TPG4190 Seismic data acquisition and processing
Lecture 20: Deconvolution

B. Arntsen

NTNU
Department of Geoscience and petroleum
borge.arntsen@ntnu.no

Trondheim fall 2020

Geophysical Analysis - lecture 10

B. Arntsen

NTNU
Department of Geoscience and petroleum
borge.arntsen@ntnu.no

Trondheim fall 2020

Contents

Convolutional model

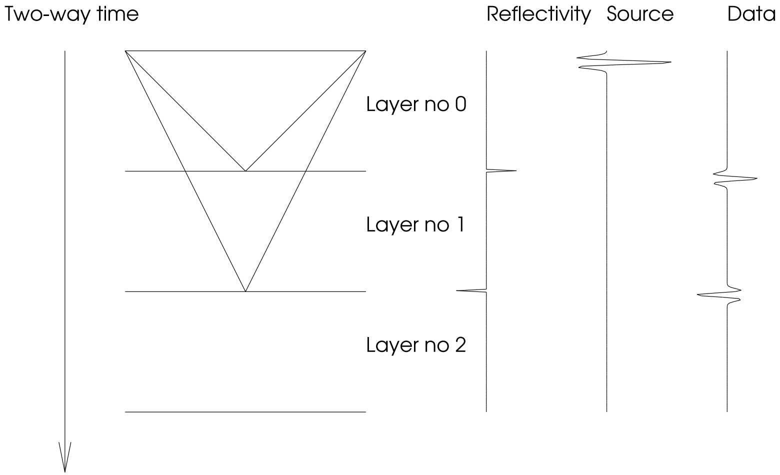


Figure: Layered earth model

Convolutional model

- vertical propagation

Reflected signal

$$a(t) = r_0 \frac{s(t - \phi_0)}{2d_0}. \tag{1}$$

- $\phi_0 = 2d_0/c_0$: two-way traveltime
- r_0 : reflection coefficient
- d_0 : layer thickness

Convolutional model

Reflection coefficient defined by

$$r_0 = \frac{c_1\rho_1 - c_0\rho_0}{c_1\rho_1 + c_0\rho_0}. \tag{2}$$

- c_0, c_1: Wave velocity
- ρ_0, ρ_1: Density

Convolutional model

- Signal recorded at the surface:

$$a(t) = r_0 \frac{s(t - \phi_0)}{2d_0} + r_1 \frac{s[t - (\phi_0 + \phi_1)]}{2d_0 + 2d_1}, \tag{3}$$

Convolutional model

- Neglect spherical spreading
- General model

$$a(t) = \sum_{k=0}^{N} r_k s(t - \tau_k), \tag{4}$$

- $\tau_k = \sum_{l=0}^{k} \phi_l$ two way traveltime to interface no k.

Convolutional model

- $\tau_k = k\Delta t$.

$$a_k = \sum_{l=0}^{N} r_l s_{k-l}, \tag{5}$$

$$\begin{aligned} a_k &= a(t = k\Delta t), \\ s_k &= s(t = k\Delta t). \end{aligned} \tag{6}$$

Continuous case:

$$a(t) = \int_{\tau=0}^{+\infty} r(\tau) s(t-\tau), \tag{7}$$

Convolutional model

- Vertical wave propagation
- Neglect vertical spreading
- Only primary reflections
- Earth is a linear filter
- $r(t)$ Earth's impulse response

Deconvolution

- Convolutional model for

seismic waves

$$y(t) = r(t) * s(t), \tag{8}$$

- $s(t)$: seismic source
- $r(t)$: reflectivity

Deconvolution

- Recover $r(t)$ when
- $s(t)$ given
- $y(t)$ given

Spiking deconvolution

- Fourier transformation of (8)

$$Y(f) = R(f)S(f) \tag{9}$$

- Solution

$$R(f) = \frac{Y(f)}{S(f)}. \tag{10}$$

Deconvolution

- Equation (10) slightly reformulated

$$R(f) = S^{-1}(f)Y(f), \tag{11}$$

- $S^{-1} = 1/S(f)$: *inverse* filter.
- S^{-1} removes the source
- recovers reflection coefficients

