Inverse Source Localization: Complete Mathematical Formulation

Part 3: Inverse Problem Formulations

Mathematical Reference Document
Version 7.23

January 2026

Abstract

This document presents the two main approaches to the inverse source localization problem: the linear formulation (distributed sources on a fixed grid) with L1, L2, and Total Variation regularization, and the nonlinear formulation (continuous point source positions). Complete derivations of optimality conditions and solution algorithms are provided.

Contents

1 The Inverse Problem

1.1 Problem Statement

Given: Noisy boundary measurements $\tilde{\boldsymbol{u}} \in \mathbb{R}^{N_b}$:

$$\tilde{\boldsymbol{u}} = \boldsymbol{u}^{\text{true}} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}) \tag{1}$$

Find: Source locations $\{\boldsymbol{z}_k\}$ and intensities $\{q_k\}$ that generated $\boldsymbol{u}^{\text{true}}$.

1.2 Two Formulations

1. **Linear (Distributed Sources):** Fix candidate source locations on a grid. Solve for intensities only.
   - Unknowns: $\boldsymbol{q} \in \mathbb{R}^M$ (intensities at M grid points)
   - Problem: Convex optimization
   - Sources discovered via sparsity-promoting regularization
2. **Nonlinear (Point Sources):** Solve for both positions and intensities.
   - Unknowns: $\boldsymbol{\theta} = (x_1, y_1, q_1, \ldots, x_K, y_K, q_K) \in \mathbb{R}^{3K}$
   - Problem: Non-convex optimization
   - Number of sources K must be specified

2 Linear Inverse Problem

2.1 Setup

2.1.1 Source Grid

Fix M candidate source locations $\{\boldsymbol{\xi}_j\}_{j=1}^M$ inside the domain Ω.

2.1.2 Green's Matrix

Define $\boldsymbol{G} \in \mathbb{R}^{N_b \times M}$ with:

$$G_{ij} = G_N(\boldsymbol{x}_i^{\text{boundary}}, \boldsymbol{\xi}_j) \tag{2}$$

The forward model is:

$$\boldsymbol{u}_{\text{boundary}} = \boldsymbol{G}\boldsymbol{q} \tag{3}$$

2.1.3 Compatibility Constraint

From the physics: $\boldsymbol{1}^T \boldsymbol{q} = \sum_{j=1}^M q_j = 0$.

2.1.4 Ill-Posedness

The linear system $\boldsymbol{G}\boldsymbol{q} = \tilde{\boldsymbol{u}}$ is typically:

- Underdetermined ($M > N_b$): infinitely many solutions
- Ill-conditioned: small noise causes large errors in naive inversion

Regularization is essential.

2.2 L2 Regularization (Tikhonov)

2.2.1 Optimization Problem

$$\boxed{\min_{\boldsymbol{q}\in\mathbb{R}^M} \frac{1}{2}\left\|\boldsymbol{G}\boldsymbol{q}-\tilde{\boldsymbol{u}}\right\|_2^2 + \frac{\alpha}{2}\left\|\boldsymbol{q}\right\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^T\boldsymbol{q}=0} \tag{4}$$

where $\alpha > 0$ is the regularization parameter.

2.2.2 Lagrangian

$$\mathcal{L}(\boldsymbol{q},\lambda) = \frac{1}{2}\left\|\boldsymbol{G}\boldsymbol{q}-\tilde{\boldsymbol{u}}\right\|_2^2 + \frac{\alpha}{2}\left\|\boldsymbol{q}\right\|_2^2 + \lambda\mathbf{1}^T\boldsymbol{q} \tag{5}$$

2.2.3 Optimality Conditions (KKT)

Stationarity:

$$\nabla_{\boldsymbol{q}\mathcal{L}} = \boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{q}-\tilde{\boldsymbol{u}}) + \alpha\boldsymbol{q} + \lambda\mathbf{1} = \mathbf{0} \tag{6}$$

Primal feasibility:

$$\mathbf{1}^T\boldsymbol{q} = 0 \tag{7}$$

2.2.4 Solution Derivation

Step 1: Rearrange stationarity condition.

