Coherent noise attenuation: A synthetic and field example

Antoine Guitton[1]

ABSTRACT

Noise attenuation using either a filtering or a subtraction scheme is achieved as long as the prediction error filter (PEF), which (1) filters the coherent noise in the first method and (2) models the noise in the second one, can be accurately estimated. If a noise model is not known in advance, I propose estimating the PEF from the residual of a previous inverse problem. At this stage, the filtering and subtraction method give similar results on both synthetic and real data. However the subtraction method can more completely separate the noise and signal when both are correlated.

INTRODUCTION

In a previous report (Guitton, 2000), I presented two methods specifically designed to obtain independent and identically distributed (*iid*) residual components. When this requirement is met, the inversion of the data becomes stable with fast convergence toward a minimum and attenuation of the coherent noise present in the data. These two inversion strategies are (1) use of a prediction error filter (PEF) to approximate the noise covariance matrix and (2) addition of a coherent noise modeling operator in the fitting goal. The first method filters the coherent noise while the second separates the coherent noise by subtraction. The subtraction method has the advantage to mitigate the correlation between signal and noise by introducing a regularization term into the inverse problem.

This paper is divided into two parts. Firstly, I use the filtering method to attenuate coherent noise with both synthetic and real data. I show that the signal/noise separation is efficient as long as the PEF used to approximate the noise covariance matrix renders the noise spectrum accurately. Secondly, I use the subtraction method on the same data to separate noise from signal. Although both methods yield similar attenuation of the coherent noise, I anticipate that the subtraction method has the potential to handle the correlation between the noise and signal.

APPROXIMATING THE NOISE COVARIANCE MATRIX WITH A PEF

To address the delicate problem of estimating the noise covariance matrix (Tarantola, 1987), while building on Claerbout and Fomel (1999), I have proposed to use a PEF for the approx-

[1]**email:** antoine@sep.stanford.edu

imation (Guitton, 2000). The PEF is estimated from a noise model or from the residual of a previous inversion. In this paper, I show that both methods filter the noise components as long as the PEF incorporates enough spectral information for the noise. Because the noise covariance matrix is supposed to filter out the inconsistent part of the data (the coherent noise), I call this method the filtering method. It is based on the following fitting goal:

$$\mathbf{0} \approx \mathbf{A_n}(\mathbf{Hm} - \mathbf{d}), \tag{1}$$

where $\mathbf{A_n}$ is a PEF estimated from the residual or from a noise model, $\mathbf{H}$ denotes a seismic operator, $\mathbf{m}$ is the model we seek, and $\mathbf{d}$ the seismic data. The corresponding least-squares inverse, or the pseudo-inverse of $\mathbf{m}$, is given by the equation

$$\hat{\mathbf{m}} = (\mathbf{H}'\mathbf{A}_\mathbf{n}'\mathbf{A_n}\mathbf{H})^{-1}\mathbf{H}'\mathbf{A}_\mathbf{n}'\mathbf{A_n}\mathbf{d}, \tag{2}$$

where $(')$ is the adjoint operator. The model space is computed iteratively rather than by using the direct inversion described in equation (2). I use two distinct strategies to compute the PEF $\mathbf{A_n}$ needed in equation (1). With the first strategy, I derive a noise model from which a PEF is estimated and kept unchanged, whereas with the second one, I compute the PEF from the residual. I have slightly modified my inversion scheme described in my earlier report (Guitton, 2000). I propose the following algorithm when the PEF is estimated from the residual:

1. Solve the inverse problem for the fitting goal $\mathbf{0} \approx$ **Hm - d**.
2. Estimate a PEF $\mathbf{A_n}$ from the residual when only coherent noise remains in the residual.
3. Restart the inverse problem ($\mathbf{m} = \mathbf{0}$) for the fitting goal in equation (1).
4. Iterate with the new PEF.
5. Reestimate the PEF $\mathbf{A_n}$ from the residual **Hm - d**.
6. Go to step (4).
7. Stop when the residual has a white spectrum.

The novelty of this algorithm is that the first PEF is not estimated from the data but rather from the residual of a previous inverse problem where no PEF is used in the fitting goal. This has the advantage of isolating the coherent noise more accurately, thus furnishing a satisfying noise model for the PEF estimation. This algorithm can be seen as a two-stage process in which the first stage helps to estimate a first PEF for our inversion and the second stage reestimates the PEF iteratively. Determining when to stop the inversion to compute the first PEF is critical: too many iterations and the coherent noise is fitted, too few and some signal remains in the residual. The filtering method is viable if the operator $\mathbf{H}$ does not model the coherent noise properly. In other words, the signal is expected to be orthogonal to the coherent noise for proper attenuation.

The sections that follow describe the application of this method with synthetic and real data. For each case, I estimate the PEF from a noise model or from the residual of a previous inversion. My results demonstrate that both methods lead to a proper attenuation of the noise components.

Filtering the coherent noise in synthetic data

I designed a synthetic case where noise and signal are completely separable. Figure 1a shows the synthetic CMP gather. The data are made of eleven hyperbolic events overlaid by a monochromatic event plus some Gaussian zero mean random noise. **H** is the velocity stack operator. In the next two sections, I show that the noise filtering is achieved with or without a noise model.

- **Filtering with a noise model**

The noise model I use is shown in Figure 2a; it contains both the monochromatic event and the random noise. The first parameter I set is the size of the PEF. Because the coherent noise of this dataset was extremely predictable, it seemed that a one dimensional PEF with three coefficients PEF should suffice, but surprisingly, it did not. Figure 2b displays the inverse spectrum of the PEF with three coefficients (`a=3,1`). Clearly, the estimated noise spectrum does not resemble the coherent noise spectrum in Figure 1b. Using 30 coefficients (`a=30,1`) significantly improved the matching (Figure 2c). According to this result, I chose a 30 coefficients PEF and iterated for the fitting goal in equation (1).

Figures 3a and b show, respectively, the estimated model space after inversion and the reconstructed data. No footprint of the coherent noise appears in either space. In addition, the residual (Figure 3c) is reasonably white, indicating that the inversion algorithm converged. The "real" residual in Figure 3d, which measures the difference between the input data and the reconstructed data, displays only the undesirable coherent noise.

- **Filtering without a noise model**

The PEF (`a=30,1`) is now iteratively updated. The first stage of this method is to iterate with no PEF in the fitting goal and then to estimate the PEF after a certain number of iterations from the residual. I iterated 10 times before estimating the filter. Figure 4a displays the residual after 10 iterations and its corresponding spectrum in Figure 4b. As expected, the residual is mostly noise, which makes the first PEF estimation reliable.

In the second stage, after the first PEF calculation, the model space is reset to zero and the conjugate gradient (the iterative solver I used) is restarted. Figure 4c shows the residual after the first iteration. The coherent noise has been properly attenuated. The spectrum in Figure 4d shows almost no trace of the monochromatic event. I then recomputed the PEF after every 10 iterations. The outcome of this process, displayed in Figure 5, is comparable to that in Figure 3.

Filtering the coherent noise in real data

The real data example I discuss here inspired the preceding synthetic gather. Figure 6a shows the CMP gather, 6b its amplitude spectrum, 6c the noise model obtained by simple low-pass

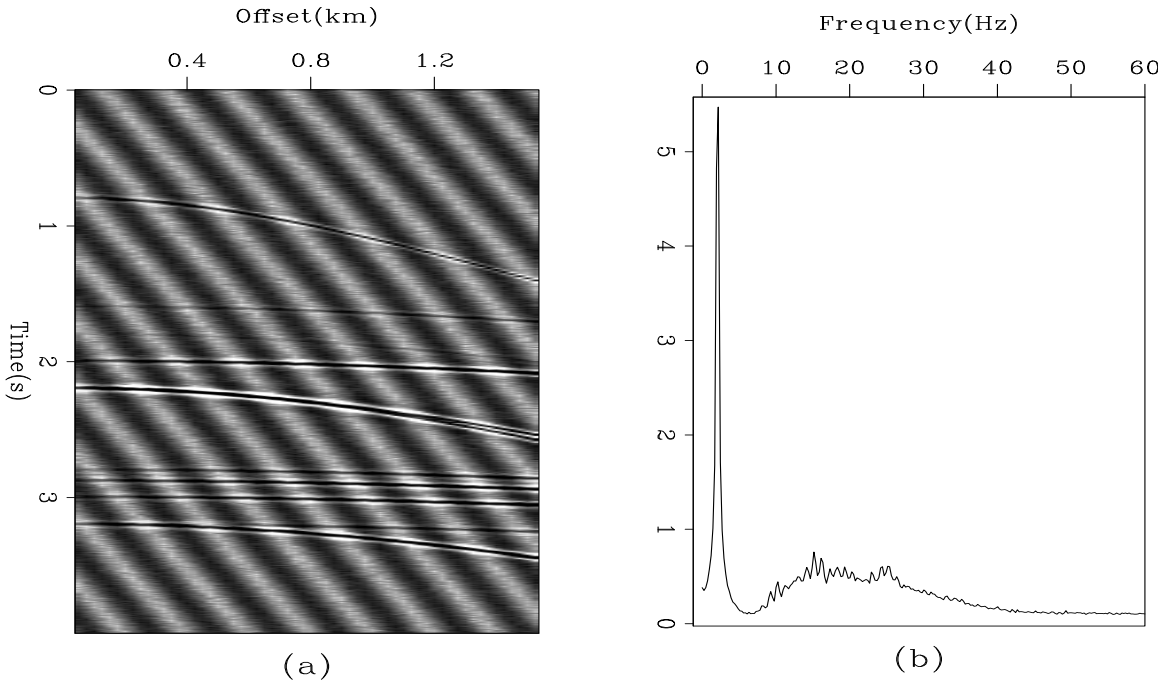

Figure 1: (a) Synthetic data. (b) The amplitude spectrum of the data in panel a. antoine1-datasynth [ER]

