Chapter 7

Radon Transforms

In this chapter we deal with the numerical implementation of the Radon transform. We will analyze the problem using the inverse problem formalism and study the problem of designing a high resolution Parabolic Radon transform for multiple attenuation.

7.1 Slant Stacks

Different techniques have been devised to identify and/or filter linear events. Generally, they have the following common framework. First, they assume that a set of linear events are recorded on an array with discrete and limited coverage. Secondly, they assume that the noise is uncorrelated with the signals. In geophysics, linear event identification has been an active field of research. Two classic examples are vertical seismic profiles (VSP) and slowness vector estimation in seismographic arrays for earthquake detection and location. In VSP processing, linear event detection-estimation is used to identify and separate the principal components of the VSP data: the up-going and the down-going waves.

A general strategy for event identification-estimation involves the following approach. First, the data are transformed to a new domain where each component may be isolated. Then, after masking the undesired components, the data are mapped back to the original domain retaining only the desired information.

In seismic processing, the Radon transform is commonly known as the $\tau - p$ (τ denotes time and p ray parameter) or slant stack transform. The original idea developed by Radon in 1917 (Deans, 1983), has provided a basic framework for many problems of image reconstruction in physics, astronomy, medicine, optics, non-destructive testing, and geophysics. In image processing, it is also called the Hough transform (Pratt, 1991),

which may be regarded as a transformation of a line in Cartesian coordinate spac e to a point in polar coordinate space.

In geophysics, the properties of the Radon transform were examined by Phinney et al. (1981), Durrani and Bisset (1984) and Tatham (1984). Chapman (1981) developed exact formulas for a point source in Cartesian or spherical coordinates, and for a line source in cylindrical coordinates. The relationship between the Radon transform and the plane wave decomposition is also well established (Stoffa et al., 1981; Treitel et al., 1982). Least squares procedures to compute the Radon transform were investigated by Thorson and Claerbout (1985), Beylkin (1987) and Kostov (1990). These authors showed how to mitigate the smearing caused by the finite aperture. Recently, Zhou and Greenhalgh (1994) linked the least squares solution to p-dependent Wiener filters. These researchers derived the slant stack formulas in the continuous domain, but the resulting algorithms are identical to those obtained by other researchers (Beylkin, 1987; Kostov, 1990).

In order to avoid the inversion of prohibitively large matrices the problem may be posed in the frequency-space domain ($f - x$). This technique was adopted by Beylkin (1987), Kostov (1990), Foster and Mosher (1992), and recently by Zhou and Greenhalgh (1994). This allows us to solve several small problems in the band that comprises the signal. Some stability concerns arise when the problem is tackled in this manner. Particularly, a least squares solution can be extremely unstable at low frequencies. In addition, it is interesting that slant stacks can be also computed in the time-space domain. Thorson and Claerbout (1985) and, recently Yilmaz and Tanner (1994), have presented high resolution least squares slant stack operators designed in time-space domain. Their procedures use an iterative inversion scheme especially devised to solve large linear sparse operators. Thorson and Claerbout (1985) have also shown how to update in each iteration the variances of the model to drive the solution to minimum entropy. Yilmaz and Taner (1994) have also developed an interesting scheme based on fuzzy logic to mitigate the alias.

7.1.1 The slant stack operator (conventional definition)

Let $u(h,t)$ represent a seismic signal. Throughout this chapter the variable t designates the time and h the offset or range. For a continuous array we define the slant stack by means of the following transformation

$$v(p,\tau) = (\mathcal{L}u)(p,\tau) = \int_{-\infty}^{\infty} u(h, t = \tau + h\,p)dh\,. \tag{7.1}$$

Where p and τ denote the slope or ray parameter and the intercept time, respectively.