From (**??**):

$$(\boldsymbol{G}^T\boldsymbol{G}+\alpha\boldsymbol{I})\boldsymbol{q} = \boldsymbol{G}^T\tilde{\boldsymbol{u}} - \lambda\mathbf{1} \tag{8}$$

Define $\boldsymbol{A} = \boldsymbol{G}^T\boldsymbol{G}+\alpha\boldsymbol{I}$. Note: $\boldsymbol{A}$ is symmetric positive definite (SPD) for $\alpha > 0$.

$$\boldsymbol{q} = \boldsymbol{A}^{-1}(\boldsymbol{G}^T\tilde{\boldsymbol{u}} - \lambda\mathbf{1}) \tag{9}$$

Step 2: Apply feasibility constraint.

Substitute (**??**) into (**??**):

$$\begin{aligned} 0 &= \mathbf{1}^T\boldsymbol{q} && (10)\\ &= \mathbf{1}^T\boldsymbol{A}^{-1}(\boldsymbol{G}^T\tilde{\boldsymbol{u}} - \lambda\mathbf{1}) && (11)\\ &= \mathbf{1}^T\boldsymbol{A}^{-1}\boldsymbol{G}^T\tilde{\boldsymbol{u}} - \lambda\mathbf{1}^T\boldsymbol{A}^{-1}\mathbf{1} && (12) \end{aligned}$$

Solving for λ:

$$\boxed{\lambda^* = \frac{\mathbf{1}^T\boldsymbol{A}^{-1}\boldsymbol{G}^T\tilde{\boldsymbol{u}}}{\mathbf{1}^T\boldsymbol{A}^{-1}\mathbf{1}}} \tag{13}$$

Step 3: Final solution.

$$\boxed{\boldsymbol{q}^* = \boldsymbol{A}^{-1}(\boldsymbol{G}^T\tilde{\boldsymbol{u}} - \lambda^*\mathbf{1})} \tag{14}$$

2.2.5 Algorithm

2.3 L1 Regularization (Lasso / Sparsity-Promoting)

2.3.1 Optimization Problem

$$\boxed{\min_{\boldsymbol{q}\in\mathbb{R}^M} \frac{1}{2}\left\|\boldsymbol{G}\boldsymbol{q}-\tilde{\boldsymbol{u}}\right\|_2^2 + \alpha\left\|\boldsymbol{q}\right\|_1 \quad \text{s.t.} \quad \mathbf{1}^T\boldsymbol{q}=0} \tag{15}$$

where $\left\|\boldsymbol{q}\right\|_1 = \sum_j |q_j|$.

Algorithm 1 L2 Regularized Solution

Require: Green's matrix $\boldsymbol{G}$, measurements $\tilde{\boldsymbol{u}}$, regularization α

1: $\boldsymbol{A} \leftarrow \boldsymbol{G}^T\boldsymbol{G} + \alpha \boldsymbol{I}$
2: Solve $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{G}^T\tilde{\boldsymbol{u}}$ for $\boldsymbol{v}$
3: Solve $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{1}$ for $\boldsymbol{w}$
4: $\lambda^* \leftarrow (\boldsymbol{1}^T\boldsymbol{v})/(\boldsymbol{1}^T\boldsymbol{w})$
5: $\boldsymbol{q}^* \leftarrow \boldsymbol{v} - \lambda^*\boldsymbol{w}$
6: **return** $\boldsymbol{q}^*$

2.3.2 Why L1 Promotes Sparsity

Theorem 2.1 (Sparsity of L1 Solutions)**.** *The L1 regularizer promotes sparsity: optimal solutions tend to have many zero components.*

Geometric Intuition. The L1 unit ball $\{\boldsymbol{q} : \|\boldsymbol{q}\|_1 \leq 1\}$ has corners at $\pm\boldsymbol{e}_j$ (standard basis vectors). The level sets of the objective (ellipsoids for quadratic loss) are more likely to touch the L1 ball at corners, yielding sparse solutions. □

