

Tomography and Location

- Forward and Inverse travel-time problems
 - Seismic tomography
 - Least Squares inverse
 - Generalised Linear Inverse
 - Iterative inverse
 - ◆ Back-projection method
 - Resolution
 - Statistical testing of results
 - Location of seismic sources
 - Data norms
- Reading:
Shearer, 5.6-5.7

Seismic (velocity) tomography

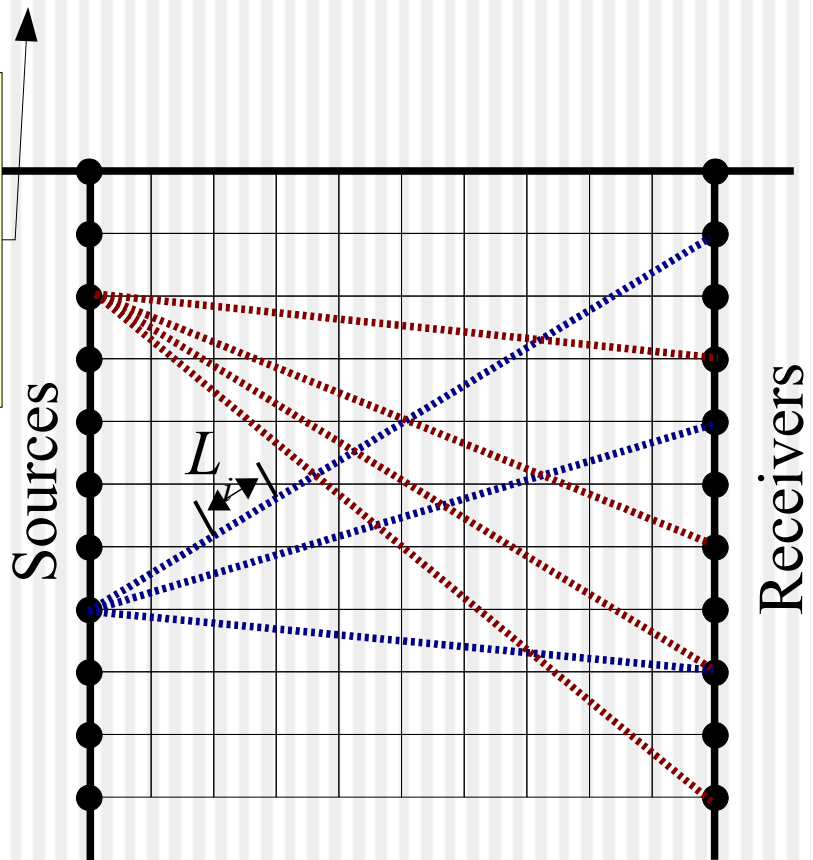
- Tomography
 - ◆ The name derived from the Greek for “section drawing” - the idea is that the section appears *almost* automatically...
 - ◆ Using multitude of source-receive pairs with rays crossing the area of interest.
 - ◆ Looking for an unknown velocity structure.
 - ◆ Depending on the type of recording used, it could be:
 - *Transmission tomography* (nearly straight rays between boreholes);
 - *Reflection tomography* (reflected rays; in this case, positions of the reflectors could be also found);
 - *Diffraction tomography* (using least-time travel paths according to Fermat rather than Snell's law; this is actually more a waveform inversion technique).

Cross-well tomography

- Consider the case of transmission “cross-well” tomography
 - ◆ This is the simplest case – rays may be considered nearly straight, the data are abundant, and the coverage is *relatively uniform*.