$v(p,\tau)$ is used to designate the signal in the $\tau - p$ domain. The adjoint transform $\mathcal{L}^*$ is given by

$$\tilde{u}(h,t) = (\mathcal{L}^* v)(p,\tau) = \int_{-\infty}^{\infty} v(p, t = \tau - hp) dp\,. \tag{7.2}$$

In the frequency domain, the pair of transformations are given by,

$$V(p,\omega) = \int_{-\infty}^{\infty} U(h,\omega) e^{i\omega ph} dh, \tag{7.3}$$

$$\tilde{U}(h,\omega) = \int_{-\infty}^{\infty} V(p,\omega) e^{-i\omega ph} dp, \tag{7.4}$$

substituting, (2.3) into (2.4) yields

$$\tilde{U}(h,\omega) = \int_{-\infty}^{\infty} U(h',\omega) \int_{-\infty}^{\infty} e^{-i\omega p(h-h')} dp\, dh' \tag{7.5}$$

which may be written as follows

$$\tilde{U}(h,\omega) = U(h,\omega) * \rho(h,\omega)\,, \tag{7.6}$$

where $*$ denotes convolution and the function ρ is given by

$$\rho(h,\omega) = \int_{-\infty}^{\infty} e^{-i\omega ph} dp\,, \tag{7.7}$$

making the substitution $z = -\omega p$ equation (7.7) becomes

$$\rho(h,\omega) = \int_{-\infty}^{\infty} \frac{1}{|\omega|} e^{ihz} dz = \frac{2\pi}{|\omega|} \delta(h)\,. \tag{7.8}$$

The convolution operator is a delta function with respect to the variable h. Using the property of the δ function,

$$\begin{aligned} \tilde{U}(h,\omega) &= \tfrac{2\pi}{|\omega|} U(h,\omega) * \delta(h) \\ &= \tfrac{2\pi}{|\omega|} U(h,\omega) \end{aligned} \tag{7.9}$$

the inversion formula becomes,

$$U(h,\omega) = \frac{|\omega|}{2\pi} \tilde{U}(h,\omega)\,. \tag{7.10}$$

The inverse is computed in two steps. First, the adjoint is used to evaluate $\tilde{U}(h,\omega)$. Then, $\tilde{U}(h,\omega)$ is multiplied by the frequency response of the ρ filter. The conventional slant stack pair in the frequency domain results in,

$$\begin{aligned} V(p,\omega) &= \int_{-\infty}^{\infty} U(h,\omega) e^{i\omega p h} dh, \\ U(h,\omega) &= \frac{|\omega|}{2\pi} \int_{-\infty}^{\infty} V(p,\omega) e^{-i\omega p h} dp\,. \end{aligned} \tag{7.11}$$

Now, consider that the range of p is a finite interval, $p \in [-P, P]$. This case leads to the following ρ filter,

$$\rho(h,\omega) = \int_{-P}^{P} e^{-i\omega p h} dp = 2P \frac{\sin(\omega P h)}{\omega P h}\,. \tag{7.12}$$

Substituting (7.12) in (7.6),

$$\tilde{U}(h,w) = 2P \int_{-\infty}^{\infty} U(h',\omega) \frac{\sin(\omega P(h-h'))}{\omega P(h-h')} dh'. \tag{7.13}$$

It is evident that the data may be recovered after solving a deconvolution problem. Spatial deconvolution is required since the infinite range of the variable p is truncated to a finite range. The wavenumber response of the ρ filter has the following expression:

$$\begin{aligned} \rho(k,\omega) &= \int_{-\infty}^{\infty} \rho(h,\omega) e^{ikh} dh \\ &= \int_{-\infty}^{\infty} \int_{-P}^{P} e^{-i(\omega p - k)h} dh dp \\ &= \int_{-P}^{P} \delta(\omega p - k) dp \end{aligned}$$

$$= \frac{1}{\omega}\int_{-\omega P}^{\omega P}\delta(k'-k)dk' \tag{7.14}$$
$$= \begin{cases} \frac{1}{\omega} & k \le |\omega P| \\ 0 \quad , & \text{otherwise} \end{cases}$$

According to the last equation, the spatial deconvolution will be unstable if the wavenumbers in the data lie outside the range $[-\omega P, \omega P]$. Equation (2.14) also shows that the deconvolution is unstable at low frequencies.

7.1.2 The inverse slant stack operator

The definition of the forward slant stack operator and its adjoint may be changed to construct another slant stack pair,

$$u(h,t) = (\mathcal{L}^* v)(p,\tau) = \int_{-\infty}^{\infty} v(p, t = \tau - hp)dp \tag{7.15}$$

$$\tilde{v}(p,\tau) = (\mathcal{L}u)(h,t) = \int_{-\infty}^{\infty} u(h, t = \tau + hp)dh, \tag{7.16}$$

the pair of transformations can be posed in the frequency-offset domain,

$$U(h,\omega) = \int_{-\infty}^{\infty} V(p,\omega)e^{-i\omega ph}dp, \tag{7.17}$$

$$\tilde{V}(p,\omega) = \int_{-\infty}^{\infty} U(h,\omega)e^{i\omega ph}dh\,. \tag{7.18}$$