2.3.3 Optimality Conditions

The L1 norm is non-differentiable. We use the subdifferential:

$$\partial \|\boldsymbol{q}\|_1 = \{g \in \mathbb{R}^M : g_j \in \partial |q_j|\} \tag{16}$$

where:

$$\partial |q_j| = \begin{cases} \{+1\} & q_j > 0 \\ \{-1\} & q_j < 0 \\ [-1, +1] & q_j = 0 \end{cases} \tag{17}$$

KKT conditions:

$$\boldsymbol{0} \in \boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}) + \alpha\,\partial\|\boldsymbol{q}\|_1 + \lambda\boldsymbol{1} \tag{18}$$

$$0 = \boldsymbol{1}^T\boldsymbol{q} \tag{19}$$

Component-wise: for each j,

$$\begin{cases} (\boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}))_j + \alpha\,\mathrm{sign}(q_j) + \lambda = 0 & \text{if } q_j \neq 0 \\ \left|(\boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}))_j + \lambda\right| \leq \alpha & \text{if } q_j = 0 \end{cases} \tag{20}$$

2.3.4 Solution via CVXPY

In practice, we use convex optimization solvers. The problem (**??**) can be reformulated as a quadratic program:

$$\min_{\boldsymbol{q},\boldsymbol{t}} \quad \frac{1}{2}\|\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}\|_2^2 + \alpha\boldsymbol{1}^T\boldsymbol{t} \tag{21}$$

$$\text{s.t.} \quad -\boldsymbol{t} \leq \boldsymbol{q} \leq \boldsymbol{t} \tag{22}$$

$$\boldsymbol{1}^T\boldsymbol{q} = 0 \tag{23}$$

where $\boldsymbol{t} \geq \boldsymbol{0}$ enforces $t_j \geq |q_j|$ at optimum.

2.4 Total Variation (TV) Regularization

2.4.1 Motivation

TV regularization promotes *piecewise constant* solutions, penalizing spatial variation rather than magnitude.

2.4.2 Discrete Gradient Operator

Definition 2.1 (Gradient Operator)**.** *On a 2D grid or mesh, define $\boldsymbol{D} \in \mathbb{R}^{E\times M}$ where E is the number of edges:*

$$(\boldsymbol{D}\boldsymbol{q})_e = q_j - q_i \tag{24}$$

for edge e connecting nodes i and j.

For a Delaunay mesh, we use edges from the triangulation. For a regular grid:

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_x \\ \boldsymbol{D}_y \end{bmatrix} \tag{25}$$

where $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ are first-difference operators in x and y.

2.4.3 Anisotropic TV

$$\mathrm{TV}(\boldsymbol{q}) = \|\boldsymbol{D}\boldsymbol{q}\|_1 = \sum_e |(\boldsymbol{D}\boldsymbol{q})_e| \tag{26}$$

2.4.4 Optimization Problem

$$\boxed{\min_{\boldsymbol{q}\in\mathbb{R}^M} \frac{1}{2}\|\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}\|_2^2 + \alpha\|\boldsymbol{D}\boldsymbol{q}\|_1 \quad \text{s.t.} \quad \boldsymbol{1}^T\boldsymbol{q} = 0} \tag{27}$$

2.4.5 Solution via ADMM

The Alternating Direction Method of Multipliers (ADMM) is effective for TV problems.