Deconvolution

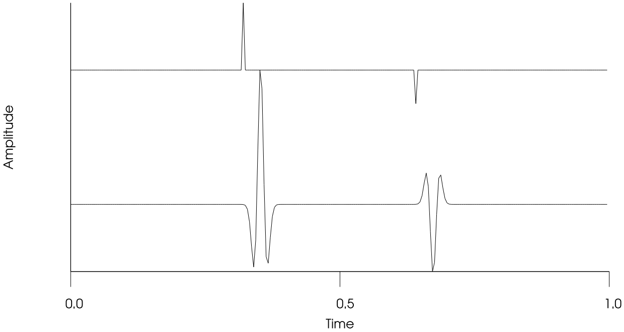


Figure: Spiking decon

The spectrum of a Ricker pulse and it's inverse

Deconvolution

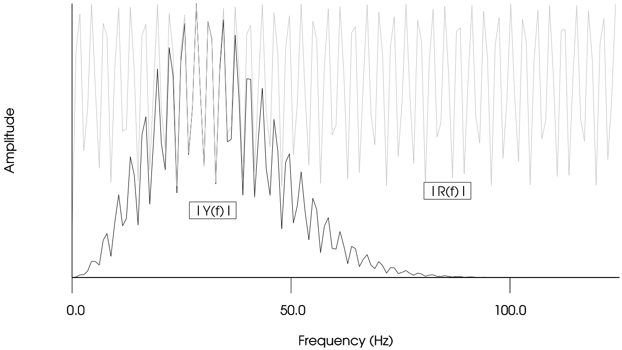


Figure: Deconvolution spectrum

The spectrum of the input data $|G(f)|$ and the deconvolved output $|R(f)|$.

Deconvolution

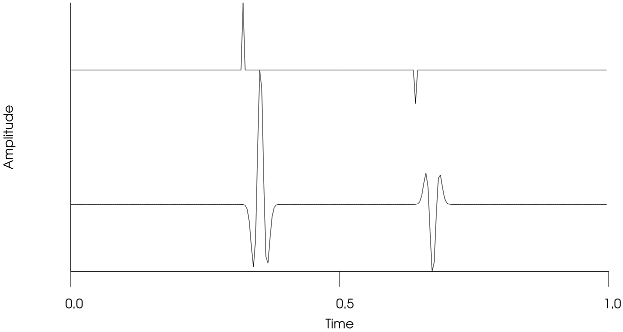


Figure: Deconvolution

Deconvolution

- Real seismic data contains noise

$$Y(f) = R(f)S(f) + N(f), \tag{12}$$

- $N(f)$: White (random) noise

$$R(f) = S^{-1}(f)Y(f) - S^{-1}(f)N(f). \tag{13}$$

Deconvolution

$S^{-1}(f)$ modified:

$$S^{-1}(f) = \frac{S^*(f)}{|S(f)|^2 + \epsilon}, \tag{14}$$

whereϵ is a small constant.
Inserting equation (14) into equation (13):

$$R(f) = \frac{S^*(f)}{|S(f)|^2 + \epsilon} Y(f) - \frac{S^*(f)}{|S(f)|^2 + \epsilon} N(f). \tag{15}$$

Increase ϵ to become much larger than $|s(f)|^2$

$$R(f) \approx \left(\frac{1}{\epsilon}\right) S^*(f)[Y(f) - N(f)]. \tag{16}$$

match-filtering is a stable approach for deconvolution.

Deconvolution

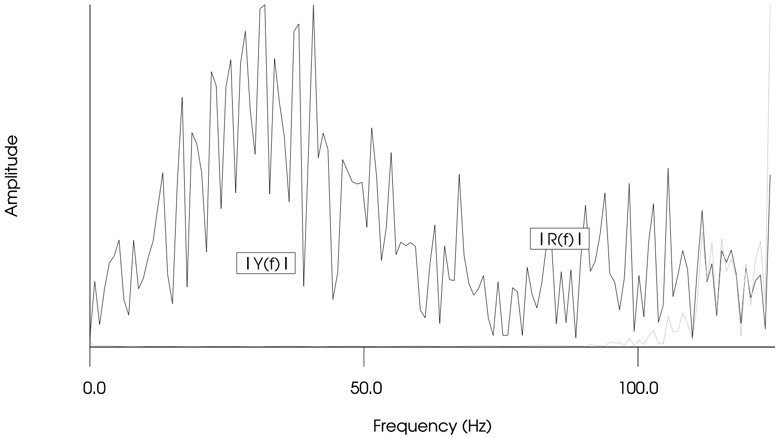


Figure: The spectrum of the input data with added noise $|Y(f) + N(f)|$ and the deconvolved output $|R(f)|$ (gray line).