Substituting, (7.17) into (7.18) yields,

$$\tilde{V}(p,\omega) = \int_{-\infty}^{\infty} V(p',\omega)\int_{-\infty}^{\infty} e^{-i\omega h(p-p')}dhdp'\,, \tag{7.19}$$

where now, the convolution is with respect to the variable p, and the convolutional operator is given by

$$\gamma(p,\omega) = \int_{-\infty}^{\infty} \frac{1}{|\omega|} e^{ihp} dh = \frac{2\pi}{|\omega|}\delta(p)\,. \tag{7.20}$$

The γ filter is a delta function with respect to the variable p. Therefore, equation (7.19) becomes,

$$\tilde{V}(p,\omega) = \frac{2\pi}{|\omega|} V(p,\omega) \tag{7.21}$$

or equivalently

$$V(p,\omega) = \frac{|\omega|}{2\pi}\tilde{V}(p,\omega)\,. \tag{7.22}$$

From the above derivation, it is clear that the ρ and the γ filters have the same frequency response. Finally, the slant stack pair becomes,

$$\begin{aligned} V(p,\omega) &= \frac{|\omega|}{2\pi}\int_{-\infty}^{\infty} U(h,\omega)e^{i\omega ph}dh, \\ U(h,\omega) &= \int_{-\infty}^{\infty} V(p,\omega)e^{-i\omega p}dp\,. \end{aligned} \tag{7.23}$$

Assuming that $h \in [-H, H]$ (finite aperture), the γ filter has the following structure

$$\gamma(p,\omega) = \int_{-H}^{H} e^{i\omega ph}dh = 2H\frac{\sin(\omega Hp)}{\omega Hp}\,. \tag{7.24}$$

Hence, $V(p,\omega)$ may be calculated by solving the following integral equation,

$$\tilde{V}(p,\omega) = 2H\int_{-\infty}^{\infty} V(p',\omega)\frac{\sin(\omega H(p-p'))}{\omega H(p-p')}dp'\,. \tag{7.25}$$

After a comparison of the slant stacks pairs, equations (7.11) and (7.23), it is clear that a deconvolution procedure is required in both cases. In the conventional slant stack transform, the deconvolution is necessary to recover the data from the $\tau - p$ space. In the inverse slant stack operator the deconvolution process is required to estimate the $\tau - p$ space. The truncation effect of the variable p may be alleviated by choosing the proper

region of support of the transform. The truncation of the variable h is associated with the resolution of the transform and cannot be alleviated by simple means. Generally, both the variables h and p are truncated. Thus, deconvolution should be carried out in both the forward and inverse transform (Zhou and Greenhalgh, 1994). However, the range of p may be chosen in such a way that most of the energy in the signal lies within this range.

7.1.3 The sampling theorem for slant stacks

Assuming that the wavefield is evenly sampled according to $U(n\Delta h,\omega), n = 0,\pm1,\pm2,\ldots$, the relationship between the $\tau - p$ and the $h - t$ spaces is given by

$$U(n\Delta h,\omega) = \frac{|\omega|}{2\pi}\int_{-\infty}^{\infty} V(p,\omega)e^{-i\omega pn\Delta h}dp\,, \tag{7.26}$$

where $V(p,\omega)$ denotes the slant stack corresponding to a continuous wavefield $U(h,\omega)$. The integration domain can be decomposed into small subdomains as follows,

$$\begin{aligned} U(n\Delta h,\omega) &= \frac{|\omega|}{2\pi}\sum_{k=-\infty}^{\infty}\int_{-(2k-1)\frac{\pi}{\omega\Delta h}}^{(2k+1)\frac{\pi}{\omega\Delta h}} V(p,\omega)e^{-i\omega pn\Delta h}dp \\ &= \frac{|\omega|}{2\pi}\sum_{k=-\infty}^{\infty}\int_{-\frac{\pi}{\omega\Delta h}}^{\frac{\pi}{\omega\Delta h}} V(p+2k\frac{\pi}{\omega\Delta h},\omega)e^{-i\omega(p+2k\frac{\pi}{\omega\Delta h})n\Delta h}dp \end{aligned} \tag{7.27}$$