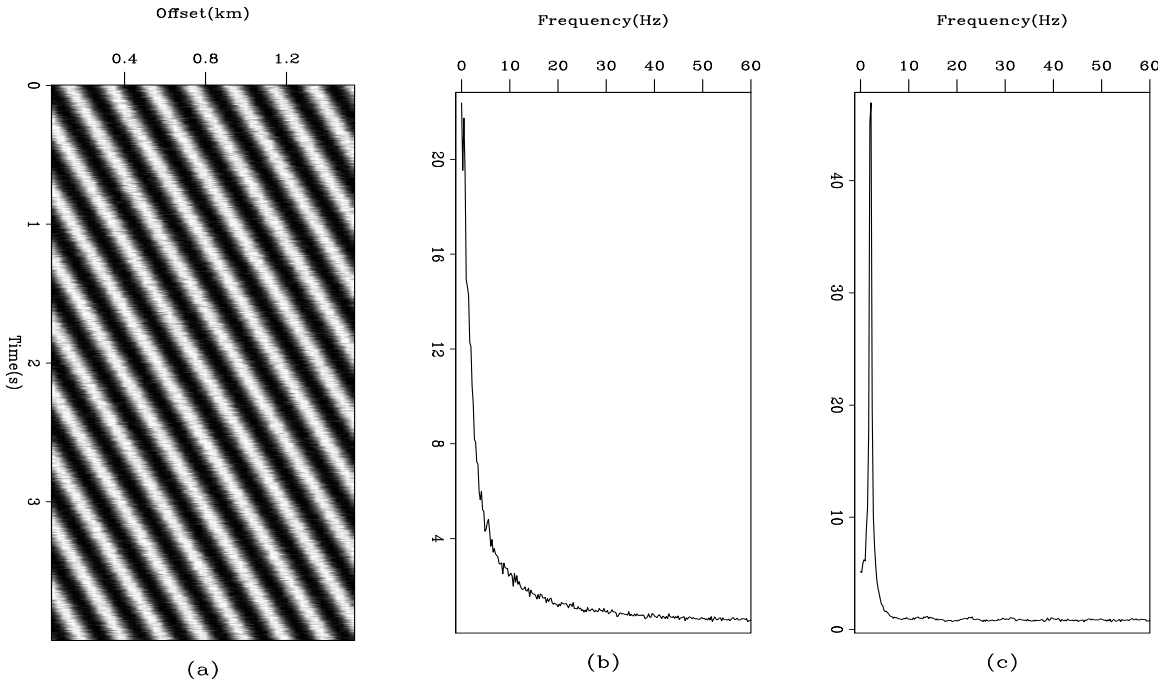

Figure 2: (a) A coherent noise model used for the synthetic data examples. (b) The inverse spectrum of the PEF estimated from the noise model, with 3 coefficients. (c) The inverse spectrum of the PEF estimated from the noise model, with 30 coefficients. antoine1-pefs [ER]

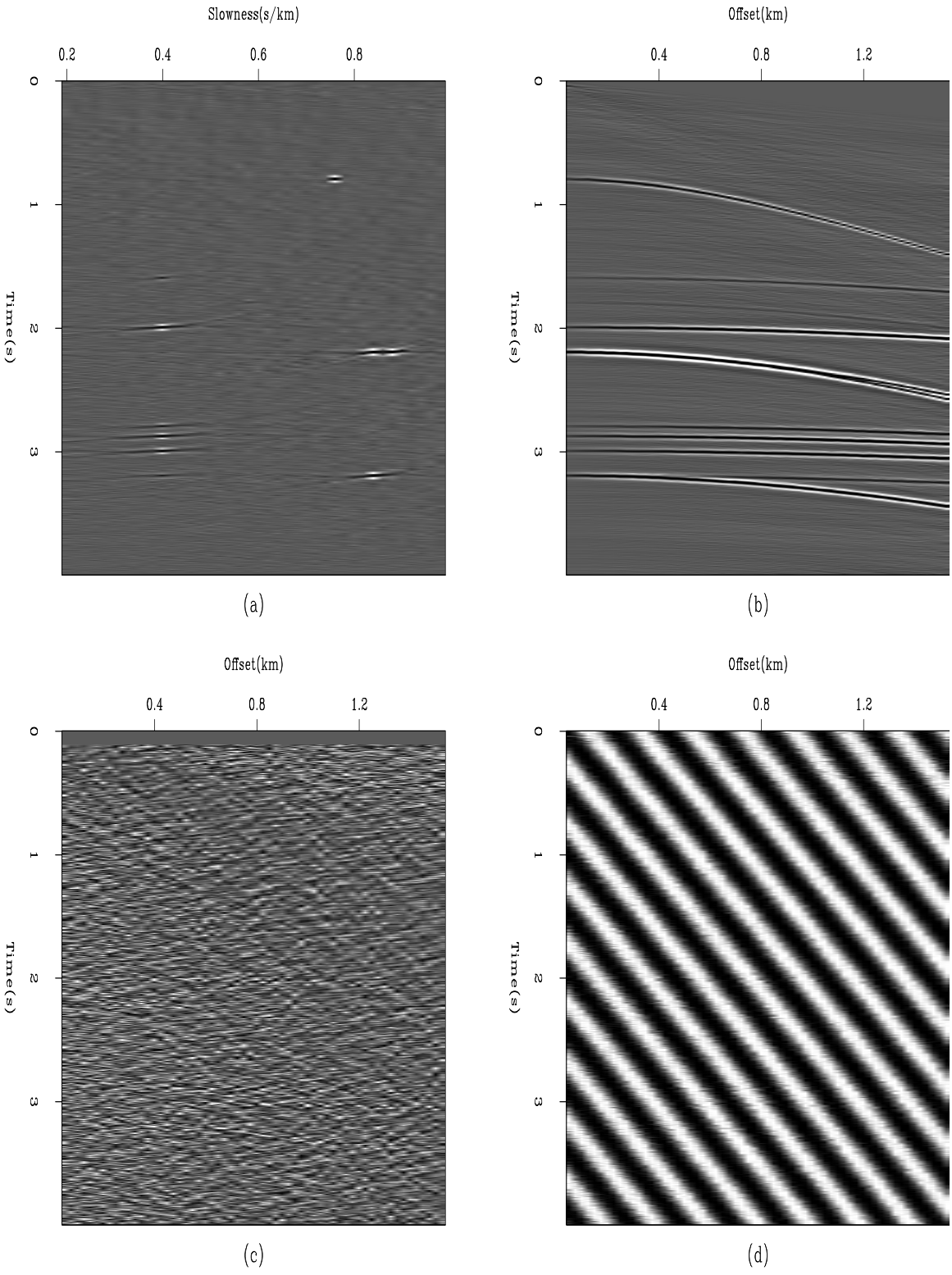

Figure 3: Filtering the coherent noise in synthetic data with a noise model. (a) An estimated model space. (b) Reconstructed data using the model space. (c) The weighted residual ($\mathbf{r} = \mathbf{A_r}(\mathbf{Hm} - \mathbf{d})$) after inversion. (d) The difference between the input data in Figure 1a and the reconstructed data in 3b. antoine1-c-synth [ER,M]

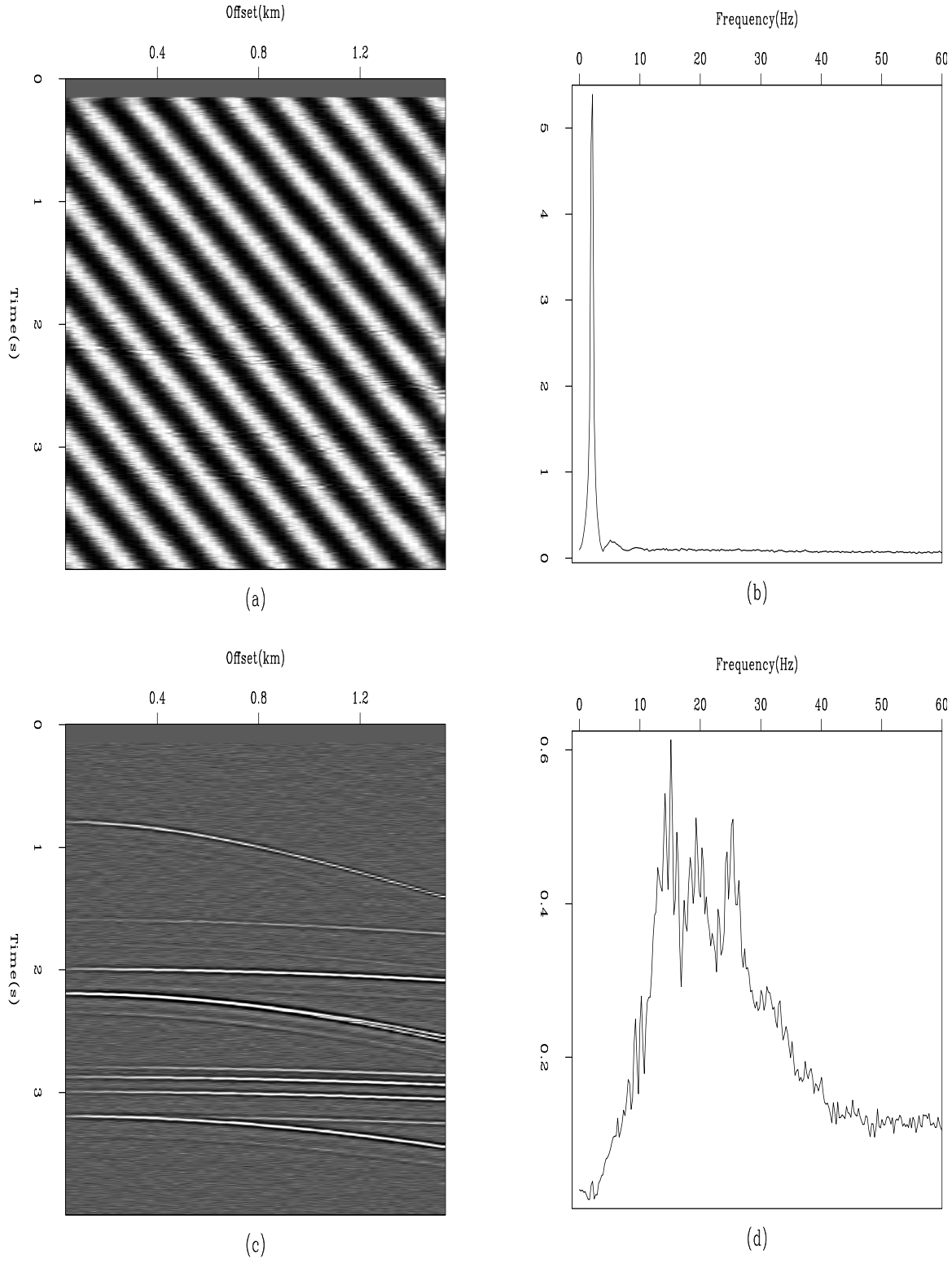

Figure 4: (a) The residual of the filtering scheme without a PEF after 10 iterations. (b) The amplitude spectrum of the residual. (c) The residual after the first iteration with a PEF (estimated from the residual in 4a) in the fitting goal. (d) The amplitude spectrum of the residual in 4c. The noise spectrum has vanished from the residual. antoine1-s-synth-rec [ER]