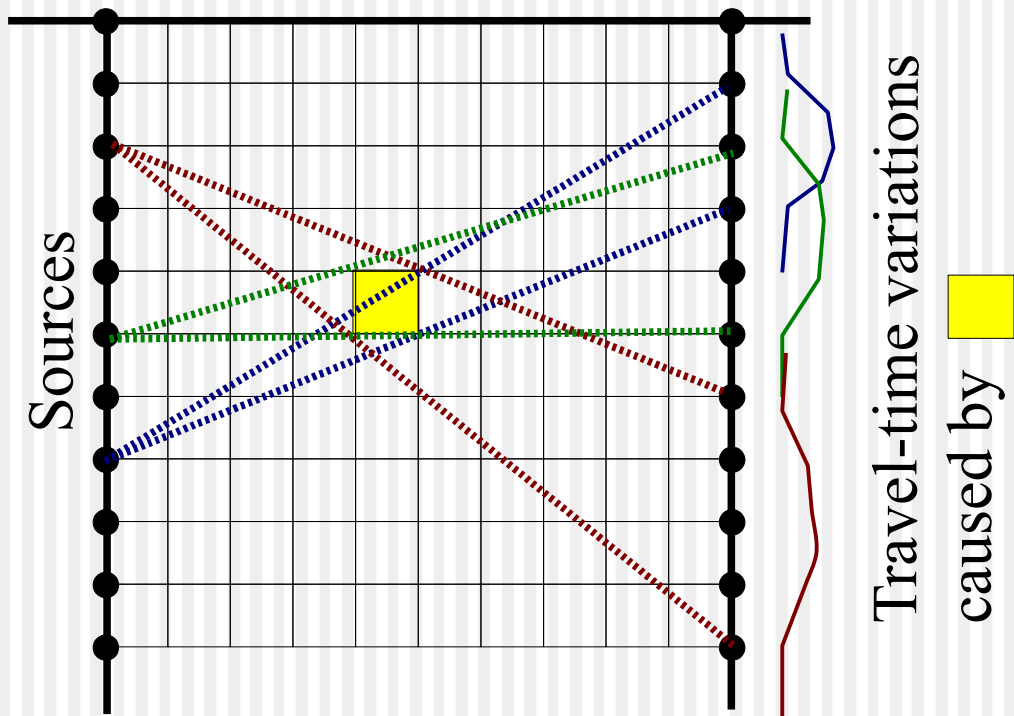
These are the three principal concerns in tomography:

- 1) linearity of the problem;
- 2) density of data coverage;
- 3) good azimuthal coverage.



Principle

- Velocity perturbations are considered as small
 - ◆ Therefore, rays are approximated as straight
- Each velocity cell leads to characteristic travel-time variations at the receivers (“impulse response”)
 - ◆ These are inverted for velocity value at



Travel-time inversion as a *linear inverse problem*

- First, we parameterize the velocity model
 - ♦ Typically, the parameterization is a grid of constant-velocity blocks (sometimes splines are used instead of the blocks).
 - ♦ This parameterization gives us a *model vector*, **m**.

$$\mathbf{m} = \begin{pmatrix} s_1 = 1/V_1 \\ s_2 = 1/V_2 \\ \dots \\ s_N = 1/V_N \end{pmatrix}.$$

- Second, we measure all travel times and arrange them into a data vector:

$$\mathbf{d}^{observed} = \begin{pmatrix} t_1 \\ t_2 \\ \dots \\ t_M \end{pmatrix}.$$