since $e^{-i2\pi nk} = 1\ \forall n\ k$, the last equation may be written in the following form

$$U(n\Delta h,\omega) = \frac{|\omega|}{2\pi}\int_{-\frac{\pi}{\omega\Delta h}}^{\frac{\pi}{\omega\Delta h}} V_d(p,\omega)e^{-i\omega pn\Delta h}dp, \tag{7.28}$$

where the relationship between the slant stack of the continuous signal and the one corresponding to the sampled wavefield, $V_d(p,\omega)$, is given by

$$V_d(p,\omega) = \sum_{k=-\infty}^{\infty} V(p+2k\frac{\pi}{\omega\Delta h},\omega). \tag{7.29}$$

Thus, the discrete signal has an $\omega - p$ representation with support in the range $p \in [-\frac{\pi}{\omega\Delta h},\frac{\pi}{\omega\Delta h}]$. The components with slope $p - 2\frac{\pi}{\omega\Delta h}, p + 2\frac{\pi}{\omega\Delta h}, p - 4\frac{\pi}{\omega\Delta h}, p + 4\frac{\pi}{\omega\Delta h},\ldots$ will

appear to have slope p and every slope outside the range $(-\frac{\pi}{\omega\Delta h}, \frac{\pi}{\omega\Delta h})$ will have an alias inside this range. If the continuous signal has all the components inside that range, the aliased components do not exist and therefore we can write $V_d(p,\omega) = V(p,\omega)$. It is clear from the above discussion that spatial sampling must be chosen so as to avoid the aliasing effect. If $P = P_{max} = -P_{min}$, the following relationship guarantees the absence of alias,

$$\Delta h \leq \frac{1}{2P f_{max}}, \tag{7.30}$$

where $f_{max} = \omega_{max}/2\pi$ is the maximum temporal frequency of the seismic signal. The product $P\, f_{max}$ is also the maximum wavenumber. Similarly, if Δh is given, the maximum ray parameter that can be retrieved ed without alias is given by

$$P_{max} = \frac{1}{2\Delta h f_{max}}. \tag{7.31}$$

For a non-symmetric slant stack, $P_{max} \neq -P_{min}$, equation (7.30) is modified as follows (Turner, 1990),

$$\Delta h \leq \frac{1}{P' f_{max}}, \tag{7.32}$$

where $P' = |P_{max} - P_{min}|$.

7.2 Discrete slant stacks

Discrete versions of equations the continuous Radon pair are obtained by replacing integrals by summations and imposing finite limits. First, assume that the seismogram contains $N = L_f - L_n$ traces, where the indices L_f and L_n denote far and near offset traces respectively.

$$v(p,\tau) = (\mathcal{L}u)(p,\tau) = \sum_{l=L_n}^{L_f} u(h_l, \tau + h_l p)\Delta h_l, \tag{7.33}$$

where $\Delta h_l = (h_{l+1} - h_l)$ for $l = L_n, \ldots, L_f - 1$. Similarly, we approximate the continuous Radon transform by the following expression

$$\tilde{u}(h,t) = (\mathcal{L}^* v)(\tau, p) = \sum_{j=J_{min}}^{J_{max}} v(h, t - hp)\Delta p_j \tag{7.34}$$

where $\Delta p_j = (p_{j+1} - p_j)$ for $j = J_{min}, \ldots, J_{max} - 1$. Taking the Fourier transform of the above equations yields

$$V(p,f) = \sum_{l=L_n}^{L_f} U(h_l, f) e^{2\pi i f h_l p} \Delta h_l \tag{7.35}$$

$$\tilde{U}(h,f) = \sum_{j=J_{min}}^{J_{max}} V(p,f) e^{-2\pi i f h p_j} \Delta p_j \,. \tag{7.36}$$

Using matrix notation it is possible to rewrite the slant stack and its adjoint as follows (f is omitted to avoid notational clutter),

$$\mathbf{m} = \mathbf{L}^H \mathbf{d} \tag{7.37}$$

$$\tilde{\mathbf{d}} = \mathbf{L}\mathbf{m} \tag{7.38}$$

The operators $\mathbf{L}$ and $\mathbf{L^H}$ form an adjoint pair. The matrix $\mathbf{L}$ is the forward operator and $\mathbf{L}^*$ denotes the adjoint operator. The vector $\mathbf{m}$ indicates the Radon space $V(p,f)$ at discrete values of p and fix frequency f, whereas the vector $\mathbf{d}$ indicates the data $U(h,f)$ at discrete values of h and fix frequency f.