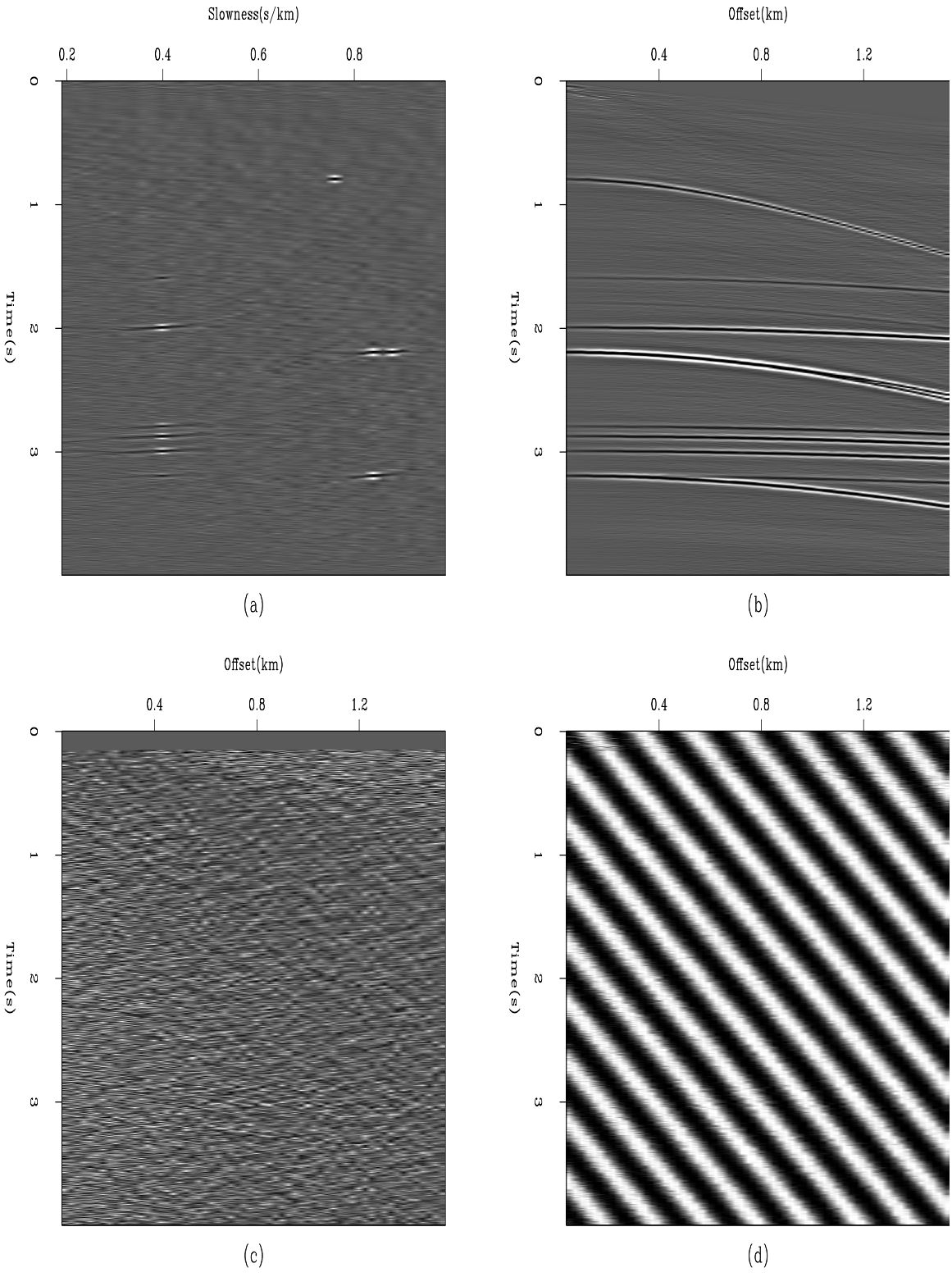

Figure 5: Filtering the coherent noise in synthetic data without a noise model.(a) An estimated model space. (b) Reconstructed data using the model in 5a. (c) The weighted residual ($\mathbf{r} = \mathbf{A_r}(\mathbf{Hm} - \mathbf{d})$) after inversion. (d) The difference between the input data in Figure 1a and the reconstructed data in 5b. antoine1-c-synth-rec [ER]

filtering of the CMP gather plus random noise, and 6d the amplitude spectrum of the model. Three types of noise appear in the data:

1. A low-velocity, low-frequency event that corresponds to the main target of the noise attenuation process.
2. Amplitude anomalies that are both local (offset 0.9 km and time 2.7 s) and trace related (offset 1.9 km).
3. A time shift near offset 2 km.

The next two sections present the results of the noise filtering methods when the real data set is used, demonstrating that the coherent noise can be filtered with a single 1-D PEF.

- **Filtering with a noise model**

In this particular case, because I was assuming stationary signal and noise, I used one PEF for the entire dataset. The coherent noise is not completely linear and the noncontinuity of the coherent noise as displayed in Figure 6c indicates that a one-dimensional PEF is preferable. I choose a 30-coefficient filter (`a=30,1`) to make sure that the PEF absorbs enough spectral information from the residual.

The result of the inversion is displayed in Figure 7. The residual (Figure 7c) is not perfectly white, but the coherent noise has been filtered out. Because the noise model does not incorporate them, the remaining artifacts are the amplitude anomalies.

The remaining very weak dipping event in the residual (Figure 7c) needs to be understood. To investigate this issue, I show in Figure 8a the spectrum of the input data; in 8b, the residual before the first iteration; and in 8c, the residual after the inversion ends. We can see that the estimated PEF has eliminated almost all the coherent noise spectrum with a remaining tiny peak at 10 Hertz. As the iterations go on (between Figures 8b and c), the peak does not change in amplitude and becomes relatively more important in the spectrum of the residual, which explains why some coherent information remains in the residual.

The bimodal shape of the spectrum in Figure 8c shows one mode around 10 Hertz for the coherent noise, and one mode around 30 Hertz for the amplitude anomalies. Because I did not reestimate the PEF iteratively, I will not get any whiter residual. I think that a nonstationary PEF would treat the noise more efficiently (Guitton et al., 2001). The limited accuracy of the filter I used for the entire residual caused the imperfect attenuation.

- **Filtering without a noise model**

As an alternative to using a noise model, I estimate the PEF iteratively. Once again, in the first stage of this method, I iterate with no PEF in the fitting goal and then estimate the PEF after a certain number of iterations from the residual. I iterated 35 times before estimating the first filter. The number of iterations turns out to be an important element in the final separation.

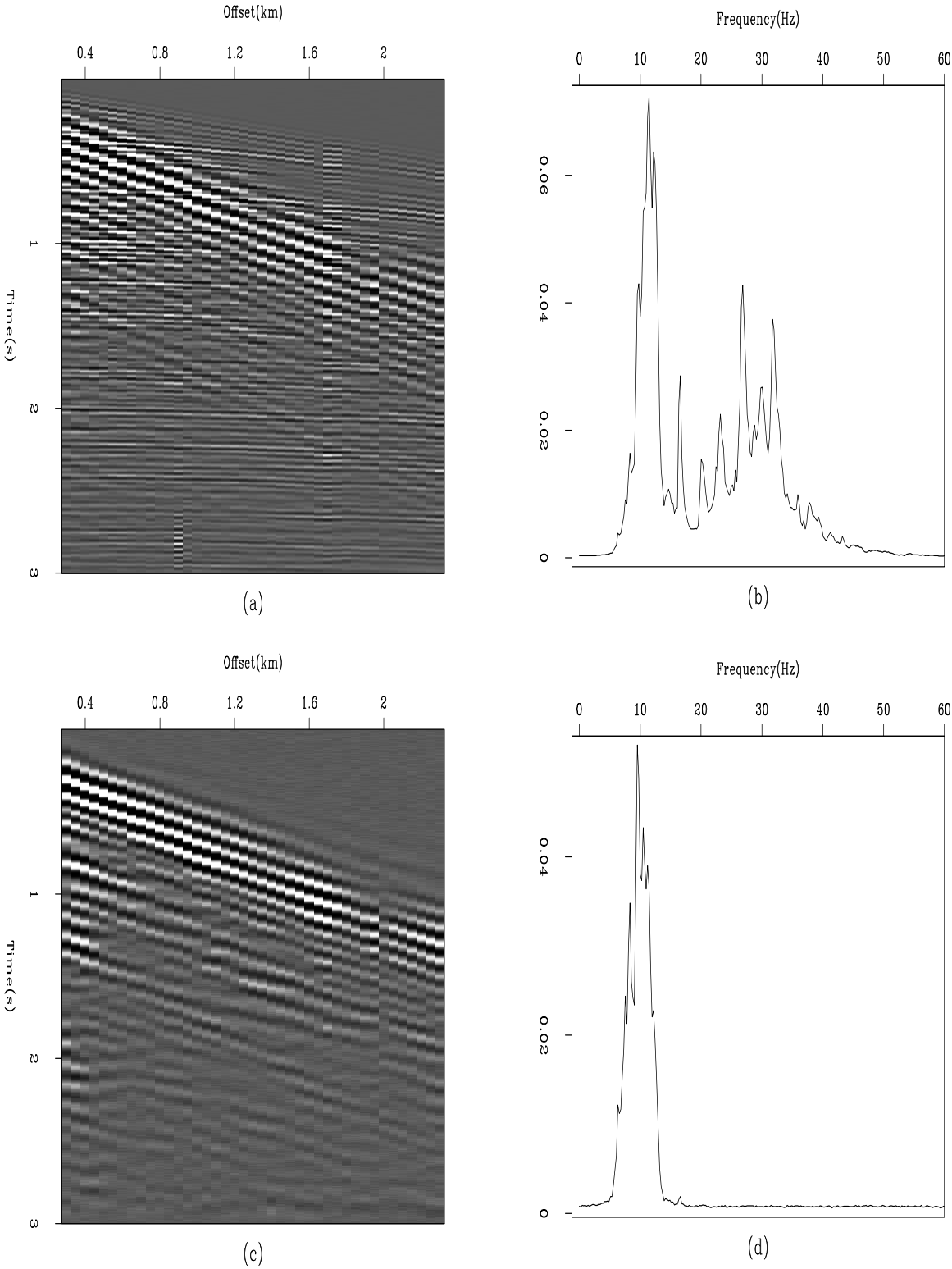

Figure 6: (a) A real CMP gather. (b) The amplitude spectrum of this gather. (c) A noise model obtained by low-passing the data and adding random zero-mean Gaussian noise. (d) The amplitude spectrum of the noise model. antoine1-datawz [ER]