Deconvolution

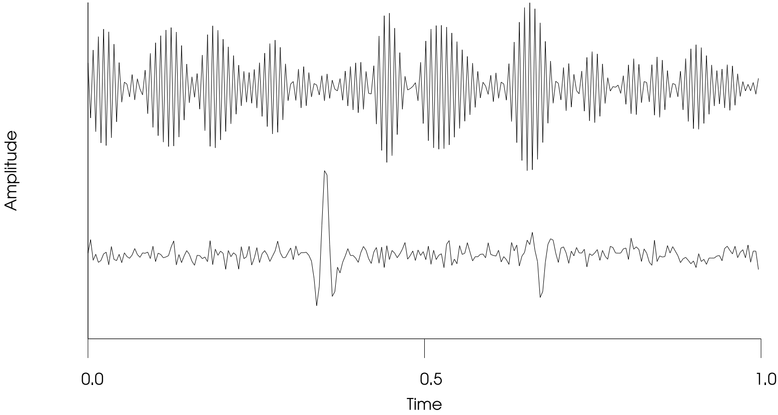


Figure: The input seismic data with added noise is shown in the lower trace, while the ouput of a spiking deconvolution filter applied to the input data is shown in the upper trace.

Deconvolution

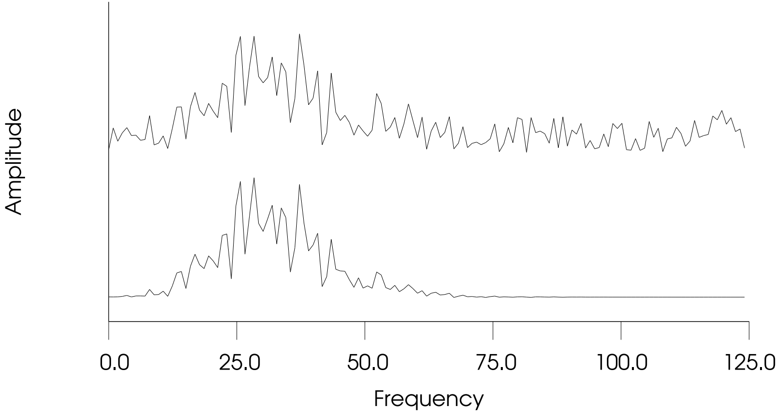


Figure: The spectrum of the input data with added noise $|Y(f) + N(f)|$ and the stabilized deconvolved output $|R(f)|$ (gray line).

Deconvolution

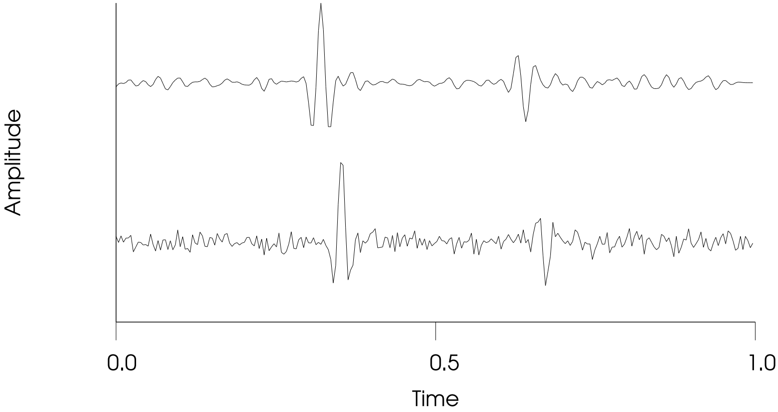


Figure: The input seismic data is shown in the lower trace, while the ouput of a match filter applied to the input data is shown in the upper trace.

Linear least-squres inversion

The convolutional model for seismic data

$$y(t) = r(t) * s(t), \tag{17}$$

- $y(t)$: Simulated Seismic data
- $r(t)$: Reflectivity
- $s(t)$: Seismic source pulse

$y(\hat{t})$: Observed seismic data

Linear least-squres inversion

We want to estimate $r(t)$ by minimizing

$$e = \frac{1}{T}\int_{-\infty}^{+\infty} dt\,[\hat{y}(t) - y(t)]^2, \tag{18}$$

Sampled functions

$$e = \frac{1}{N}\sum_{k=0}^{N}[\hat{y}_k - y_k]^2. \tag{19}$$

Inserting equation (5) into equation (19)

$$e = \frac{1}{N}\sum_{k=0}^{N}[\hat{y}_k - \sum_{l=0}^{N} r_l s_{k-l}]^2. \tag{20}$$

