RESOLUTION ANALYSIS BY RANDOM PROBING

Andreas Fichtner
ETH Zurich

Tristan van Leeuwen
Utrecht University
"Solving an inverse problem means to describe the infinite-dimensional space of data-fitting models."

George Backus & Freeman Gilbert, 1968
1. Why resolution analysis is becoming more and more difficult

A simple example
The Resolution Matrix

\[ R \text{ m}_{\text{true}} = \text{m}_{\text{est}} \]

Resolution matrix How the true Earth is smeared into an image. Dimension \( N \times N \).

True Earth model Dimension \( N \).

Estimated Earth model Dimension \( N \).
The Resolution Matrix

\[ R \cdot m_{\text{true}} = m_{\text{est}} \]

Resolution matrix How the true Earth is smeared into an image.
Dimension \( N \times N \).

True Earth model
Dimension \( N \).

Estimated Earth model
Dimension \( N \).

- In the days of Backus & Gilbert: \( N = O(10^2) \) \( \Rightarrow \) \( R \) is \( O(10^2) \) times larger than \( m \).
The Resolution Matrix

Resolution matrix: How the true Earth is smeared into an image. Dimension $N \times N$.

True Earth model: Dimension $N$.

Estimated Earth model: Dimension $N$.

• In the days of Backus & Gilbert: $N = O(10^2) \rightarrow R$ is $O(10^2)$ times larger than $m$.
• Today: $N = O(10^7) \rightarrow R$ is $O(10^7)$ times larger than $m$. 
The Resolution Matrix

\[ \mathbf{R} \mathbf{m}_{\text{true}} = \mathbf{m}_{\text{est}} \]

**Resolution matrix** How the true Earth is smeared into an image. Dimension \( N \times N \).

**True Earth model**
- Dimension \( N \).

**Estimated Earth model**
- Dimension \( N \).

- In the days of Backus & Gilbert: \( N = O(10^2) \) → \( \mathbf{R} \) is \( O(10^2) \) times larger than \( \mathbf{m} \).
- Today:
  - \( N = O(10^7) \) → \( \mathbf{R} \) is \( O(10^7) \) times larger than \( \mathbf{m} \).

- As data volumes and computing power grow:
  - We can construct bigger and bigger models \( \mathbf{m}_{\text{est}} \).
The Resolution Matrix

Resolution matrix How the true Earth is smeared into an image. Dimension $N \times N$.

True Earth model
Dimension $N$.

Estimated Earth model
Dimension $N$.

The problem:

• In the days of Backus & Gilbert: $N = O(10^2) \rightarrow R$ is $O(10^2)$ times larger than $m$.
• Today: $N = O(10^7) \rightarrow R$ is $O(10^7)$ times larger than $m$.

• As data volumes and computing power grow:
  - We can construct bigger and bigger models $m_{est}$.
  - We loose our ability to quantify the quality of $m_{est}$. 
We need **scalable** methods to infer useful **aspects** of resolution.
We need scalable methods to infer useful aspects of resolution.

Objectives of this Webinar:

• Describe 2 methods to quantify resolution when R is too expensive to compute and too big to store.

• One method for linear problems, and one for (mildly) nonlinear problems.

• Both based on random probing techniques.
2. Estimating the number of resolved parameters

\[ \text{tr } R \]
$m_i$  

- random test model vector
- Expectation: $E[m_i]=0$
- Covariance: $\text{cov}(m_i,m_j)=\delta_{ij}$ [uncorrelated components]
Estimating the Number of Resolved Parameters

- $m_i$
  - Random test model vector
  - Expectation: $E[m_i]=0$
  - Covariance: $\text{cov}(m_i,m_j)=\delta_{ij}$ [uncorrelated components]

- $R_{ij}$
  - A resolution matrix

Too large to computer.
Too large to store.
Too large to comprehend fully.
ESTIMATING THE NUMBER OF RESOLVED PARAMETERS

\( m_i \)
- random test model vector
- Expectation: \( E[m_i] = 0 \)
- Covariance: \( \text{cov}(m_i, m_j) = \delta_{ij} \) [uncorrelated components]