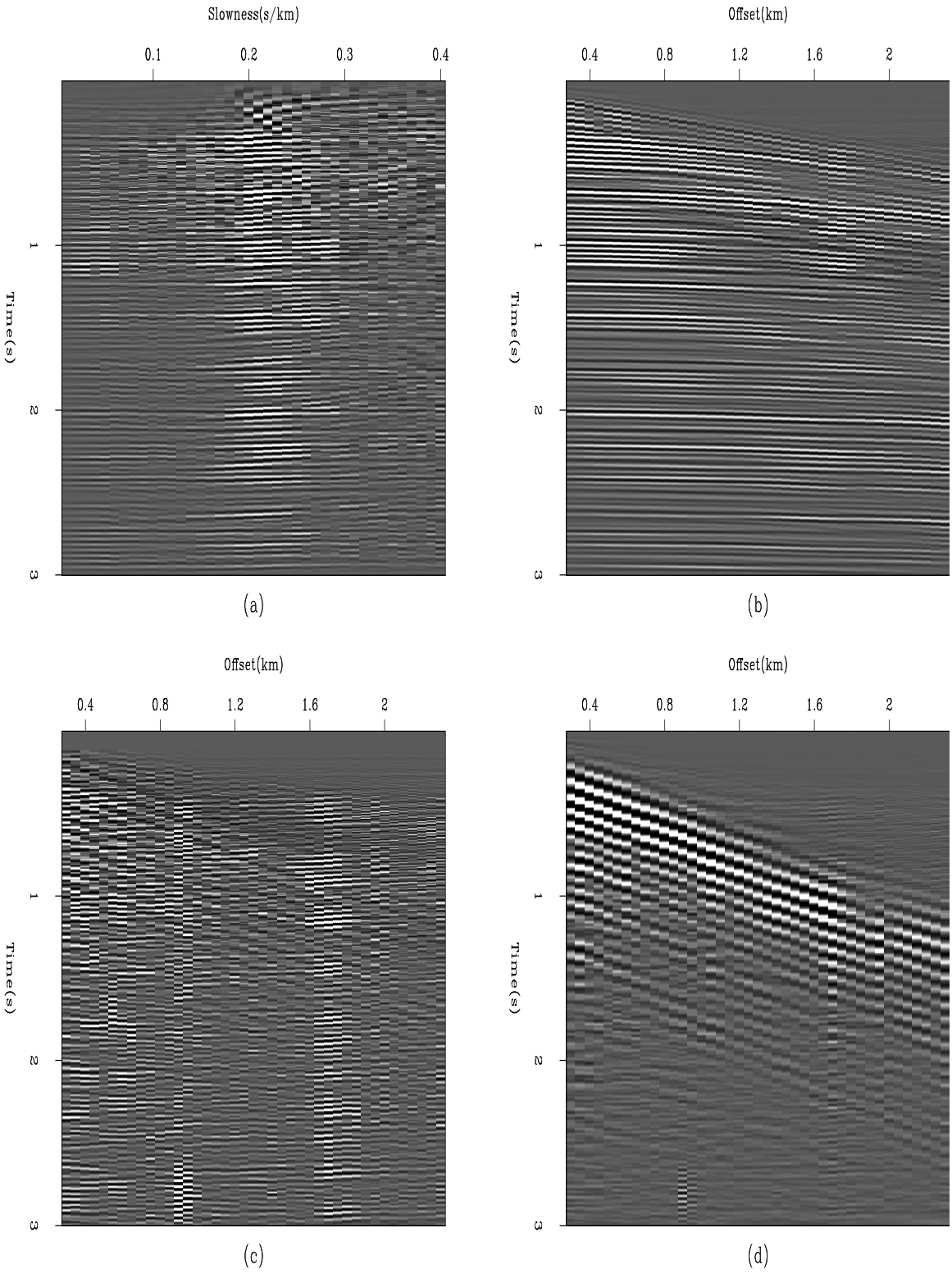

Figure 7: Filtering the coherent noise in real data with a noise model. (a) An estimated model space. (b) Reconstructed data using this model space. (c) The weighted residual ($\mathbf{r} = \mathbf{A_r}(\mathbf{Hm} - \mathbf{d})$) after inversion. (d) The difference between the input data in Figure 6a and the reconstructed data in 7b. antoine1-c-syntwz08 [ER,M]

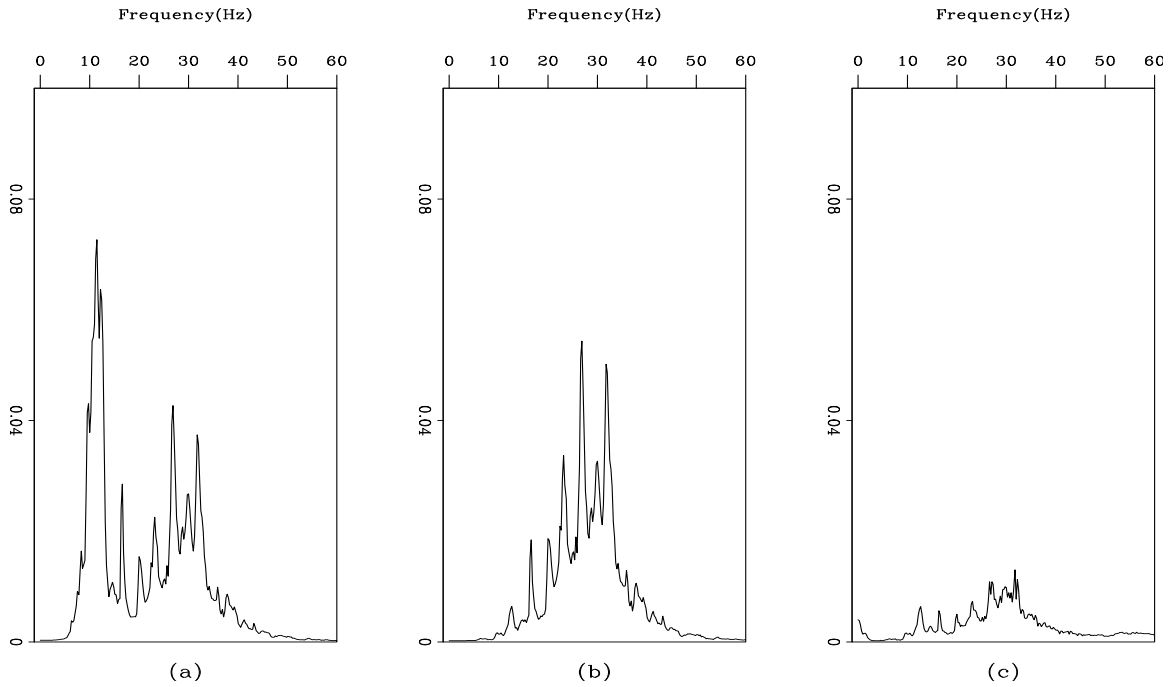

Figure 8: Filtering the coherent noise in real data with a noise model. (a) The amplitude spectrum of the input data in Figure 6a. (b) The amplitude spectrum of the residual before the first iteration. (c) The amplitude spectrum of the residual after inversion. antoine1-sp-wz08 [ER,M]

Figures 9a and b shows the spectrum of the residual during the first stage for 10 and 35 iterations. After 10 iterations, I have one peak at 10 Hertz, and a second peak around 30 Hertz. If I estimate a PEF after 10 iterations, it will absorb the two remaining events. The separation tends to be very sensitive to the second peak. Because it overlaps with the signal spectrum, the noise spectrum at 30 Hertz should be avoided in the PEF. I then decided to estimate a PEF after 35 iterations (Figure 9b) since the second peak was considerably attenuated. Figure 9c shows the inverse spectrum of the PEF estimated from the residual with the spectrum in Figure 9b. The main features of the spectrum are correctly absorbed in the filter.

In the second stage of the inversion scheme, I reestimated the PEF only twice during the remaining 38 iterations. There were 73 iterations in total, 35 iterations for the first stage and 38 for the second. Figure 10 displays the final results. In Figure 10c, the residual appears fairly white with little coherent information left. However, there remains a very weak background event with the same slope as the coherent noise I wish to attenuate.

To investigate this problem further, I compute the spectrum of the residual after the inversion ends. Figure 11a shows the spectrum of the input data; Figure 11b, the spectrum of the residual; and Figure 11c, the same spectrum at a smaller scale. The residual looks white with a slightly higher peak at 10 Hertz, which explains the residual in Figure 10c. Nonetheless, the final result is, I think, very satisfactory and compares favorably with the result displayed in Figure 7 when a noise model is known.

To demonstrate the usefulness of a PEF in the fitting goal (equation (1)) further, I computed a comparison, shown in Figure 12, of the normalized objective functions for the iterative scheme with and without a PEF. The PEF clearly improves the optimization.

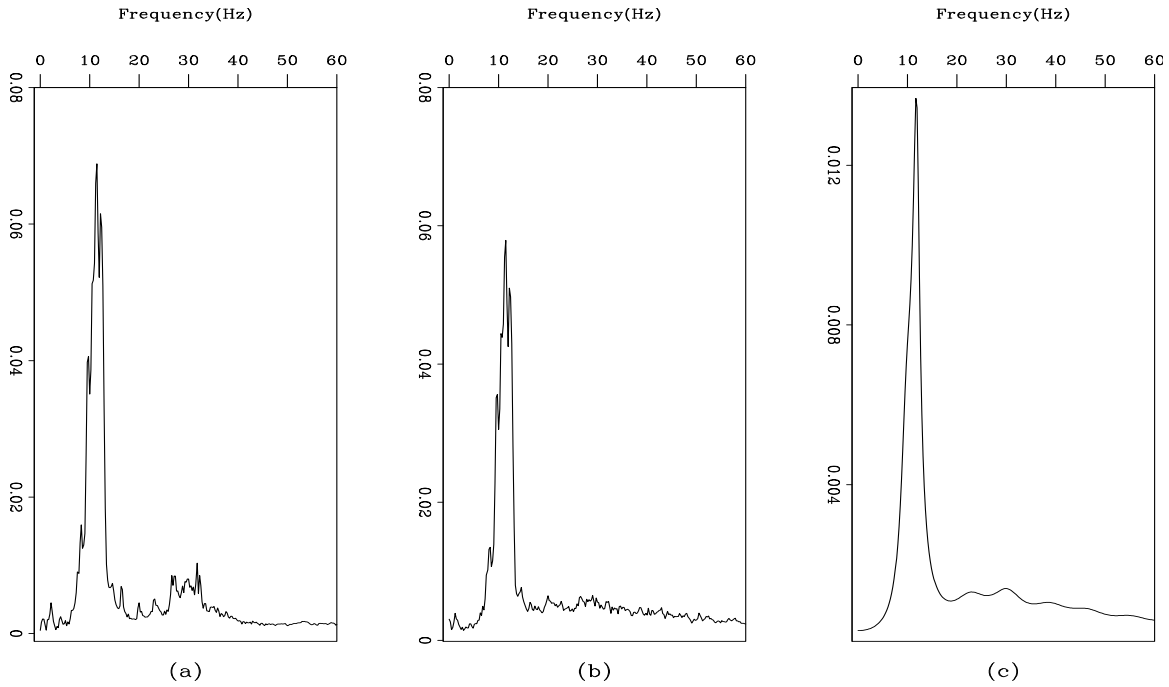

Figure 9: The amplitude spectrum of the residual in the first stage after (a) 10 iterations and (b) 35 iterations. (c) The inverse spectrum of the PEF estimated from the residual with the spectrum in 9b; as expected, 9c resembles 9b. antoine1-sp3-wz08 [ER]

Discussion of the filtering method

Using a PEF to approximate the covariance matrix (equation (1)) leads to a proper filtering of the coherent noise present in the data. Nonetheless, this PEF has to estimate accurately the noise pattern. If a noise model does not exist, the PEF can be calculated from the residual of a previous inverse problem in which no attempt has been made to attenuate the noise.