Variable splitting: Introduce $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{q}$:

$$\min_{\boldsymbol{q},\boldsymbol{z}} \quad \frac{1}{2}\|\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}\|_2^2 + \alpha\|\boldsymbol{z}\|_1 \tag{28}$$

$$\text{s.t.} \quad \boldsymbol{D}\boldsymbol{q} = \boldsymbol{z}, \quad \boldsymbol{1}^T\boldsymbol{q} = 0 \tag{29}$$

Augmented Lagrangian:

$$L_\rho(\boldsymbol{q},\boldsymbol{z},\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}\|_2^2 + \alpha\|\boldsymbol{z}\|_1 + \boldsymbol{y}^T(\boldsymbol{D}\boldsymbol{q} - \boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{q} - \boldsymbol{z}\|_2^2 \tag{30}$$

ADMM iterations:

$$\boldsymbol{q}^{k+1} = \operatorname*{argmin}_{\boldsymbol{q}:\boldsymbol{1}^T\boldsymbol{q}=0} \frac{1}{2}\|\boldsymbol{G}\boldsymbol{q} - \tilde{\boldsymbol{u}}\|_2^2 + \frac{\rho}{2}\left\|\boldsymbol{D}\boldsymbol{q} - \boldsymbol{z}^k + \boldsymbol{u}^k\right\|_2^2 \tag{31}$$

$$\boldsymbol{z}^{k+1} = \operatorname*{argmin}_{\boldsymbol{z}} \alpha\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\left\|\boldsymbol{D}\boldsymbol{q}^{k+1} - \boldsymbol{z} + \boldsymbol{u}^k\right\|_2^2 \tag{32}$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{D}\boldsymbol{q}^{k+1} - \boldsymbol{z}^{k+1} \tag{33}$$

where $\boldsymbol{u} = \boldsymbol{y}/\rho$ (scaled dual variable).

***q*-subproblem:** Linear system with constraint:

$$(\boldsymbol{G}^T\boldsymbol{G} + \rho\boldsymbol{D}^T\boldsymbol{D})\boldsymbol{q} = \boldsymbol{G}^T\tilde{\boldsymbol{u}} + \rho\boldsymbol{D}^T(\boldsymbol{z}^k - \boldsymbol{u}^k) - \lambda\boldsymbol{1} \tag{34}$$

where λ enforces $\mathbf{1}^T \boldsymbol{q} = 0$.

***z*-subproblem:** Soft thresholding:

$$z_e^{k+1} = S_{\alpha/\rho}((\boldsymbol{D}\boldsymbol{q}^{k+1})_e + u_e^k) \tag{35}$$

where $S_\tau(x) = \text{sign}(x)\max(|x| - \tau, 0)$ is the soft threshold operator.

2.5 Parameter Selection: L-Curve Method

2.5.1 The L-Curve

For each α, compute the solution $\boldsymbol{q}(\alpha)$ and plot:

- x-axis: $\log_{10} \|\boldsymbol{G}\boldsymbol{q}(\alpha) - \tilde{\boldsymbol{u}}\|_2$ (residual)
- y-axis: $\log_{10} R(\boldsymbol{q}(\alpha))$ (regularizer: $\|\boldsymbol{q}\|_2$, $\|\boldsymbol{q}\|_1$, or $\|\boldsymbol{D}\boldsymbol{q}\|_1$)

The curve typically has an "L" shape:

- **Large α:** Over-regularized, smooth but poor fit
- **Small α:** Under-regularized, good fit but noisy
- **Corner:** Optimal trade-off

2.5.2 Corner Detection

Definition 2.2 (L-Curve Corner)**.** *The optimal α^* is at the point of maximum curvature:*

$$\alpha^* =_\alpha \kappa(\alpha) \tag{36}$$

where κ is the curvature of the L-curve.

In log-log space with $x = \log r$, $y = \log \rho$:

$$\kappa = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}} \tag{37}$$

Practical implementation: Use discrete approximation or the "maximum distance from line" heuristic:

1. Draw line from first to last point on L-curve
2. Find point with maximum perpendicular distance from line

3 Nonlinear Inverse Problem

3.1 Problem Formulation

Unknowns: For K sources, $\boldsymbol{\theta} = (x_1, y_1, q_1, \ldots, x_K, y_K, q_K) \in \mathbb{R}^{3K}$.