7.2.1 The discrete slant stack operator (conventional definition)

The slant stack operator, equation (7.37), maps the $t - x$ space into the $\tau - p$ domain; the adjoint, equation (7.38), maps the $\tau - p$ domain into the $t - x$ domain. It is clear that since $\mathbf{L}$ is non-orthogonal $\mathbf{L}$ and $\mathbf{L}^H$ do not constitute an inverse pair. Given $\mathbf{m} = \mathbf{L}\mathbf{d}$, the problem is how to recover $\mathbf{d}$. A relationship between $\mathbf{d}$ and $\tilde{\mathbf{d}}$ is obtained after substituting (7.37) into (7.38)

$$\tilde{\mathbf{d}} = \mathbf{L}\,\mathbf{L^H}\,\mathbf{d}\,. \tag{7.39}$$

Equation (2.41) is uniquely invertible in $f \in B$ provided that $\det(\mathbf{L}\,\mathbf{L}^{\mathbf{H}}) \neq \mathbf{0}$ in the band B,

$$\begin{aligned} \mathbf{d} = & \ (\mathbf{L}\,\mathbf{L}^{H})^{-1}\tilde{\mathbf{d}} \\ = & \ \mathbf{G}^{-1}\tilde{\mathbf{d}}\,. \end{aligned} \tag{7.40}$$

The $N \times N$ matrix $\mathbf{G} = \mathbf{L}\,\mathbf{L}^{H}$ represents a discrete version of the ρ filter. The pair of transformations which map a signal from $f-h$ to $f-p$ and vice-versa is given by

$$\begin{aligned} \mathbf{m} = & \ \mathbf{L}\mathbf{d} \\ \mathbf{d} = & \ \mathbf{G}^{-1}\mathbf{L}\mathbf{m}\,. \end{aligned} \tag{7.41}$$

The vector $\mathbf{m}$ always exists since it is obtained by means of a simple mapping. Both expressions constitute an inverse pair when the inverse of $\mathbf{G}$ exits. The forward and inverse pair does not permit to adequately model the signal when additive noise is present. If the data are contaminated with noise, the noise is mapped to the Radon domain.

7.2.2 The least squares solution

Assume that the data is the result of applying a Radon operator (slant stack) to a $\mathbf{m}$.

$$\mathbf{d} = \mathbf{L}\mathbf{m} \tag{7.42}$$

The idea is to find $\mathbf{m}$ such that the following objective function is minimized (Yilmaz, 1994),

$$J = ||\mathbf{d} - \mathbf{L}\mathbf{m}||^2 \tag{7.43}$$

The solution to this problem is the least squares solution

$$\mathbf{m} = (\mathbf{L}^{H}\mathbf{L})^{-1}\mathbf{L}^{H}\mathbf{d} \tag{7.44}$$

In general the inverse needs to be stabilized using a damping parameter.

$$\mathbf{m} = (\mathbf{L}^H\mathbf{L} + \mu\mathbf{I})^{-1}\mathbf{L}^H\mathbf{d} \tag{7.45}$$

In general, this is the approach that it is used to compute slant stacks and parabolic Radon transfomrm. Other techniques to improve the resoltion of these operators where proposed by Sacchi and Ulrych (1995).