The next section describes what I call the subtraction method which does not filter the noise but rather predicts it. This method has the ability to decrease the effects of signal/noise correlations.

SUBTRACTING COHERENT NOISE

In contrast to the filtering method which attenuates the coherent noise by approximating the noise covariance matrix with a PEF, the following method aims to model the coherent noise

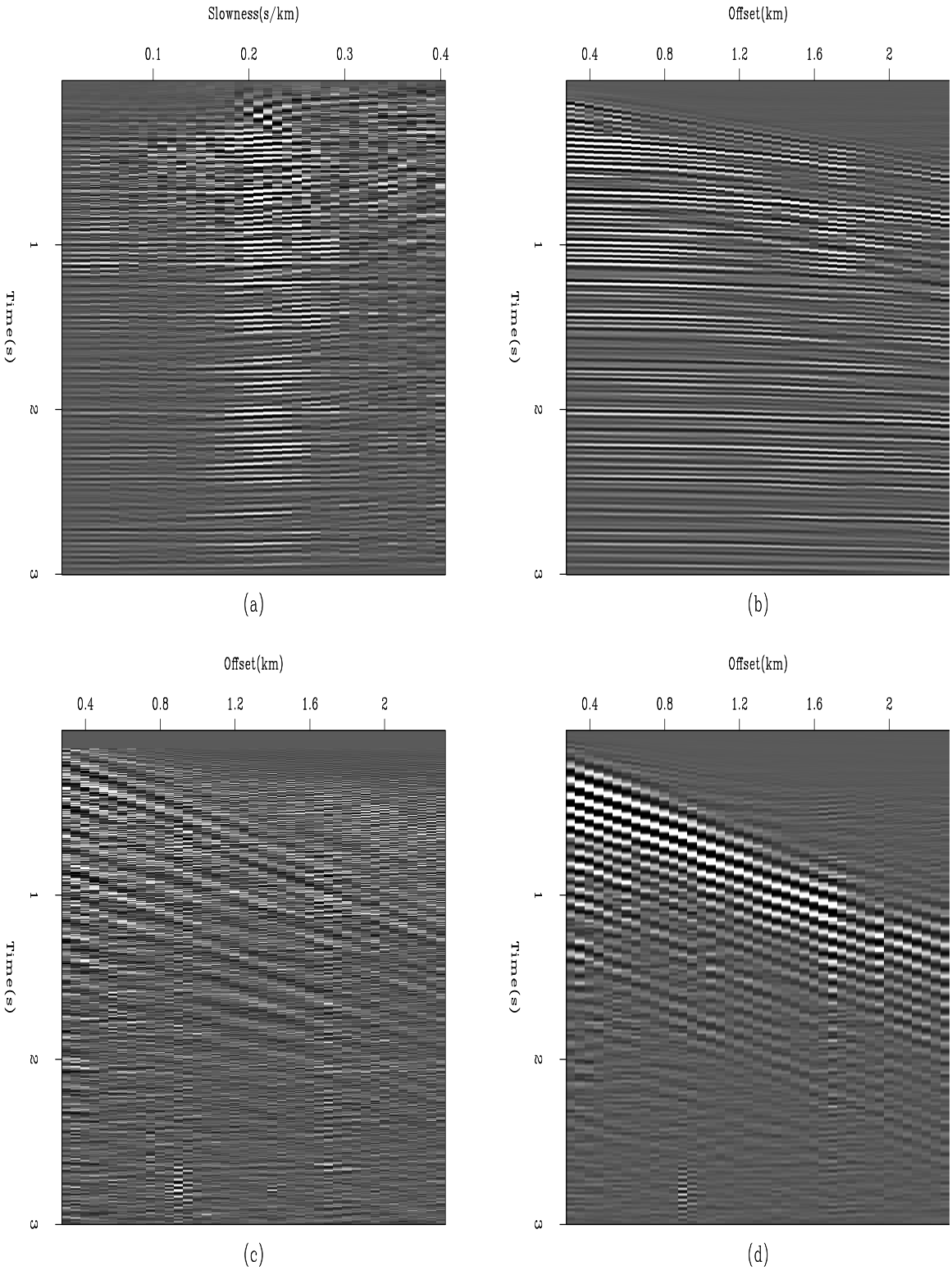

Figure 10: Filtering the coherent noise in real data without a noise model. (a) An estimated model space. (b) The reconstructed data using the model space. (c) The weighted residual ($\mathbf{r} = \mathbf{A_r}(\mathbf{Hm} - \mathbf{d})$) after inversion. (d) The difference between the input data in Figure 6a and the reconstructed data in 10b. antoine1-c-wz08-rec [ER]

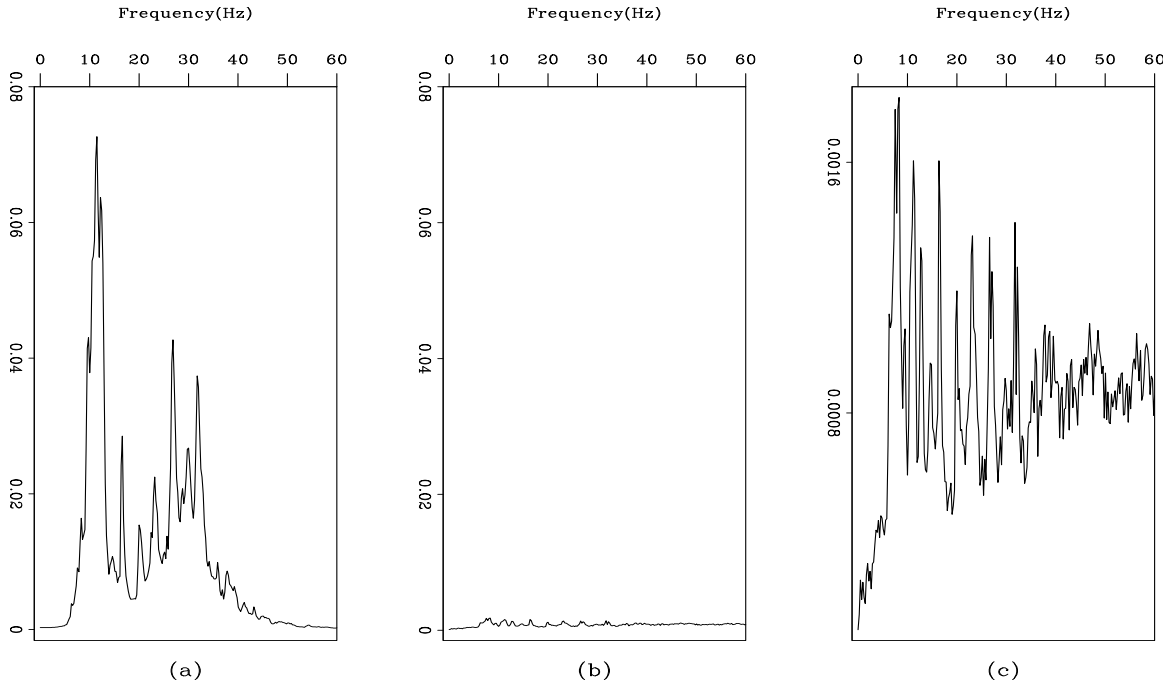

Figure 11: Filtering the coherent noise in real data without a noise model. (a) The amplitude spectrum of the input data in Figure 6a. (b) The amplitude spectrum of the residual after inversion. (c) The amplitude spectrum of the residual after inversion using a smaller scale than in 11b. antoine1-sp2-wz08 [ER]

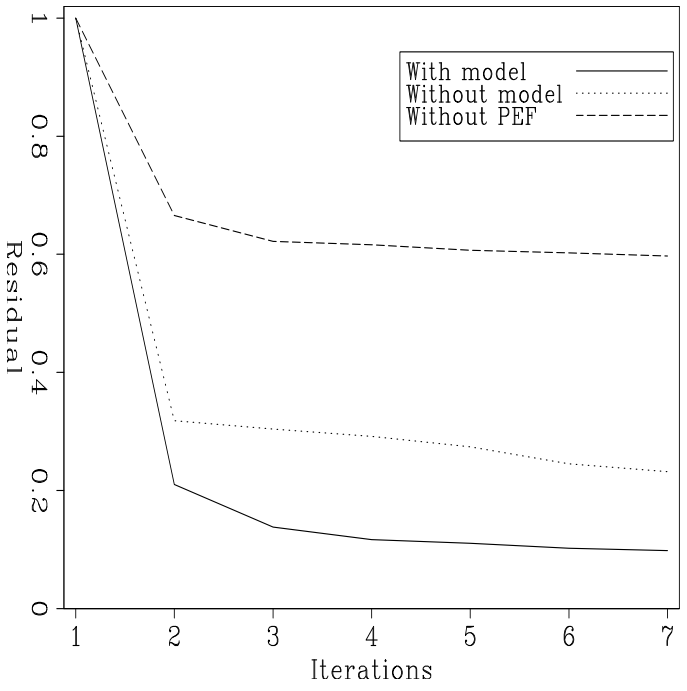

Figure 12: Convergence of the iterative schemes with or without a PEF for seven iterations. The PEF (continuous and dotted lines) allows the best convergence. antoine1-conv [ER]

and then subtract it from the input data. This method is based on the addition of a coherent noise modeling operator to the fitting goal (Nemeth, 1996) as follows:

$$\mathbf{0} \approx \mathbf{H}\mathbf{m_s} + \mathbf{A_n^{-1}}\mathbf{m_n} - \mathbf{d}, \tag{3}$$

where $\mathbf{H}$ is the signal modeling operator, $\mathbf{m_s}$ denotes the model for the signal, $\mathbf{A_n}$ is a PEF that models the coherent noise present in the data $\mathbf{d}$, and $\mathbf{m_n}$ denotes the model for the noise. This method is based on the central assumption that the data $\mathbf{d}$ result from the addition of noise $\mathbf{n}$ and signal $\mathbf{s}$. Thus, ideally, each operator $\mathbf{H}$ and $\mathbf{A_n}$ models a different part of the data. The PEF $\mathbf{A_n}$ and the operator $\mathbf{H}$ have different physical units. To balance the two operators, I use the fitting goal

$$\mathbf{0} \approx \mathbf{H}\mathbf{m_s} + \gamma\mathbf{A_n^{-1}}\mathbf{m_n} - \mathbf{d}, \tag{4}$$

where

$$\gamma = \frac{\|\mathbf{H}'\mathbf{d}\|_2}{\|\mathbf{A_n^{-1'}}\mathbf{d}\|_2}. \tag{5}$$

The constant γ is computed at the first iteration and kept constant. If I set $\mathbf{B} = \gamma\mathbf{A_n^{-1}}$, I can write the fitting goal in the following form:

$$\mathbf{0} \approx \begin{pmatrix} \mathbf{H} & \mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{m_s} \\ \mathbf{m_n} \end{pmatrix} - \mathbf{d}. \tag{6}$$