\( R_{ij} \)
- A resolution matrix

\( E[m_i R_{ij} m_j] \)
Estimating the Number of Resolved Parameters

$m_i$
- random test model vector
- Expectation: $E[m_i]=0$
- Covariance: $\text{cov}(m_i, m_j) = \delta_{ij}$ [uncorrelated components]

$R_{ij}$
- A resolution matrix

$E[m_i R_{ij} m_j] = R_{ij} E[m_i m_j]$
Estimating the Number of Resolved Parameters

$m_i$  
- random test model vector  
- Expectation: $E[m_i]=0$  
- Covariance: $\text{cov}(m_i,m_j)=\delta_{ij}$ [uncorrelated components]

$R_{ij}$  
- A resolution matrix

$$E[m_i R_{ij} m_j] = R_{ij} E[m_i m_j]$$
$$= R_{ij} \left( E[m_i]E[m_j] + \text{cov}(m_i,m_j) \right)$$
m_i

- random test model vector
- Expectation: E[m_i]=0
- Covariance: cov(m_i,m_j)=\delta_{ij} [uncorrelated components]

R_{ij}

- A resolution matrix

E[m_i R_{ij} m_j] = R_{ij} E[m_i m_j]

= R_{ij} (E[m_i]E[m_j] + cov(m_i, m_j))

= R_{ij} \delta_{ij} = R_{ii} = \text{tr } R
**ESTIMATING THE NUMBER OF RESOLVED PARAMETERS**

$m_i$

- random test model vector
- Expectation: $E[m_i]=0$
- Covariance: $\text{cov}(m_i, m_j)=\delta_{ij}$ [uncorrelated components]

$R_{ij}$

- A resolution matrix

$$E[m_i R_{ij} m_j] = R_{ij} E[m_i m_j]$$
$$= R_{ij} \left( E[m_i] E[m_j] + \text{cov}(m_i, m_j) \right)$$
$$= R_{ij} \delta_{ij} = R_{ii} = \text{tr } R = \text{number of resolved model parameters}$$

Hutchinson’s method [Hutchinson, 1990]
m_i

- random test model vector
- Expectation: E[m_i]=0
- Covariance: cov(m_i,m_j)=δ_{ij} [uncorrelated components]

R_{ij}

- A resolution matrix

E[m_iR_{ij}m_j] = R_{ij} E[m_i m_j]

= R_{ij} (E[m_i] E[m_j] + cov(m_i,m_j))

= R_{ij} δ_{ij} = R_{ii} = tr R = number of resolved model parameters

**Very simple recipe:**

- Choose a random test model \( m \).
- Try to recover this model in a synthetic inversion [i.e. compute \( m_{est} = Rm \)].
- Multiply the result with \( m \) itself: \( m^T m_{est} = m^T R m \).
- Average over some random realisations.
- The resolution matrix itself never has to be computed!
OTHER RANDOM PROBING TECHNIQUES


3. Random probing for resolution analysis in tomography

Estimating position- and direction-dependent resolution lengths.
• Misfit $\chi$ in the vicinity of the optimal model $m$:

\[
\chi(m + \delta m) = \chi(m) + \frac{1}{2} \delta m^T H(m) \delta m
\]

Hessian operator
Inverse posterior covariance [assuming Gaussian errors]
Column: point-spread function

$H$ is too expensive to compute and store.

• But we can infer properties of $H$ from its application to random test models.
**RANDOM PROBING PRINCIPLE**

- Assume $H$ is Gaussian [for simplicity and illustration]:

\[
H(x; y) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{1}{2\sigma^2} (x - y)^2}
\]

\[
h(y) = \int H(x; y)v(x) \, dx
\]
• Assume $\mathbf{H}$ is Gaussian [for simplicity and illustration]:

$$H(x; y) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{1}{2\sigma^2} (x-y)^2}$$

$$h(y) = \int H(x; y)v(x) \, dx$$

• Length scales of $\mathbf{h}$ contain information on length scales of $\mathbf{H}$. 
• **Auto-correlation of the output h** [averaged over many realisations]:

![Graph showing average auto-correlations of h](image)
• **Auto-correlation** of the output $h$ [averaged over many realisations]:

• **Asymptotically:** width of auto-correlation $= \sqrt{2} \cdot$ width of $H$
1. Resolution and correlations
   • The Hessian acts as a smoother of random functions.
   • The smoothed functions carry information on resolution.
   • Can be extracted with correlations.