Objective:

$$\boxed{J(\boldsymbol{\theta}) = \frac{1}{2}\left\|\boldsymbol{u}^{\text{forward}}(\boldsymbol{\theta}) - \tilde{\boldsymbol{u}}\right\|_2^2} \tag{38}$$

where $\boldsymbol{u}^{\text{forward}}(\boldsymbol{\theta})$ computes boundary values from sources at (x_k, y_k) with intensities q_k.

Constraints:

$$(x_k, y_k) \in \Omega \quad \text{(sources inside domain)} \tag{39}$$

$$\sum_{k=1}^{K} q_k = 0 \quad \text{(compatibility)} \tag{40}$$

3.2 Why It's Nonlinear

The forward model is:

$$u_i^{\text{forward}} = \sum_{k=1}^{K} q_k \, G_N(\boldsymbol{x}_i, (x_k, y_k)) \tag{41}$$

This is **linear** in q_k but **nonlinear** in (x_k, y_k) because $G_N(\boldsymbol{x}, \boldsymbol{\xi})$ depends nonlinearly on $\boldsymbol{\xi}$.

3.3 Gradient Computation

For gradient-based optimization, we need:

$$\nabla_{\boldsymbol{\theta}} J = \boldsymbol{J}^T (\boldsymbol{u}^{\text{forward}} - \tilde{\boldsymbol{u}}) \tag{42}$$

where $\boldsymbol{J} = \frac{\partial \boldsymbol{u}^{\text{forward}}}{\partial \boldsymbol{\theta}}$ is the Jacobian.

3.3.1 Jacobian Components

For the k-th source:

$$\frac{\partial u_i}{\partial x_k} = q_k \frac{\partial G_N}{\partial \xi_1}(\boldsymbol{x}_i, (x_k, y_k)) \tag{43}$$

$$\frac{\partial u_i}{\partial y_k} = q_k \frac{\partial G_N}{\partial \xi_2}(\boldsymbol{x}_i, (x_k, y_k)) \tag{44}$$

$$\frac{\partial u_i}{\partial q_k} = G_N(\boldsymbol{x}_i, (x_k, y_k)) \tag{45}$$

3.3.2 Green's Function Gradient (Unit Disk)

From Part 1, using complex notation:

$$G_N(z, \zeta) = -\frac{1}{2\pi} \ln |z - \zeta| - \frac{1}{2\pi} \ln \left|1 - z\bar{\zeta}\right| + C \tag{46}$$

The gradient with respect to source position $\zeta = (\xi_1, \xi_2)$:

$$\frac{\partial G_N}{\partial \xi_1} = -\frac{1}{2\pi} \left(\frac{-(x - \xi_1)}{|\boldsymbol{x} - \boldsymbol{\xi}|^2} + \frac{x}{\left|1 - z\bar{\zeta}\right|^2} \right) \tag{47}$$

$$\frac{\partial G_N}{\partial \xi_2} = -\frac{1}{2\pi} \left(\frac{-(y - \xi_2)}{|\boldsymbol{x} - \boldsymbol{\xi}|^2} + \frac{y}{\left|1 - z\bar{\zeta}\right|^2} \right) \tag{48}$$

3.4 Optimization Methods

3.4.1 L-BFGS-B

Limited-memory BFGS with box constraints.

Advantages:

- Fast convergence for smooth problems
- Handles box constraints (bounds on positions)
- Moderate memory usage

Disadvantages:

- Finds local minima only
- Sensitive to initialization

3.4.2 Differential Evolution

Global optimization via evolutionary algorithm.

Advantages:

- Global search (escapes local minima)
- No gradient required
- Robust to noise

Disadvantages:

- Slow (many function evaluations)
- May not converge to exact optimum

3.4.3 Multi-Start Strategy

Run L-BFGS-B from multiple random initializations:

1. Sample N_{start} initial configurations
2. Run L-BFGS-B from each
3. Return best solution

3.5 Handling the Compatibility Constraint

Method 1: Elimination. Set $q_K = -\sum_{k=1}^{K-1} q_k$, reducing unknowns to $3K - 1$.