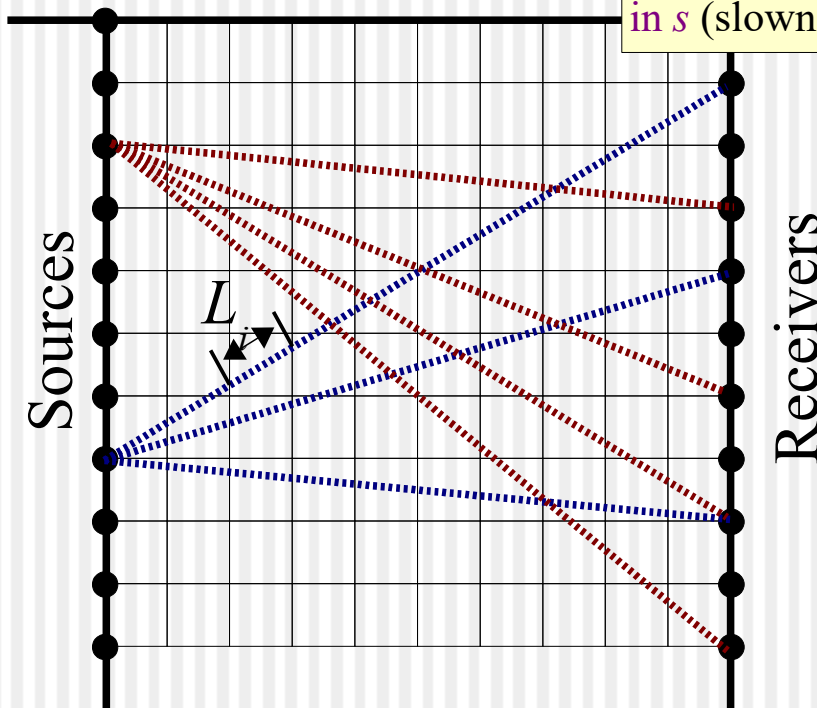
Forward model

- Third, we formulate the *forward model* to predict \mathbf{d} from \mathbf{m} . To achieve this, we need to *trace rays* through the model and measure the length of every ray's segment in each model block, L_{ij} .

- The travel time for i -th ray is then:

$$t_i = \sum_j L_{ij} \frac{1}{V_j} = \sum_j L_{ij} s_j.$$

Note that the expression is non-linear in V but **linear** in s (slowness).



Generalized Linear Inverse

- The model for travel times: $t_i = \sum_j L_{ij} s_j$
can be written in matrix form:

$$\mathbf{d} = \mathbf{L} \mathbf{m}$$

- Now, we want to substitute $\mathbf{d} = \mathbf{d}^{\text{observed}}$ and solve for unknown \mathbf{m} . This is called the *inverse problem*.
- Typically, matrix L is not invertible (it is not square), and so it is inverted in some *generalized* (averaged) sense.
- Any solution in the linear form

$$\mathbf{m} = \mathbf{L}_g^{-1} \mathbf{d}^{\text{observed}}$$

is called the *generalized linear inverse*.

- The problem is thus in finding a suitable form for \mathbf{L}_g^{-1} .

Projection into model space

- Often, tomography problems are typically overdetermined (contain many more ray paths than grid model blocks).
- In such cases, the following approach to constructing \mathbf{L}_g^{-1} works well:

- ◆ multiply by transposed \mathbf{L}^T :

$$\mathbf{L}^T d^{observed} = \mathbf{L}^T \mathbf{L} m ,$$

- ◆ hence:

$$m = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T d^{observed} .$$

This operation
“back-projects”
the redundant data
into “model space”

This is the
“least-squares” solution
It is used
in the
well-known GLI3D
program
for refraction
statics

Least Squares Inverse

- Note that the solution is a linear combination of data values:

$$\mathbf{m} = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T \mathbf{d}^{\text{observed}} = \mathbf{L}_g^{-1} \mathbf{d}^{\text{observed}} .$$

$$\mathbf{L}_g^{-1} = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T .$$

The “generalized inverse” matrix

- The reason for its name of “Least Squares” is in minimizing the mean square of data misfits:

$$\text{Misfit}(\mathbf{m}) = (\mathbf{d}^{\text{observed}} - \mathbf{L}\mathbf{m})^T (\mathbf{d}^{\text{observed}} - \mathbf{L}\mathbf{m}) .$$

- ♦ Exercise: show this!

Damped Least Squares

- Sometimes the matrix $\mathbf{L}^T\mathbf{L}$ is singular and its inverse is unstable.
 - ♦ This happens, e.g., when some cells are not crossed by any rays, or there are groups of cells traversed by the same rays only.
- In such cases, the inversion can be *regularized* by adding a small positive diagonal term to $\mathbf{L}^T\mathbf{L}$:

$$\mathbf{m} = (\mathbf{L}^T \mathbf{L} + \varepsilon \mathbf{I})^{-1} \mathbf{L}^T \mathbf{d}^{\text{observed}} .$$

- ♦ This is called the *Damped Least Squares* solution.
- ♦ ε is chosen such that stability is achieved and the non-zero contributions in $\mathbf{L}^T\mathbf{L}$ are affected only slightly.

Weighted Least Squares

- Often, different types of data are included in **d**
 - For example, different travel times, t_i , may be measured with different uncertainties δt_i
- In such cases, it is useful to apply weights to the equations:

$$\mathbf{W} \mathbf{d} = \mathbf{W} \mathbf{L} m$$

where **W** is a diagonal **weight matrix**:

$$\mathbf{W} = \text{diag} \left(\frac{1}{\delta t_1}, \frac{1}{\delta t_2}, \frac{1}{\delta t_3}, \dots \right)$$

Weighted Least Squares (cont.)