2. Convergence
   • Correlations themselves may require large sample sizes to converge.
   • The width of the correlation converges extremely quickly.
     ➢ Useful resolution proxies may already be obtained with very few samples.
Synthetic full-waveform inversion in 2D
Synthetic inversion setup

(a) target model
(b) initial model
(c) final model
- Synthetic inversion setup
• Application of random test models to the Hessian via second-order adjoints
• Local auto-correlation of the output in different directions.
- Application of random test models to the Hessian via second-order adjoints.
- Local auto-correlation of the output in different directions.
- Estimated width of the point-spread functions in $x_1$-direction [resolution length].

- Around 5-10 samples to converge.
- Resolution is strongly heterogeneous.

**SYNTHETIC EXAMPLE IN 2D**

![Resolution length in $x_1$-direction](image)
• Estimated width of the point-spread functions in $x_2$-direction [resolution length].
Real-data application
**Technical summary:**

**Data**

- 52 earthquakes, >1000 stations
- body waves, surface waves, ...
- periods: \(10 - 150\) s

**Forward modelling**

- spectral elements
- **3D visco-elastic, anisotropic**

**Inversion**

- initial model from previous European FWI
- adjoint-based CG
- invert for \(v_{sh}, v_{sv}, v_p, \rho\) and source location/mechanism
isotropic S velocity

S VELOCITY MODEL

Fichtner & Villasenor, EPSL 2015.
S VELOCITY MODEL

isotropic S velocity variations

50 km

100 km

600 km

300 km

Fichtner & Villasenor, EPSL 2015.
POSITION- AND DIRECTION-DEPENDENT RESOLUTION LENGTHS

N-S direction

E-W direction

radial direction

50 km

300 km

$\lambda_{\text{lat}}$ [km]

$\lambda_{\text{lon}}$ [km]

$\lambda_{r}$ [km]
Conclusions
Limitations:

1. Local analysis
Limitations:

1. Local analysis

Benefits:

1. Quantify spatial resolution and inter-parameter trade-offs.
Limitations:

1. Local analysis

Benefits:

1. Quantify spatial resolution and inter-parameter trade-offs.

2. Low computational costs
   - around 5 Hessian-model applications
   - equivalent to around 5 CG iterations
   - much less than a synthetic inversion
Limitations:

1. Local analysis

Benefits:

1. Quantify spatial resolution and inter-parameter trade-offs.

2. Low computational costs
   • around 5 Hessian-model applications
   • equivalent to around 5 CG iterations
   • much less than a synthetic inversion

3. Low algorithmic complexity
   • easy to implement without modifications of existing codes
Limitations:

1. Local analysis

Benefits:

1. Quantify spatial resolution and inter-parameter trade-offs.

2. Low computational costs
   - around 5 Hessian-model applications
   - equivalent to around 5 CG iterations
   - much less than a synthetic inversion

3. Low algorithmic complexity
   - easy to implement without modifications of existing codes

4. Scalability
   - 5 random models sufficient in 1, 2 and 3 dimensions [empirical]
Limitations:

1. Local analysis

Benefits:

1. Quantify spatial resolution and inter-parameter trade-offs.

2. Low computational costs
   - around 5 Hessian-model applications
   - equivalent to around 5 CG iterations
   - much less than a synthetic inversion

3. Low algorithmic complexity
   - easy to implement without modifications of existing codes

4. Scalability
   - 5 random models sufficient in 1, 2 and 3 dimensions [empirical]

5. Applicability to any tomographic technique
Thanks for your attention!