in Fourier series.
- Given N the maximal frequency, if we choose

$$\beta = \frac{\sqrt{\max\{p(N), q(N)\}} - \sqrt{\max\{p(1), q(1)\}}}{\sqrt{\max\{p(N), q(N)\}} + \sqrt{\max\{p(1), q(1)\}}}$$

and

$$\alpha = \frac{1-\beta}{1+\beta} \max\{p(N), q(N)\}$$

then

$$\max_{1 \leq n \leq N} |\tilde{\rho}(n; \alpha)| \leq \beta,$$

that is,

$$\max_{1 \leq n \leq N} \rho(n; \alpha) \leq \sqrt{\beta} = \sqrt{\frac{\sqrt{\max\{p(N), q(N)\}} - \sqrt{\max\{p(1), q(1)\}}}{\sqrt{\max\{p(N), q(N)\}} + \sqrt{\max\{p(1), q(1)\}}}}.$$

Content

One-shot inversion method

- Before:

  Solve the inverse problem using least-square method:
  - Compute u and its adjoint p at current σ^n.

    Large problem? $\rightarrow$ Iterative solver: domain decomposition.
  - Update σ^{n+1} from σ^n using gradient descent.

  $\rightarrow$ Combine iterations for u and for σ?

- One-shot inversion method: iterates at the same time on
  - the solution of forward problem,
  - the unknown of inverse problem.

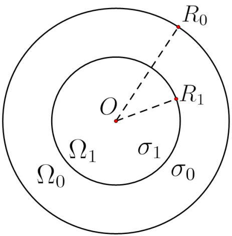
- Fact:
  - Not new
  - Introduced in some linear/moderate non-linear inverse problems
  - Convergence? Assume strong convexity of cost function...

Try to apply one-shot inversion method - Example (1/2)

- Problem: One-inner-circle (two subdomains Ω_0, Ω_1).
  Forward problem:

$$\begin{cases} \mathbf{div}(\sigma \nabla u) &= 0, & B_{R_0}, \\ \sigma \dfrac{\partial u}{\partial \nu} &= g(\phi), & \partial B_{R_0}. \end{cases}$$

Piecewise constant function $\sigma = \sigma_0 . \mathbb{1}_{\Omega_0} + \sigma_1 . \mathbb{1}_{\Omega_1} > 0$.



- Fix σ_0 in Ω_0, find σ_1 in Ω_1 $\rightarrow$ One variable σ_1!

Try to apply one-shot inversion method - Example (2/2)

Indexes: n: gradient descent step, m: DDM step.
Purpose: Build gradient descent $\{\sigma^n\}_n$ for σ_1?

- Normal step n: Already know σ^n
  - Use suitable solver to solve for exact u^n and p^n;
  - Compute exact gradient at σ^n using u^n, p^n;
  - Update σ^{n+1} from σ^n.
- Step n with one-shot: Already know σ^n
  - Do only M steps of DDM for u and p, detail:
    Temporary functions $v_0^m, v_1^m, w_0^m, w_1^m$ where $0 \leq m \leq M$;
    Initial: $v_0^0 = u_0^{n-1}, v_1^0 = u_1^{n-1}, w_0^0 = p_0^{n-1}, w_1^n = p_1^{n-1}$;
    Do M steps for DDM, receive $v_0^M, v_1^M, w_0^M, w_1^M$;
    Save $u_0^n = v_0^M$, $u_1^n = v_1^M$, $p_0^n = w_0^M$, $p_1^n = w_1^M$.
  - Compute non-exact gradient at σ^n using $u_0^n, u_1^n, p_0^n, p_1^n$;
  - Update σ^{n+1} from σ^n.

Hope: Incomplete iteration for u, p but still have convergence of $\{\sigma^n\}_n$!

Outlook

Fact:

Noisy data $\rightarrow$ Unstable problem $\rightarrow$ GD cannot converge...

$\Longrightarrow$ Should stop when:

- a sufficient accuracy is obtained;
- the solution of the inverse problem does not blow up.

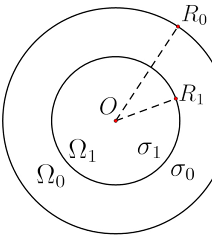
Noisy data + Incomplete iterations on direct and adjoint problems?

$\Longrightarrow$ Try to analyze the convergence of the scheme for noisy data with early stopping rule.

Content

Implementation: Some numerical examples

I will share my screen with you, thank you for watching!



- Fix σ_0, find σ_1?
- Info:

  $R_0 = 1, R_1 = 0.8$;

  $\sigma_0 = 1, \sigma_1 = 10$, guess (for σ_1) = 8;

  $g(\phi) = \cos(4\phi)$.
- E.x. Gradient descent with direct solver; Test combined iterations.

Content

Plan

- Long-time plan:

  ▶ Try some DDMs, ensure the convergence before combining iterations for one-shot. Analyze the convergence of some one-shot methods for simplified case, then general case;

  ▶ Try for linear problem, then non-linear problems.

- Recently:

  ▶ Study numerically the convergence of combined system by adjusting the number of DDM's iterations in each gradient descent step. Worth to do more DDM's iterations?

  ▶ Analyze the convergence of some one-shot methods for simplified model.

End of presentation

♡ Thank you for your attention ♡