Method 2: Penalty. Add penalty term $\mu(\sum_k q_k)^2$ to objective.

Method 3: Projection. After each optimization step, project: $q_k \leftarrow q_k - \bar{q}$ where $\bar{q} = \frac{1}{K}\sum_k q_k$.

4 Quality Metrics

4.1 For Linear Solvers

Traditional metrics (RMSE of peaks) are **misleading** for distributed solutions.

4.1.1 Localization Score

Definition 4.1 (Localization Score)**.**

$$S_{loc} = \frac{\sum_{j=1}^{M} |q_j| \cdot w_j}{\sum_{j=1}^{M} |q_j|} \tag{49}$$

where $w_j = \max_k \exp\left(-\frac{\|\boldsymbol{\xi}_j - \boldsymbol{z}_k^{true}\|^2}{2\sigma^2}\right)$ *is the Gaussian weight to nearest true source.*

Interpretation:

- $S_{\text{loc}} = 1$: All intensity concentrated exactly at true sources
- $S_{\text{loc}} \approx 0$: Intensity far from true sources

4.1.2 Sparsity Ratio

Definition 4.2 (Sparsity Ratio)**.**

$$S_{spar} = \min\left(\frac{K_{target}}{N_{90\%}}, 1\right) \tag{50}$$

where $N_{90\%}$ is the number of points containing 90% of total intensity.

Interpretation:

- $S_{\text{spar}} = 1$: Intensity in exactly K_{target} points (sparse)
- $S_{\text{spar}} \approx 0$: Intensity spread across many points (diffuse)

4.2 For Nonlinear Solvers

4.2.1 Position RMSE

$$\text{RMSE}_{\text{pos}} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{z}_k^{\text{rec}} - \boldsymbol{z}_{\pi(k)}^{\text{true}}\right\|^2} \tag{51}$$

where π is the optimal matching between recovered and true sources (Hungarian algorithm).

4.2.2 Intensity RMSE

$$\text{RMSE}_{\text{int}} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(q_k^{\text{rec}} - q_{\pi(k)}^{\text{true}})^2} \tag{52}$$

4.2.3 Boundary Residual

$$\text{Res} = \frac{\left\|\boldsymbol{u}^{\text{forward}}(\boldsymbol{\theta}^{\text{rec}}) - \tilde{\boldsymbol{u}}\right\|_2}{\|\tilde{\boldsymbol{u}}\|_2} \tag{53}$$

5 Summary of Algorithms

5.1 Linear Inverse (Distributed)

1. **Build Green's matrix $\boldsymbol{G}$** (forward solves for each grid point)
2. **Select** α via L-curve
3. **Solve regularized problem**:
   - L1: CVXPY or coordinate descent
   - L2: Closed-form (Equations **??**–**??**)
   - TV: ADMM
4. **Evaluate** localization score, sparsity ratio

5.2 Nonlinear Inverse (Point Sources)

1. **Specify** number of sources K
2. **Initialize** positions (random or from linear solution peaks)
3. **Optimize**:
   - L-BFGS-B with multi-start, or
   - Differential evolution
4. **Enforce** compatibility $\sum q_k = 0$
5. **Evaluate** position RMSE, intensity RMSE

End of Mathematical Formulation

A Notation Summary

Symbol	Meaning
Ω	Domain (open, bounded, simply connected)
$\partial\Omega$	Boundary of domain
$\mathbb{D}$	Unit disk $\{z : \|z\| < 1\}$
Δ	Laplacian operator
∇	Gradient operator
G_N	Neumann Green's function
G_0	Free-space Green's function
$\boldsymbol{z}_k$	True source position
q_k	Source intensity
$\boldsymbol{\xi}_j$	Grid point (candidate source location)
$\boldsymbol{G}$	Green's matrix ($N_b \times M$)
$\boldsymbol{q}$	Source intensity vector
$\tilde{\boldsymbol{u}}$	Measured boundary values
α	Regularization parameter
$\boldsymbol{D}$	Discrete gradient operator