Linear least-squres inversion

Minimize error by

$$\frac{\partial e}{\partial r_m} = 0, \tag{21}$$

for $m = 0, 1, 2, ..., N$.
Applying to equation (20)

$$0 = \frac{-2}{N} \sum_{k=0}^{N} \left[\hat{y}_k - \sum_{l=0}^{N} s_{k-l} r_l \right] s_{k-m}. \tag{22}$$

Linear least-squres inversion

After reorganizing and multiplication with $-N/2$

$$0 = \sum_{k=0}^{N} \hat{y}_k s_{k-m} - \sum_{k=0}^{N} \sum_{l=0}^{N} s_{k-l} s_{k-m} r_l, \tag{23}$$

To simplify introduce the *cross correlation*

$$\phi_{\hat{y}s}(m) = \sum_{k=0}^{N} \hat{y}_k s_{k-m}, \tag{24}$$

and the *autocorrelation*

$$\phi_{ss}(m) = \sum_{k=0}^{N} s_k s_{k-m}, \tag{25}$$

Equation (23) then becomes

$$\phi_{\hat{y}s}(m) = \sum_{l=0}^{N} \sum_{k=0}^{N} s_{k-l} s_{k-m} r_l. \tag{26}$$

Linear least-squres inversion

Make a change of the summation variable $k' = k - l$ which gives $k = k' + l$

$$\phi_{\hat{y}s}(m) = \sum_{l=0}^{N} \sum_{k'=-l}^{N-l} s_{k'} s_{k'-(m-l)} r_l. \tag{27}$$

To get

$$\phi_{\hat{y}s}(m) = \sum_{l=0}^{N} \phi_{ss}(m-l) r_l, \tag{28}$$

for $m = 0, 1, , , , N$. Equation (28) is known as Levinson's *normal equations*.

Linearized inversion and deconvolution

Solution of equation (28) corresponds to deconvolution in the frequency domain given by equation (14) with $\epsilon = 0$.

$$\phi_{ab}(t) = \int_{-\infty}^{+\infty} d\tau \, a(\tau) b(\tau - t), \tag{29}$$

which is the same as

$$\phi_{ab}(t) = \int_{-\infty}^{+\infty} d\tau \, a(\tau + t) b(\tau). \tag{30}$$

The last equation is proved by substitution of variables by $u = \tau - t$.

The auto-correlation becomes

$$\phi_{aa}(t) = \int_{-\infty}^{+\infty} d\tau \, a(\tau) a(\tau + t). \tag{31}$$

The Fourier transform of equation (30) is

$$\Phi_{ab}(f) = \int_{-\infty}^{+\infty} dt \int_{-\infty}^{+\infty} d\tau \, \exp(-2\pi i f t) a(\tau + t) b(\tau), \tag{32}$$

Linearized inversion and deconvolution

which becomes after a change of variable $u = \tau + t$:

$$\Phi_{aa}(f) \int_{-\infty}^{+\infty} d\tau \ [\exp(2\pi i f \tau) b(\tau)]^* \int_{-\infty}^{+\infty} du \ \exp(-2\pi i f u) a(u), \quad (33)$$

which is the same as

$$\Phi_{ab}(f) = A(f) B^*(f). \quad (34)$$

If $a = b$ we get

$$\Phi_{aa}(f) = A(f) A^*(f) = |A(f)|^2. \quad (35)$$

The Fourier transform of the auto-correlation is the square of the amplitude spectrum, which is also called the power spectrum.

Linearized inversion and deconvolution

Fourier-transform equation (28) and use equations (34) and (35) :

$$\hat{Y} S^*(f) = |S(f)|^2 R(f), \tag{36}$$

Solving equation (36) for $R(f)$ we get

$$R(f) = \frac{Y\hat{(}f)S^*(f)}{|S(f)|^2} \tag{37}$$

which is exactly the same solution as given by equation (19) but with $\epsilon = 0$.