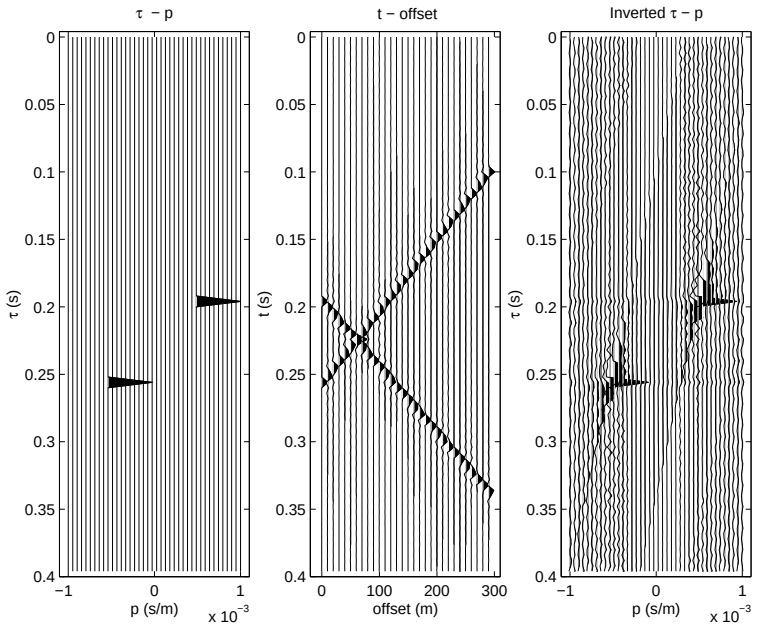

Figure 7.1: Left: Ideal $\tau - p$ panel. Center: Data generated by forward transforming the ideal $\tau - p$ panel. Right: Inverted $\tau - p$ panel.

7.2.3 Example

In Figure (7.2) we display 3 panels. The first panel is the ideal $\tau - p$ signal; the second panel is the $\tau - p$ signal transformed to $offset - time$. These are two linear events with positive and negative slope. The third panel (left) is the inverted $\tau - p$ signal using least-squares. It is clear that artifacts have been created in the inverted $\tau - p$. These artifacts are generated by alias. As we have already seen f_{max}, the maximum ray parameter and the spatial sampling must satisfied a Nyquist condition (equation (7.30)). This condition is not satisfied and, therefore, the $\tau - p$ domain exhibits alias.

In Figure (7.2) I muted the the $\tau - p$ domain eliminating all the contribution where $p > 0$. The muted $\tau - p$ domain is used to reconstruct the data, this is displayed in the Figure (**??**) [Left]. This procedure can be used to discriminate down-going and up-going wavefields in Vertical Seismic Profiles (VSP).

Figures (7.3) and (7.4) displayed a simulation similar to the one described above but now the original signal in $t - offset$ has spectral components that are contained in the $5 - 35$Hz band. In other words I have eliminated the alias artifacts.

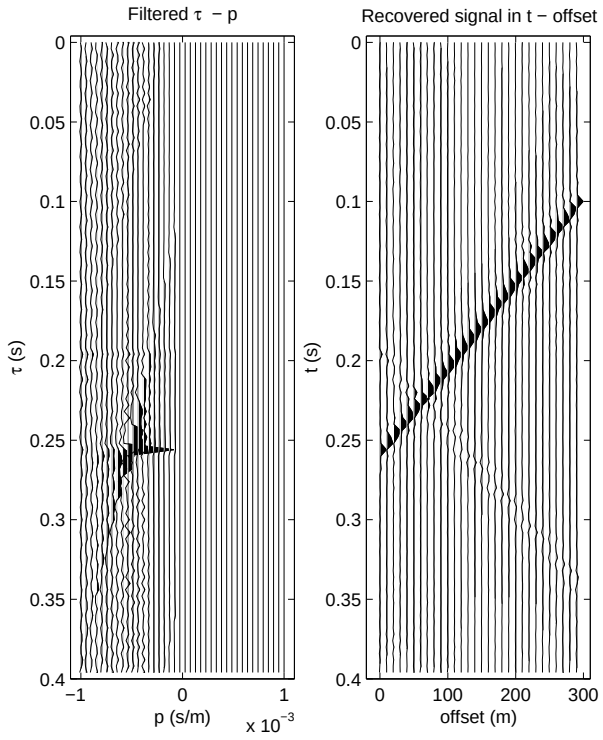

Figure 7.2: Left: Inverted $\tau - p$ panel after muting. Right: Data reconstructed by forward modeling the inverted/muted $\tau - p$ panel.

7.3 Parabolic Radon Transform (Hampson, 1986)

This is a simple modificarion to the slant stack, instead of integrating along curves of the form

$$t = \tau + ph$$

we use curves of the type

$$t = \tau + qh^2$$

.

This is a good approximation to process data containing hyperbolic events after NMO correction Parabolic Radon Transform are utilized to removed multiple reflections. After NMO correction the moveout of the primaries is zero. The residual move out of the multiples follows in a first order approximation a parabolic moveout. The transform is used to isolated multiples from primaries in order to mute (filter) them out.