- This corresponds to a modified least-squares misfit function:

$$\text{Misfit}(m) = (d^{\text{observed}} - Lm)^T W^T W (d^{\text{observed}} - Lm)$$

and solution:

$$m = L_g^{-1} d^{\text{observed}}$$

$$L_g^{-1} = (L^T W^T W L + \varepsilon I)^{-1} L^T W$$

Smoothness constraints

- When using finely-sampled models...
 - some cells may be poorly constrained;
 - solutions can become 'rough' (highly variable, noisy – see below)
- To remove roughness, additional 'smoothness constraint' equations can be added
 - These equations will be additional rows in \mathbf{L} , for example:
 - $w m_i = w \text{Average}(\text{Adjacent } m_j)$
 - Zero Laplacian: $w \nabla^2 m = 0$
- These equations require **small weights** w

Simple Iterative Inverse

- Sometimes matrix $\mathbf{L}^T \mathbf{L}$ is also too large to invert, or even to store
- It can be approximated by its diagonal:

$$\mathbf{m} = \left[\text{diag}(\mathbf{L}^T \mathbf{L}) + \varepsilon \mathbf{I} \right]^{-1} \mathbf{L}^T \mathbf{d}^{\text{observed}}.$$

- ◆ The diagonal only contains one value per model cell (sum of squared L 's for all rays crossing it)
- ◆ Contributions to \mathbf{m} can be evaluated during a pass through all data and **without storing** matrices \mathbf{L} or $\mathbf{L}^T \mathbf{L}$.
- Variants of this method are known as:
 - ◆ **Back-projection** method;
 - ◆ **SIRT** (Simultaneous Iterative Reconstruction technique)
 - ◆ **ART** (Algebraic Reconstruction Technique)

Simple Iterative Inverse *(how it works)*

- Iteration:

$$\delta_1 d = d^{observed} - d_0,$$

$$\delta_1 m = L_g^{-1} \delta_1 d.$$

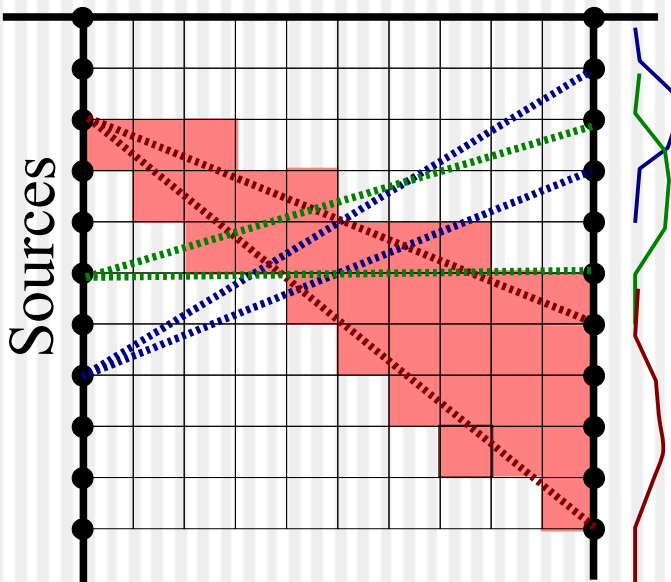
Travel times in “background model”

$$\delta_2 d = \delta_1 d - L \delta_1 m,$$

$$\delta_2 m = L_g^{-1} \delta_2 d,$$

Approximate inverse of any kind

...



For each ray, the observed travel-time perturbation is thus “back-projected” into the slowness model

Resolution matrix

- Assessment of the *quality of inversion method* is often done by using the *Resolution Matrix*
 - ◆ Regardless of the selected form of the inverse, we can:
 - 1) Perturb 1 parameter (grid node) of the model;
 - 2) Perform forward modeling (generate synthetic data);
 - 3) Perform the inverse.
 - ◆ When repeated for each parameter, this process results in a resolution matrix:

$$\mathbf{R} = \mathbf{L}_g^{-1} \mathbf{L}$$

- Note that \mathbf{R} *does not* depend on the data values but depends on sampling
 - ◆ Crossing rays are VERY important in tomography.