Linearized inversion and deconvolution

To stabilize the solution add a damping-constant ϵ in the error function given by equation (19)

$$e = \frac{1}{N}\sum_{k=0}^{N}[\hat{y}_k - y_k]^2 + \epsilon \sum_{k=0}^{N} r_k^2. \tag{38}$$

This gives the normal equations

$$\phi_{\hat{y}s}(m) = \sum_{l=0}^{N}[\phi_{ss}(m-l) + \epsilon\delta_{m,l}]r_l, \tag{39}$$

where the Kronecker symbol $\delta_{m,l}$ is defined as

$$\delta_{m,l} = \begin{cases} 1 & m = l \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

A Fourier-transform of equation (39) gives exactly the stabilized deconvolution solution in equation (37).

Predictive deconvolution

- Predict the input y_k at future time $y_{k+\alpha}$.

$$y_{k+\alpha} = \sum_{l=0}^{M} p_l y_{k-l}. \tag{41}$$

- Use Wiener filtering
- Input: y_k
- Desired output: $y_{k+\alpha}$.

- Right hand side crosscorrelation:

$$\phi_{y_{k+\alpha}y_k}(m) = \sum_{l=m}^{M} y_{l+\alpha}y_{l-m}. \tag{42}$$

- Change of variable $l' = l + \alpha$:

$$\phi_{y_{k+\alpha}y_k}(m) = \sum_{l'=\alpha+m}^{M} y_{l'}y_{l'-(m+\alpha)} = \phi_{yy}(m+\alpha). \tag{43}$$

- The normal equations then become

$$\sum_{l=0}^{M} \phi_{yy}(l-m)p_l = \phi_{yy}(m+\alpha), \tag{44}$$

for $m = 0, 1, \ldots, M$.

- Prediction error

$$\epsilon_k = y_{k+\alpha} - \hat{y}_{k+\alpha}. \tag{45}$$

- Prediction error represents non-predictable part of input data
- If input data is stationary, white and random with zero mean

$$\begin{aligned} \phi_{xx}(m) &= 0, \text{ for } m \neq 0 \\ \phi_{xx}(m) &= \sigma^2. \text{ for } m = 0 \end{aligned} \tag{46}$$

- Right hand side is zero for nonzero valuesof the prediction distance
- Filter coefficients zero
- Predicted value zero, which is mean of input data

Inverse filters for multiples

- Replace the predicted value with equation (41)

$$\epsilon_{k+\alpha} = y_{k+\alpha} - \sum_{l=0}^{N} p_l y_{k-l}. \tag{47}$$

- z-transform of equation (45)

$$A(z) = \sum_{k=0}^{N} a_k z^k \tag{48}$$

$$\sum_{k=0}^{N} z^k \epsilon_{k+\alpha} = \sum_{k=0}^{N} z^k y_{k+\alpha} - \sum_{j=0}^{N} z^k \sum_{l=0}^{N} p_l y_{k-l}, \tag{49}$$

$$z^{-\alpha} \sum_{m=k+\alpha}^{N+\alpha} z^m \epsilon_m = z^{-\alpha} \sum_{m=k+\alpha}^{N+\alpha} z^m y_m - \sum_{j=0}^{N} z^k \sum_{l=0}^{N} p_l y_{k-l}. \tag{50}$$

Predictive deconvolution

$$z^{-\alpha}E(z) = z^{-\alpha}Y(z) - P(z)Y(z). \tag{51}$$

$$E(z) = [1 - z^{\alpha}P(z)]Y(z). \tag{52}$$

Predictive deconvolution

- The filter

$$[1 - z^{\alpha} P(z)], \tag{53}$$

- Z-transform of the time-domain filter

$$p_k = 1, 0, 0, 0, 0, \ldots, 0, -p_0, -p_1, \ldots, -p_N, \tag{54}$$

- The filter contains $\alpha - 1$ zeros.
- For $\alpha = 1$

$$p_k = 1, -p_0, -p_1, \ldots, -p_N, \tag{55}$$

- Statistical Wiener spiking deconvolution filter.