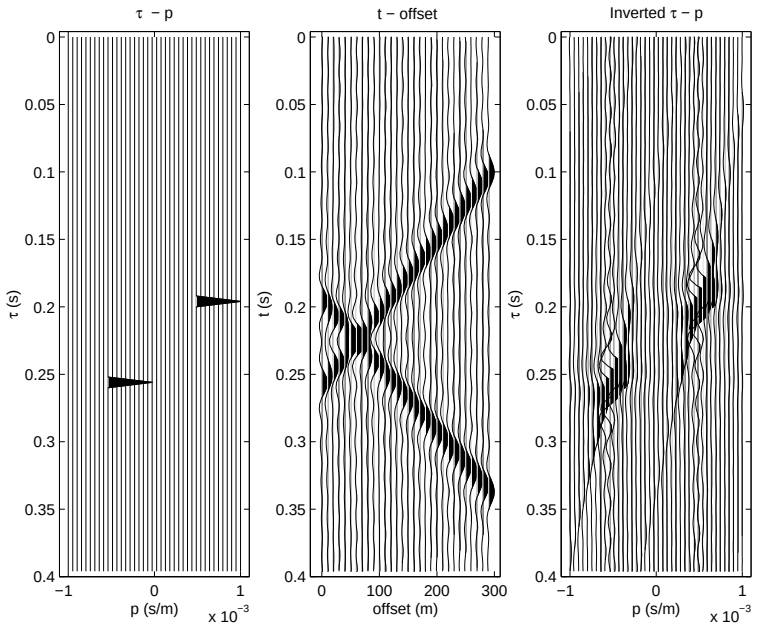

Figure 7.3: Left: Ideal $\tau - p$ panel. Center: Data generated by forward transforming and band-limiting the ideal $\tau - p$ panel. Band-limiting is needed to eliminate alias. Right: Inverted $\tau - p$ panel.

Assume that in a CMP (common mid point gather) you have two events: a primary and a multiple. Let us assume that the intercept time of these events T_0 is the same.

The travel-time curve for the primary is given by:

$$T_p = \sqrt{(T_0^2 + h^2/v_p^2)} \tag{7.46}$$

,

and the travel-time for the multiple is given by:

$$T_m = \sqrt{(T_0^2 + h^2/v_m^2)}\,. \tag{7.47}$$

If the multiple is generated by a shallow layer or by the water column we can consider $v_p > v_m$.

Now suppose that we apply NMO correction to the complete data set with the NMO law that uses the velocity of the primary. The NMO correction entails applying the following time shift to the data

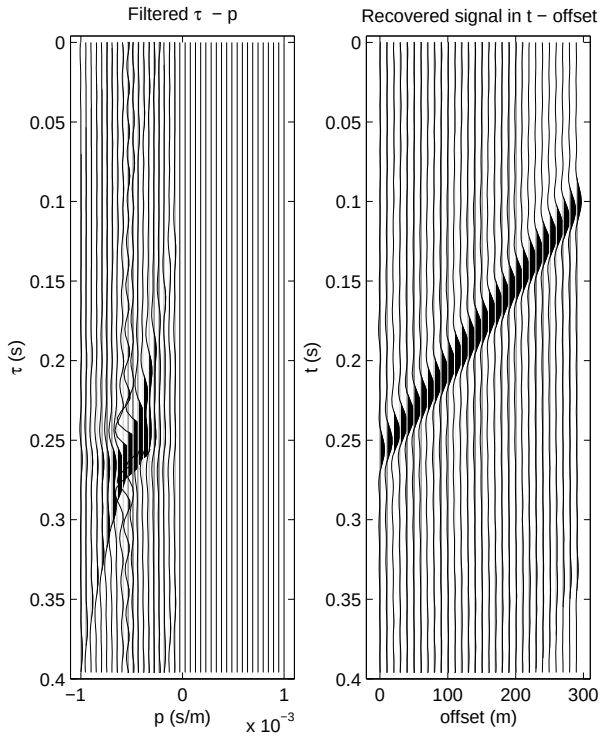

Figure 7.4: Left: Inverted $\tau - p$ panel after muting. Right: Data reconstructed by forward modeling the inverted/muted $\tau - p$ panel. Note that the alias artifacts have disappeared.