With $\mathbf{L} = (\mathbf{H}\ \ \mathbf{B})$ and $\mathbf{m}' = (\mathbf{m_s}\ \ \mathbf{m_n})$, the fitting goal becomes

$$\mathbf{0} \approx \mathbf{L}\mathbf{m} - \mathbf{d}, \tag{7}$$

which leads to the familiar normal equations

$$\mathbf{L}'\mathbf{L}\mathbf{m} = \mathbf{L}'\mathbf{d} \tag{8}$$

that can be written as

$$\begin{pmatrix} \mathbf{H}'\mathbf{H} & \mathbf{H}'\mathbf{B} \\ \mathbf{B}'\mathbf{H} & \mathbf{B}'\mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{m_s} \\ \mathbf{m_n} \end{pmatrix} = \begin{pmatrix} \mathbf{H}' \\ \mathbf{B}' \end{pmatrix} \mathbf{d}. \tag{9}$$

Based on the results derived in the appendix (equation (34)), the least-squares inverse of $\mathbf{m}$ is

$$\begin{pmatrix} \hat{\mathbf{m}}_\mathbf{s} \\ \hat{\mathbf{m}}_\mathbf{n} \end{pmatrix} = \begin{pmatrix} (\mathbf{H}'\overline{\mathbf{R_n}}\mathbf{H})^{-1}\mathbf{H}'\overline{\mathbf{R_n}} \\ (\mathbf{B}'\overline{\mathbf{R_s}}\mathbf{B})^{-1}\mathbf{B}'\overline{\mathbf{R_s}} \end{pmatrix} \mathbf{d}, \tag{10}$$

where $\overline{\mathbf{R_s}} = \mathbf{I} - \mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'$ and $\overline{\mathbf{R_n}} = \mathbf{I} - \mathbf{B}(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'$.
$\mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'$ is the signal resolution matrix and $\mathbf{B}(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'$ is the noise resolution matrix. With $\mathbf{d} = \mathbf{s} + \mathbf{n}$, we have

$$\overline{\mathbf{R_s}}\mathbf{d} \approx \overline{\mathbf{R_s}}\mathbf{n}, \tag{11}$$

$$\overline{\mathbf{R_n}}\mathbf{d} \approx \overline{\mathbf{R_n}}\mathbf{s}. \tag{12}$$

The operators $\overline{\mathbf{R_s}}$ and $\overline{\mathbf{R_n}}$ perform signal and noise filtering, respectively. These operators are also called projectors and discussed in detail by Guitton et al. (2001) in this report. We can see that equation (10) for the signal model $\mathbf{m_s}$ resembles equation (2), except for the filtering operator $\overline{\mathbf{R_n}}$. In other words, $\mathbf{A_n'A_n}$ in equation (2) plays the role of $\overline{\mathbf{R_n}}$ in equation (10). The correct separation of the signal **s** and noise **n** depends on the invertibility of the Hessians $\mathbf{H'}\overline{\mathbf{R_n}}\mathbf{H}$ and $\mathbf{B'}\overline{\mathbf{R_s}}\mathbf{B}$ in equation (10) (Nemeth, 1996).

The signal/noise separation is perfect if the signal and noise operators predict distinct parts of the data. The separation becomes more difficult if the two operators overlap. However, Nemeth (1996) has proven that the Hessians can be inverted if we introduce some regularization in the fitting goal. Hence, the noise prediction can proceed even if signal and noise are correlated. For instance, if we use a model space regularization (Fomel, 1997), we can consider

$$\mathbf{0} \approx \mathbf{Lm} - \mathbf{d}, \tag{13}$$

$$\mathbf{0} \approx \epsilon\mathbf{Cm}, \tag{14}$$

where

$$\mathbf{C} = \begin{pmatrix} \mathbf{C_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_n} \end{pmatrix} \tag{15}$$

is the covariance operator for the model **m**. It is important to keep in mind that ϵ does not have to be the same for the noise model covariance and the signal model covariance. As proven in the appendix (equation (39)), for ϵ constant, the least-squares solution of the regularized problem is

$$\begin{pmatrix} \hat{\mathbf{m}}_\mathbf{s} \\ \hat{\mathbf{m}}_\mathbf{n} \end{pmatrix} = \begin{pmatrix} (\mathbf{H'}\overline{\mathbf{R_n}}\mathbf{H} + \epsilon^2\mathbf{C_s'C_s})^{-1}\mathbf{H'}\overline{\mathbf{R_n}} \\ (\mathbf{B'}\overline{\mathbf{R_s}}\mathbf{B} + \epsilon^2\mathbf{C_n'C_n})^{-1}\mathbf{B'}\overline{\mathbf{R_s}} \end{pmatrix} \mathbf{d}. \tag{16}$$

The regularization operator can also include cross-terms accounting for the correlation between signal and noise, as follows:

$$\mathbf{C} = \begin{pmatrix} \mathbf{C_s} & \mathbf{C_{sn}} \\ \mathbf{C_{sn}} & \mathbf{C_n} \end{pmatrix}. \tag{17}$$

Although I have not explored this possibility, the regularization clearly offers the possibility to better separate noise and signal when the two components are correlated.

With respect to the filtering method, I concluded that a noise model is not mandatory as long as we can derive it iteratively using a two-stage process. Accordingly, I used the following algorithm to accomplish the separation with the subtraction method:

1. Solve $\mathbf{0} \approx \mathbf{Hm_s} - \mathbf{d}$.
2. Estimate a PEF $\mathbf{A_n}$ from the residual.
3. Restart a new inverse problem for the fitting goal in equation (4).

4. Stop when the residual has a white spectrum.

With the data in my study, I did not reestimate the PEF iteratively because I noticed that the first filter was accurate enough to predict the noise. However, for more complicated noisy events, an iterative scheme might be preferable. In the next section, I use the subtraction scheme to separate coherent noise from synthetic and real data. So far, no regularization is included in the inversion.

Subtracting the coherent noise in synthetic data

For this first series of tests, I used the synthetic data in Figure 1. The first stage, after which I estimate a PEF from the residual, took 45 iterations. The filter is one-dimensional with 30 coefficients (`a=30,1`). In the second stage, I use the inverse of the filter as the noise modeling operator shown in equation (4).

The subtraction results in a reduction of the noise similar to that with the filtering method. Figure 13 shows the result of the inversion. The residual (Figure 13d) is white, the model space is well resolved with all the curvatures present (Figure 13a), and the noise model (Figure 13c) clearly displays the linear coherent noise. The top and bottom of the residual (Figure 13d) were masked to avoid edge effects caused by the helical boundary conditions. Figure 13b shows the reconstructed data. This result compares favorably with that in Figure 3b. Finally, Figure 14a compares the amplitude spectrum of the input data (Figure 14a) and of the residual (Figure 14b). Obviously, the residual energy is very small.

Subtracting the coherent noise in real data

For purposes of comparison, I used the CMP gather of Figure 6 again. The filter size is as before (`a=30,1`). I iterated 45 times before estimating the PEF for the coherent noise. Then I iterated 20 times, for a total of 65 iterations.

The coherent noise attenuation again resembles that obtained with the filtering approach. In Figure 15, I show the result of the inversion. Figure 15a displays the model space, Figure 15b the reconstructed data from 15a, Figure 15c the noise model for the data ($\mathbf{A_n^{-1}m_n}$), and Figure 15d the residual. The noise model is mainly composed of the coherent noise I am trying to attenuate. The residual shows linear events scattered throughout the panel. I think this problem can be partially solved using nonstationary filters as opposed to a stationary one (Guitton et al., 2001). The importance of the linear events in the residual does not, however, negate the efficiency of this method. Because the amplitude spectrum of the residual in Figure 16b is whiter than the spectrum of the input data in Figure 16a, the fitting of the data was successful.

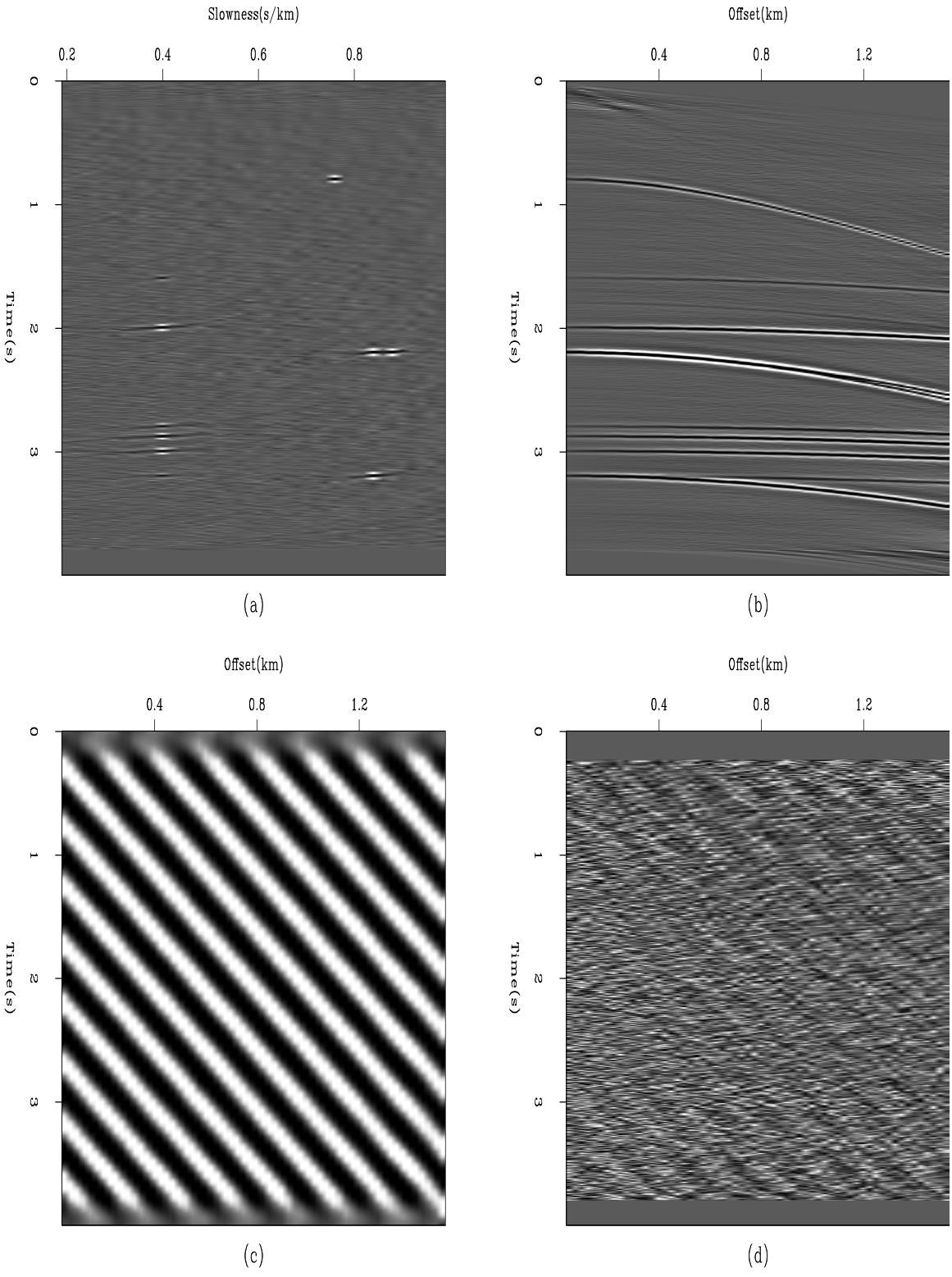

Figure 13: Subtracting the coherent noise in synthetic data. (a) An estimated model space. (b) The reconstructed data from the model space. (c) The estimated coherent noise $\mathbf{A_n^{-1}m_n}$. (d) The residual after inversion. Click *Movie* to see how the four panels evolve as the iterations continue. antoine1-compsy [ER,M]