Checkerboard resolution test

- Test of the resolution in the model when computation of the *Resolution Matrix* is impossible or impractical
- Method:

- ◆ Create an artificial model perturbation in the form of alternating positive and negative anomalies (“checkerboard”)

- ◆ Predict the data in this model:

$$d' = L m_{checker}$$

- ◆ Invert the resulting synthetic data:

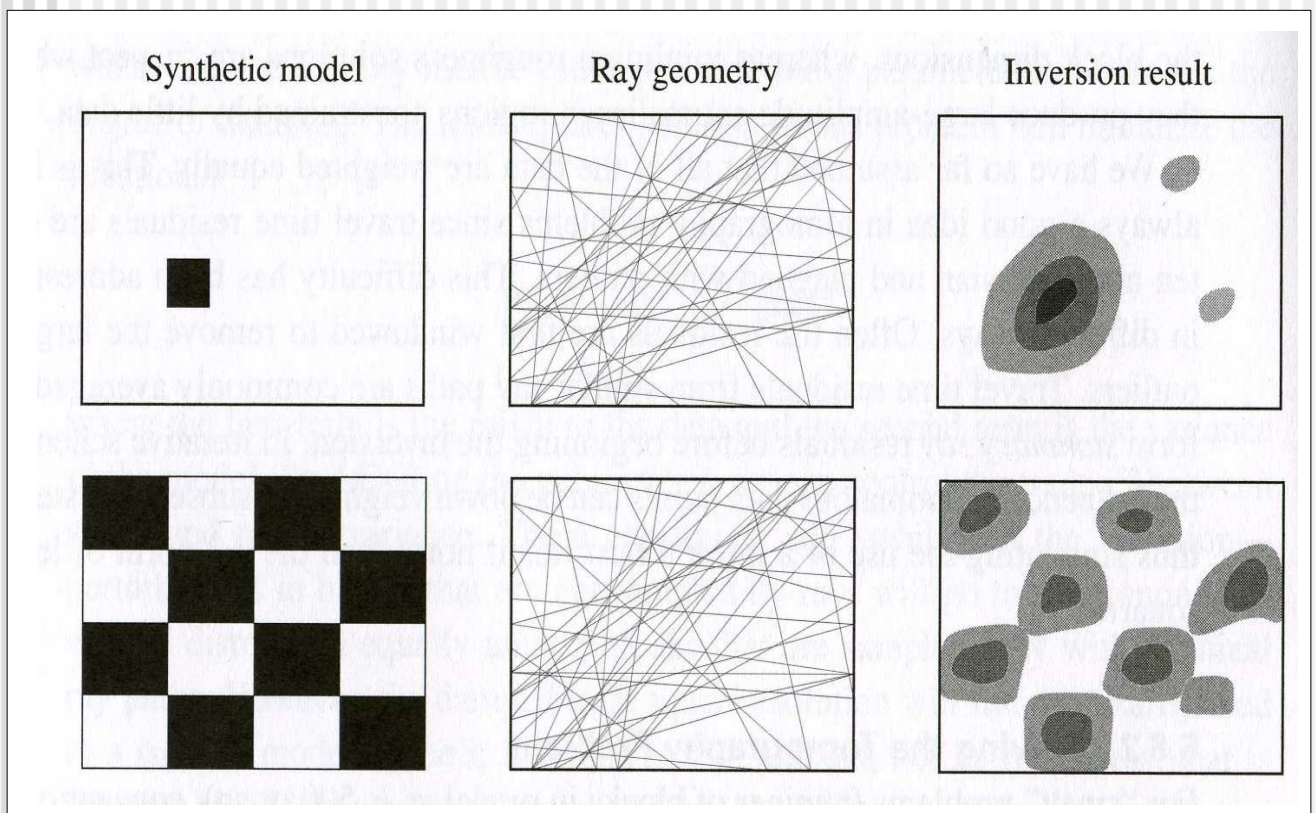
$$m' = L_g^{-1} d' = L_g^{-1} L m_{checker}$$

- ◆ Compare the result to the input model

- The degree of reproduction of the anomalies indicate the quality of inversion

Checkerboard resolution test (cont.)