Predictive deconvolution

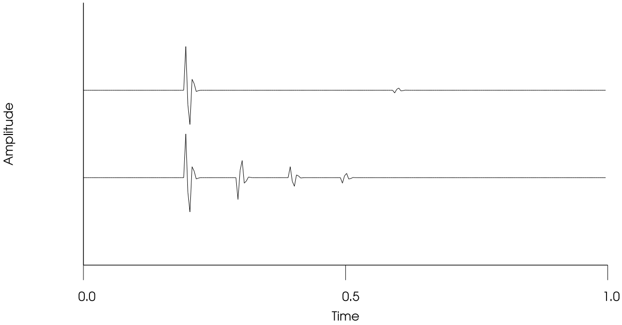


Figure: Input seismic data with multiples (lower trace) and output from multiple inverse filter (top)

Predictive deconvolution

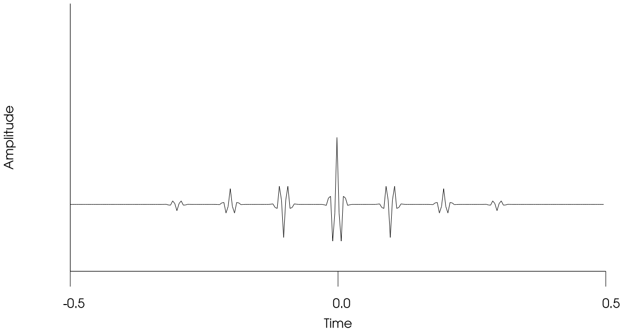


Figure: Autocorrelation of the input data shown in figure 9

Predictive deconvolution

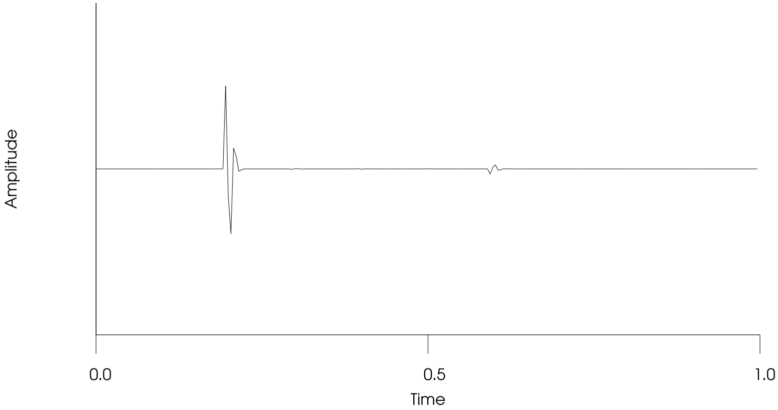
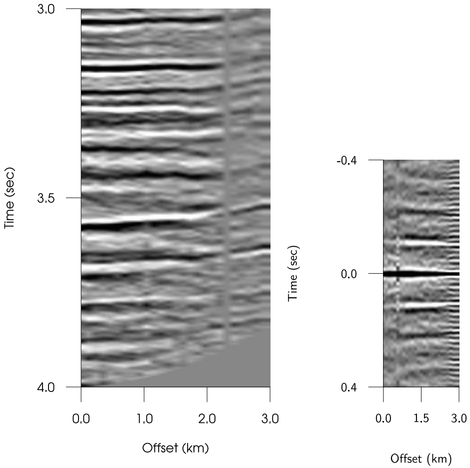
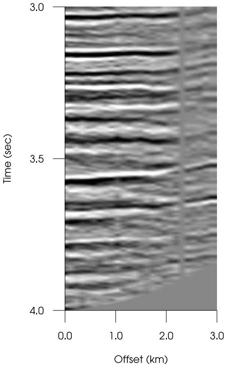


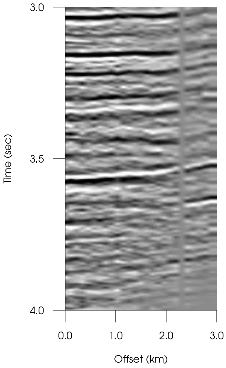
Figure: Prediction error filter applied to the input data shown in figure 9

Predictive deconvolution

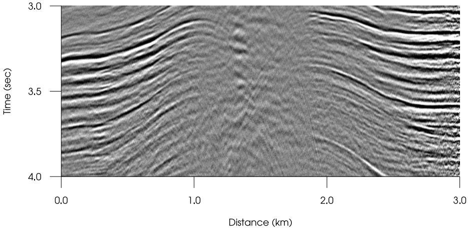


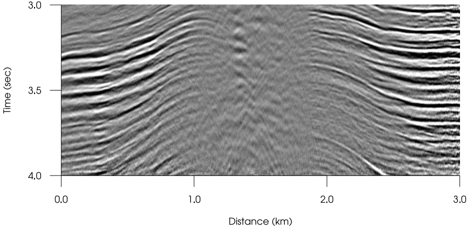
Predictive deconvolution





Predictive deconvolution





Predictive deconvolution