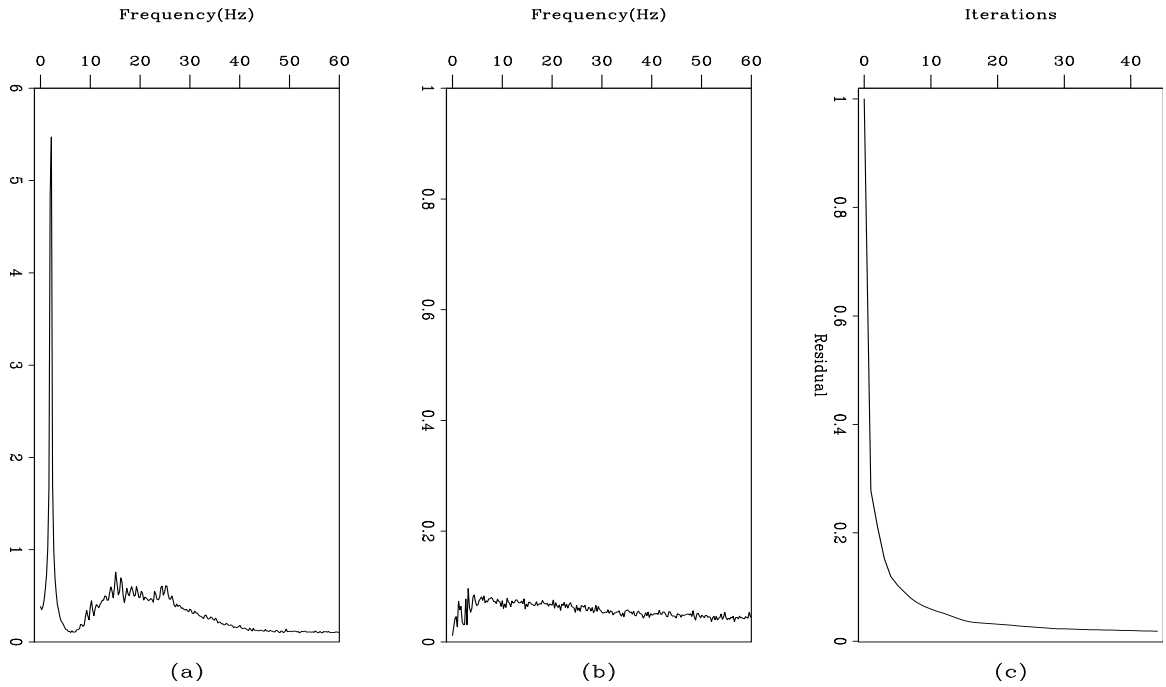

Figure 14: (a) The amplitude spectrum of the input data in Figure 1a. (b) The amplitude spectrum of the residual after inversion. (c) The normalized objective function. Click *Movie* to see how the two panels b and c evolve as the iterations continue. antoine1-compsf [ER,M]

Discussion of the subtraction method

The subtraction method is a compelling alternative to the filtering scheme. So far, the results of the two methods are comparable. However, this study shows that the correlation between the noise and signal can be incorporated in the inverse problem if we add a regularization term. I intend to investigate this possibility in more detail to see whether the subtraction method can yield more accurate noise filtering.

CONCLUSION

The signal/noise separation problem is addressed in two different ways. The first method intends to approximate the noise covariance matrix with prediction error filters. The second method introduces a noise modeling operator in the fitting goal. Although both methods separate the noise from the signal, the subtraction method can also mitigate the effects of the correlation between signal and noise by incorporating a regularization term into the fitting goal.

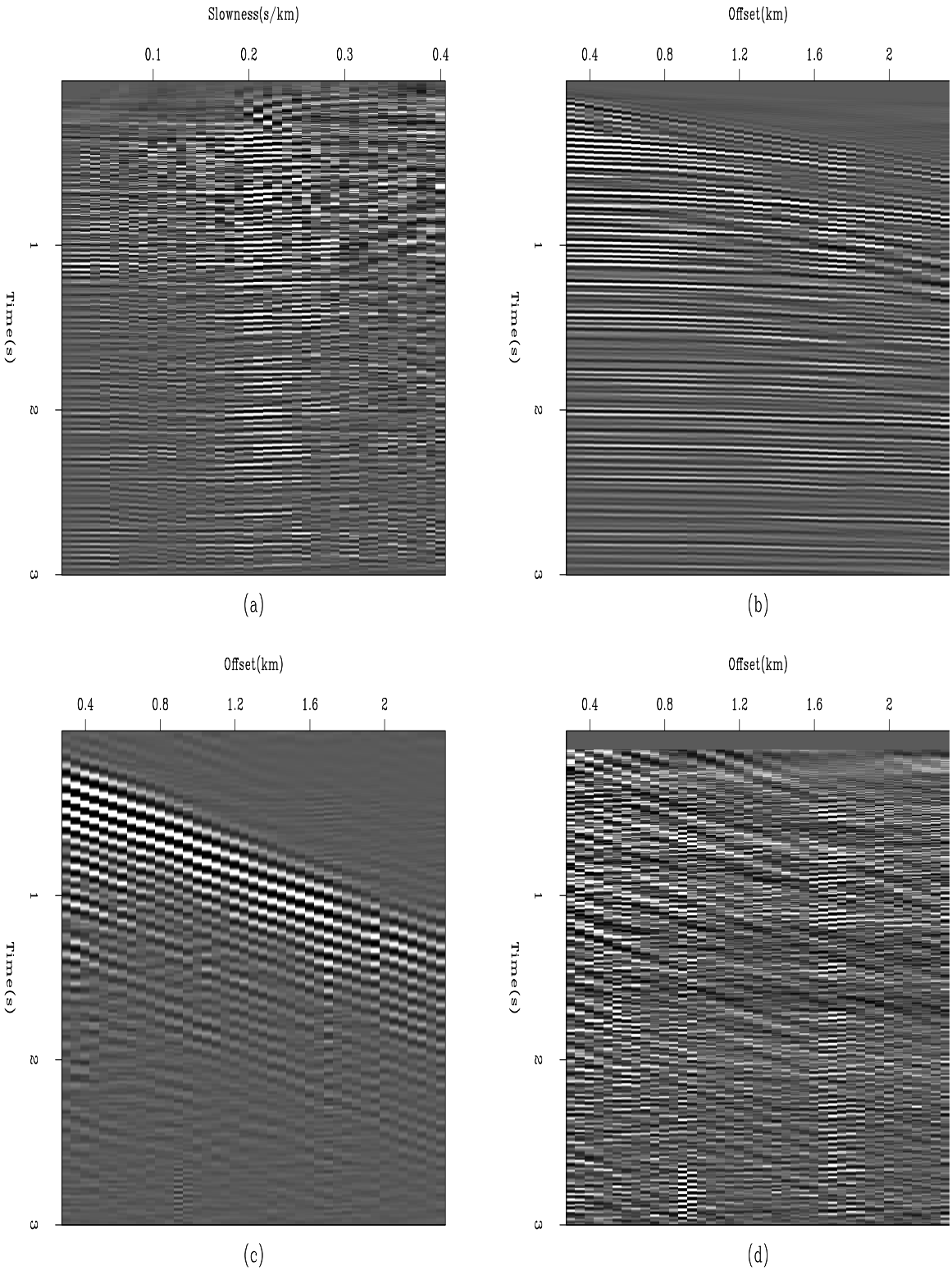

Figure 15: Subtracting the coherent noise in real data. (a) An estimated model space. (b) The reconstructed data from the model space. (c) The estimated coherent noise $\mathbf{A_n^{-1}m_n}$. (d) The residual after inversion. Click *Movie* to see how the four panels evolve as the iterations continue. antoine1-compwz [ER,M]

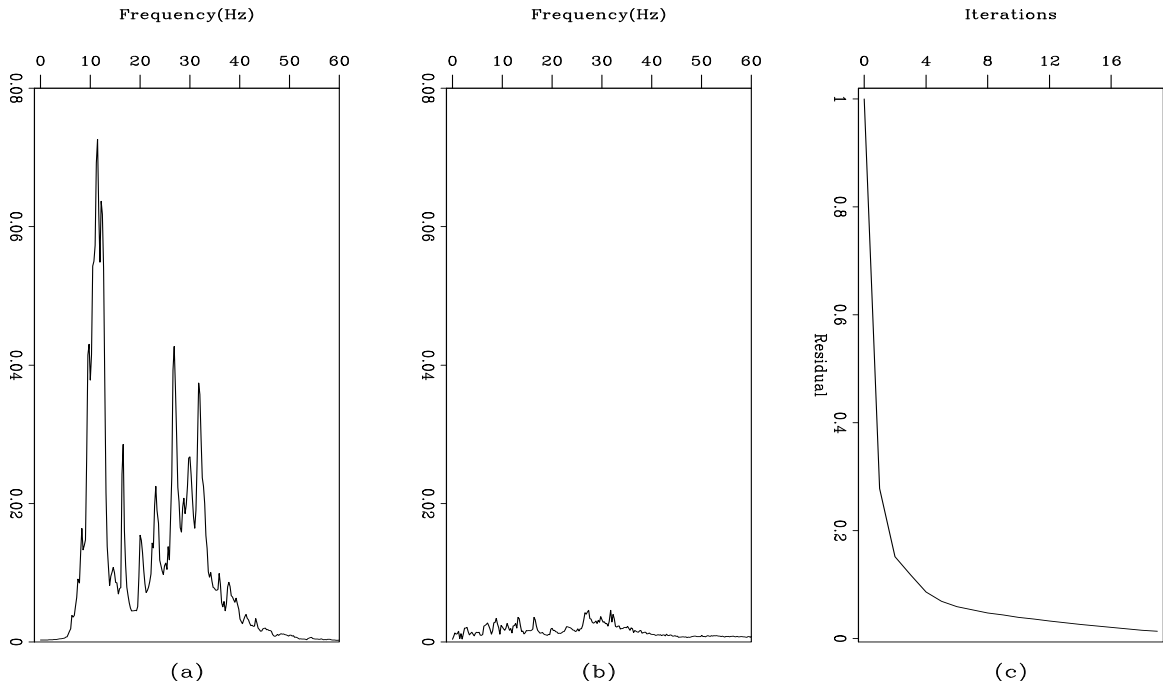

Figure 16: (a) The amplitude spectrum of the input data in Figure 6a. (b) The amplitude spectrum of the residual after inversion. (c) The normalized objective function. Click *Movie* to see how the two panels b and c evolve as the iterations continue. antoine1-compwf [ER,M]

REFERENCES

Claerbout, J. F., and Fomel, S., 1999, Geophysical Estimation by Example: Class notes, http://sepwww.stanford.edu/sep/prof/index.html.