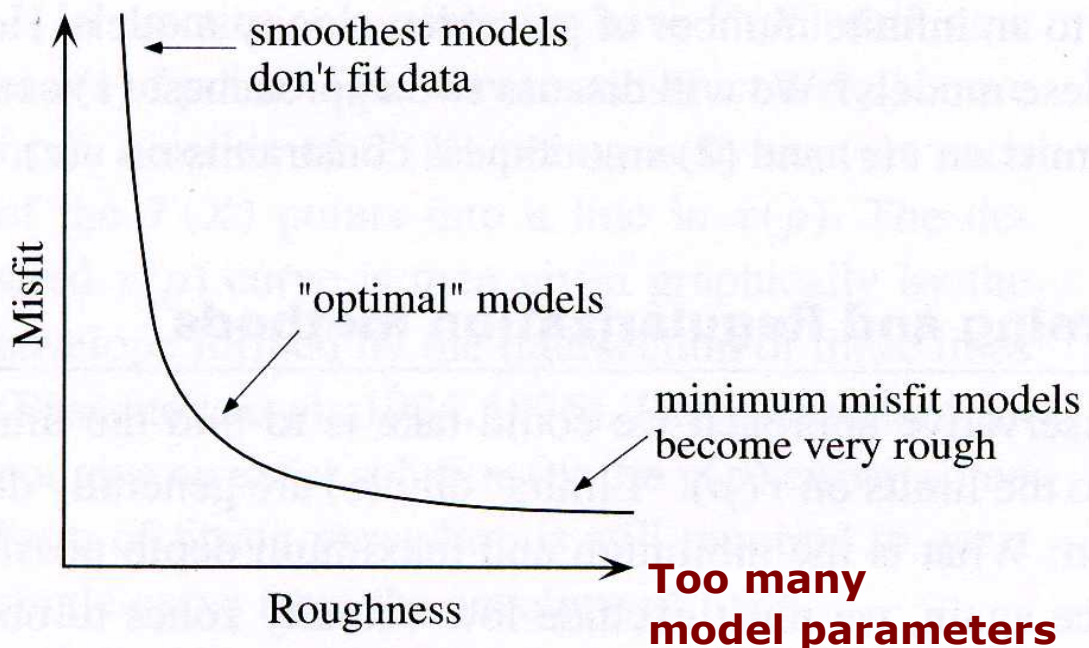
- Schematic example from travel-time tomography:



Trade-off between data fit and model simplicity

- Too simple models often cannot explain the data
- However, excessively detailed models can “over-fit” the data and result in overly complex model
 - ◆ This complexity may be spurious and caused by the noise
- We need to look for “optimally” complex models

Too few model parameters



Test for statistical significance

- How can we verify that the model fits the data within reasonable error?
 - ◆ Complex models (with large numbers of unknowns) would often fit the data well;
 - ◆ Because the data contains *noise*, we should not **over-fit** the data!
- The χ^2 test is commonly used to determine whether the remaining data misfit is likely to be random:

$$\chi^2 = \frac{\sum_{i=1}^N (t_i - t_i^{\text{observed}})^2}{\sigma^2}$$

- ◆ Here, σ is the estimated data uncertainty
- ◆ It needs to be somehow measured from the data (see eq. 5.31 in Shearer)

χ^2 test (cont.)

- The p.d.f of χ^2 is controlled by $N_{df} = N_{data} - N_{model}$ ("number of data degrees of freedom").
- For a given N_{df} , tabulated percentage points of p.d.f. (χ^2) can be used to determine whether the residual data misfit is likely to be random:

N_{df}	At 95%	At 50%	At 5%
5	1.15	4.35	11.07
10	3.94	9.34	18.31
20	10.85	19.34	31.41
50	34.76	49.33	67.5
100	77.03	99.33	124.34

- The 95-% level is commonly used.

Source Location Problem

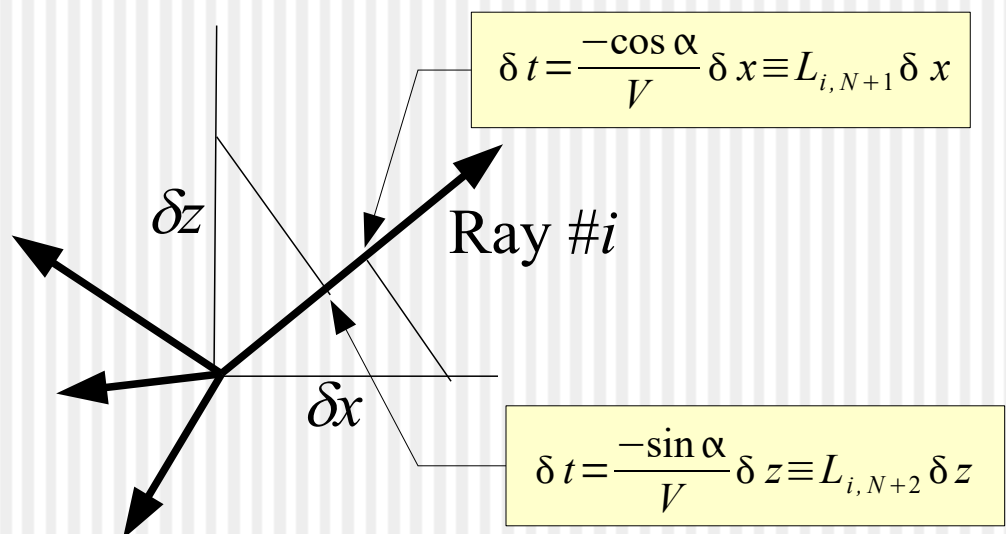
- When using a natural (impulsive) source, its location can also be determined by a similar approach.
 - ◆ This method is used for locating earthquakes worldwide
 - ◆ For monitoring creep of mine walls (potash exploration)
 - ◆ Monitoring reservoirs during injection (Weyburn)
- To solve this problem, we:
 - ◆ Start from some reasonable approximation for source coordinates and solve the velocity tomography problem.
 - ◆ Add the coordinates and time of the source to model vector:

$$\mathbf{m} = \begin{pmatrix} s_1 = 1/V_1 \\ s_2 = 1/V_2 \\ \dots \\ s_N = 1/V_N \\ x_{source} \\ z_{source} \\ t_{source} \end{pmatrix}.$$

(In two dimensions)

Source Location (cont.)

- Include into the matrix **L** time delays associated with shifting the source by δx or δz :



- ◆ Now, when solved, the Generalized Inverse will yield the corrections to the location (δx , δz).
- This process is often iterated: with the new source location, velocities are recomputed, and sources relocated again, etc.

Measures of data misfit (“data norms”)

- The Least-Squares (“L2”) norm can be highly sensitive to data outliers:

$$\varepsilon_{L2} = \sum_{i=1}^N (t_i - t_i^{observed})^2$$

- However, it is the easiest to use (only for this norm, L^{-1}_g exists).

- Other useful norms:

- L_n norms: $\varepsilon_{L_n} = \sum_{i=1}^N |t_i - t_i^{observed}|^n$

- L_∞ norm: $\varepsilon_{L_\infty} = \max_i |t_i - t_i^{observed}|$

- The “ L_1 ” norm is less sensitive to outliers (*i.e.*, anomalous errors), and therefore also often used:

$$\varepsilon_{L_1} = \sum_{i=1}^N |t_i - t_i^{observed}|$$

L_1 -norm inversion

- Solutions minimizing L_1 and similar norms are derived from L_2 by *iterative reweighting*:

1) Use the least-squares inverse to minimize

$$\varepsilon_{L2} = \sum_{i=1}^N (t_i - t_i^{observed})^2$$

2) Apply weights based on current data errors:

$$W_i = \frac{1}{\sqrt{|t_i - t_i^{observed}|}}$$

- The misfit then approximates ε_{L1} :

$$\varepsilon_{L2} = \sum_{i=1}^N W_i^2 (t_i - t_i^{observed})^2 \approx \sum_{i=1}^N |t_i - t_i^{observed}|$$

3) Iterate