Demmel, J. W., 1997, Applied Numerical Linear Algebra: SIAM.

Fomel, S., 1997, On model-space and data-space regularization: A tutorial: SEP–**94**, 141–164.

Guitton, A., Brown, M., Rickett, J., and Clapp, R., 2001, A pattern-based technique for ground-roll and multiple attenuation: SEP–**108**, 249–274.

Guitton, A., 2000, Coherent noise attenuation using Inverse Problems and Prediction Error Filters: SEP–**105**, 27–48.

Nemeth, T., 1996, Imaging and filtering by least-squares migration: Ph.D. thesis, University of Utah.

Tarantola, A., 1987, Inverse Problem Theory: Elsevier Science Publisher.

APPENDIX

In this section, I show how a 2×2 block matrix can be inverted and then I use the result to invert the Hessian in equation (9) with or without regularization.

Inversion of a 2×2 block matrix

Let us define the 2×2 block matrix $\mathbf{M}$ as follows:

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}, \tag{18}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are matrices. First, we consider the matrix equation

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{E} \\ \mathbf{F} \end{pmatrix} = \begin{pmatrix} \mathbf{G} \\ \mathbf{H} \end{pmatrix}. \tag{19}$$

If we multiply the top row by $-\mathbf{C}\mathbf{A}^{-1}$ and add it to the bottom, we have

$$(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})\mathbf{F} = \mathbf{H} - \mathbf{C}\mathbf{A}^{-1}\mathbf{G}. \tag{20}$$

Then we can easily find $\mathbf{F}$ and $\mathbf{E}$. The quantity $(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ is called the *Schur complement* of $\mathbf{A}$ and, denoted as $\mathbf{S_A}$, appears often in linear algebra (Demmel, 1997). The derivation of $\mathbf{F}$ and $\mathbf{E}$ can be written in a matrix form

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_A} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{21}$$

which resembles an *LDU* decomposition of $\mathbf{M}$. Alternatively, we have the *UDL* decomposition

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{S_D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{pmatrix}, \tag{22}$$

where $\mathbf{S_D} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ is the Schur complement of $\mathbf{D}$. The inversion formulas are then easy to derive as follows:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_A^{-1}} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \tag{23}$$

and

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{S_D^{-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}. \tag{24}$$

The decomposition of the matrix $\mathbf{M}$ offers opportunities for fast inversion algorithms. The final expressions for $\mathbf{M}$ are

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} [\mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{S_A^{-1}}\mathbf{C}\mathbf{A}^{-1}] & -\mathbf{A}^{-1}\mathbf{B}\mathbf{S_A^{-1}} \\ -\mathbf{S_A^{-1}}\mathbf{C}\mathbf{A}^{-1} & \mathbf{S_A^{-1}} \end{pmatrix} \tag{25}$$

and

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{S_D^{-1}} & -\mathbf{S_D^{-1}BD^{-1}} \\ -\mathbf{D^{-1}CS_D^{-1}} & [\mathbf{D^{-1}} + \mathbf{D^{-1}CS_D^{-1}BD^{-1}}] \end{pmatrix}. \tag{26}$$

Equations (25) and (26) yield the matrix inversion lemma

$$(\mathbf{A} - \mathbf{BD^{-1}C})^{-1} = \mathbf{A^{-1}} + \mathbf{A^{-1}B}(\mathbf{D} - \mathbf{CA^{-1}B})^{-1}\mathbf{CA^{-1}}. \tag{27}$$

Inversion of the Hessian

Using the preceding results, I can invert for the Hessian in equation (9), either with or without regularization.

- **Without regularization**

The fitting goal is

$$\mathbf{0} \approx \mathbf{Lm} - \mathbf{d}, \tag{28}$$

with $\mathbf{L} = (\mathbf{H}\ \ \mathbf{B})$ and $\mathbf{m}' = (\mathbf{m_s}\ \ \mathbf{m_n})$. The matrix equation we want to solve is

$$\begin{pmatrix} \mathbf{H'H} & \mathbf{H'B} \\ \mathbf{B'H} & \mathbf{B'B} \end{pmatrix} \begin{pmatrix} \mathbf{m_s} \\ \mathbf{m_n} \end{pmatrix} = \begin{pmatrix} \mathbf{H'd} \\ \mathbf{B'd} \end{pmatrix}, \tag{29}$$

where $\mathbf{m_s}$ and $\mathbf{m_n}$ are the unknowns. For $\mathbf{m_s}$, I use the bottom row of equation (25). For $\mathbf{m_n}$, I use the top row of equation (26). We have, then,

$$\begin{aligned} \hat{\mathbf{m}}_\mathbf{s} &= (\mathbf{H'H} - \mathbf{H'B}(\mathbf{B'B})^{-1}\mathbf{B'H})^{-1}\mathbf{H'd} - \\ &\quad (\mathbf{H'H} - \mathbf{H'B}(\mathbf{B'B})^{-1}\mathbf{B'H})^{-1}\mathbf{H'B}(\mathbf{B'B})^{-1}\mathbf{B'd}, \end{aligned} \tag{30}$$

$$\begin{aligned} \hat{\mathbf{m}}_\mathbf{n} &= (\mathbf{B'B} - \mathbf{B'H}(\mathbf{H'H})^{-1}\mathbf{H'B})^{-1}\mathbf{B'd} - \\ &\quad (\mathbf{B'B} - \mathbf{B'H}(\mathbf{H'H})^{-1}\mathbf{H'B})^{-1}\mathbf{B'H}(\mathbf{H'H})^{-1}\mathbf{H'd}, \end{aligned} \tag{31}$$

which can be simplified as follows:

$$\hat{\mathbf{m}}_\mathbf{s} = (\mathbf{H'}(\mathbf{I} - \mathbf{B}(\mathbf{B'B})^{-1}\mathbf{B'})\mathbf{H})^{-1}\mathbf{H'}(\mathbf{I} - \mathbf{B}(\mathbf{B'B})^{-1}\mathbf{B'})\mathbf{d}, \tag{32}$$

$$\hat{\mathbf{m}}_\mathbf{n} = (\mathbf{B'}(\mathbf{I} - \mathbf{H}(\mathbf{H'H})^{-1}\mathbf{H'})\mathbf{B})^{-1}\mathbf{B'}(\mathbf{I} - \mathbf{H}(\mathbf{H'H})^{-1}\mathbf{H'})\mathbf{d}. \tag{33}$$

$\mathbf{B}(\mathbf{B'B})^{-1}\mathbf{B'}$ is the coherent noise resolution matrix, whereas $\mathbf{H}(\mathbf{H'H})^{-1}\mathbf{H'}$ is the signal resolution matrix. Denoting $\overline{\mathbf{R_s}} = \mathbf{I} - \mathbf{H}(\mathbf{H'H})^{-1}\mathbf{H'}$ and $\overline{\mathbf{R_n}} = \mathbf{I} - \mathbf{B}(\mathbf{B'B})^{-1}\mathbf{B'}$ yields the following simplified expression for $\hat{\mathbf{m}}_\mathbf{s}$ and $\hat{\mathbf{m}}_\mathbf{n}$:

$$\begin{pmatrix} \hat{\mathbf{m}}_\mathbf{s} \\ \hat{\mathbf{m}}_\mathbf{n} \end{pmatrix} = \begin{pmatrix} (\mathbf{H'}\overline{\mathbf{R_n}}\mathbf{H})^{-1}\mathbf{H'}\overline{\mathbf{R_n}} \\ (\mathbf{B'}\overline{\mathbf{R_s}}\mathbf{B})^{-1}\mathbf{B'}\overline{\mathbf{R_s}} \end{pmatrix} \mathbf{d}. \tag{34}$$

- **With model space regularization**

The fitting goal becomes

$$\mathbf{0} \approx \mathbf{L}\mathbf{m} - \mathbf{d}, \tag{35}$$
$$\mathbf{0} \approx \epsilon\mathbf{C}\mathbf{m}, \tag{36}$$

with $\mathbf{L} = (\mathbf{H}\ \ \mathbf{B})$, $\mathbf{m}' = (\mathbf{m_s}\ \ \mathbf{m_n})$ and

$$\mathbf{C} = \begin{pmatrix} \mathbf{C_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_n} \end{pmatrix}. \tag{37}$$

The matrix equation we want to solve is

$$\begin{pmatrix} \mathbf{H}'\mathbf{H} + \epsilon^2\mathbf{C}_\mathbf{s}'\mathbf{C_s} & \mathbf{H}'\mathbf{B} \\ \mathbf{B}'\mathbf{H} & \mathbf{B}'\mathbf{B} + \epsilon^2\mathbf{C}_\mathbf{n}'\mathbf{C_n} \end{pmatrix} \begin{pmatrix} \mathbf{m_s} \\ \mathbf{m_n} \end{pmatrix} = \begin{pmatrix} \mathbf{H}'\mathbf{d} \\ \mathbf{B}'\mathbf{d} \end{pmatrix}. \tag{38}$$

Using equations (25) and (26) we obtain

$$\begin{pmatrix} \hat{\mathbf{m}}_\mathbf{s} \\ \hat{\mathbf{m}}_\mathbf{n} \end{pmatrix} = \begin{pmatrix} (\mathbf{H}'\overline{\mathbf{R_n}}\mathbf{H} + \epsilon^2\mathbf{C}_\mathbf{s}'\mathbf{C_s})^{-1}\mathbf{H}'\overline{\mathbf{R_n}} \\ (\mathbf{B}'\overline{\mathbf{R_s}}\mathbf{B} + \epsilon^2\mathbf{C}_\mathbf{n}'\mathbf{C_n})^{-1}\mathbf{B}'\overline{\mathbf{R_s}} \end{pmatrix} \mathbf{d}, \tag{39}$$

where $\overline{\mathbf{R_s}} = \mathbf{I} - \mathbf{H}(\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'$ and $\overline{\mathbf{R_n}} = \mathbf{I} - \mathbf{B}(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'$.