$$\Delta T_{NMO} = T_0 - \sqrt{(}T_0^2 + h^2/v_{NMO}^2) \,, \tag{7.48}$$

therefore, the time of the primary after NMO is

$$T_p(After) = T_p + \Delta T_{NMO} \tag{7.49}$$

It is clear that is the NMO velocity is the velocity of the primary, the time of the primary becomes

$$T_p(After) = T_0 \,. \tag{7.50}$$

In other words, the primary has the same time for all offsets (a flat event). What is the time of the multiple after NMO?. Let us try to compute it,

$$T_m(After) = T_m + T_0 - \sqrt{(T_0^2 + h^2/v_{NMO}^2)}\,, \tag{7.51}$$

or after replacing T_m

$$T_m(After) = T_0 + \sqrt{(T_0^2 + h^2/v_m^2)} - \sqrt{(T_0^2 + h^2/v_{NMO}^2)} \tag{7.52}$$

The two square roots in the above equation can be expanded in Taylor series (Keeping only up to the second order term) we have

$$T_m(After) \approx T_0 + \frac{1}{2T_0\, v_m^2}h^2 - \frac{1}{2T_0\, v_{NMO}^2}h^2 \tag{7.53}$$

which can be re-written as

$$T_m(After) \approx T_0 + qh^2 \tag{7.54}$$

where

$$q = \frac{1}{2\,T_0}(\frac{1}{v^2} - \frac{1}{v_{NMO}^2})\,. \tag{7.55}$$

It is clear that a transform with parabolic integration path can be constructed by simple interchanging in the original slant stack h by h^2. Some people prefer to parameterize the parabola in terms of the residual moveout time at far offset,

$$t = \tau + q\frac{h^2}{h_{max}^2}\,,$$

then it is clear that the parameter q is nothing else that the moveout in seconds at the far offset trace.

In Figures (7.56) and (7.57) we portrayed a primary and a multiple before and after parabolic Radon transform filtering. In this example q is residual moveout at far offset.

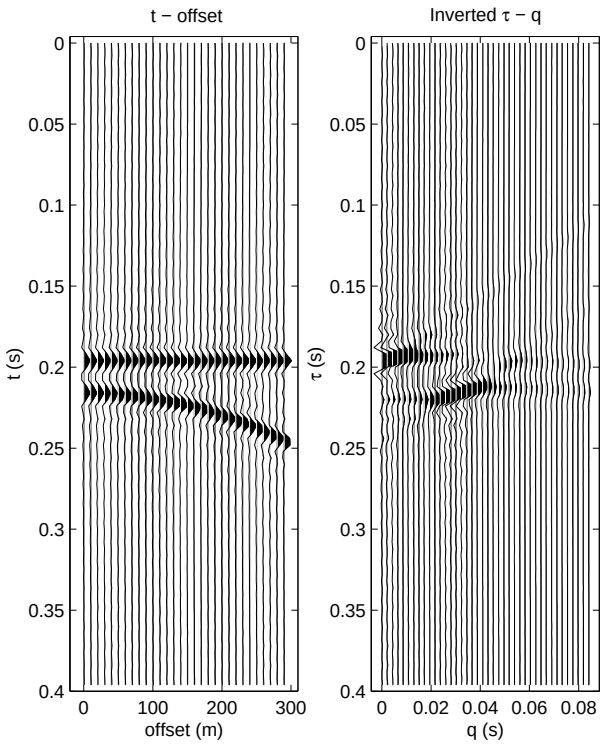

Figure 7.5: Left:A Primary and a multiple after NMO correction. Inverted $\tau - q$ panel.

7.4 High resolution Parabolic Radon Transform

The high resolution Parabolic Radon transform proposed by Sacchi and Ulrych (1995) entails the utilization of a regularization technique that leads to an operator that does not exhibit a Toeplitz structure. In the original formulation of the high resolution Radon transform the operator is inverted using Cholesky decomposition. This is quite expensive compared to the classical least squares Radon transform that uses the Levinson recursion to invert a Toeplitz form.

We propose a method to achieve high resolution at a computational cost of the order of the conventional parabolic least squares Radon transform. This feature makes our new algorithm quite attractive to process large data sets.

The Parabolic Radon transform is a widely accepted technique for multiple removal (Hampson, 1986). The technique can be implemented in the frequency domain via a fast algorithm that exploits the Toeplitz structure of the least squares Radon operator (Kostov, 1990; Darche, 1990). Recently, Sacchi and Ulrych (1995) proposed a high resolution algorithm to increment the ability of the transform to distinguish events with similar moveout curves. This algorithm is based on a procedure that attempts to find a sparse representation of the reflections in the parabolic Radon domain. A similar al-

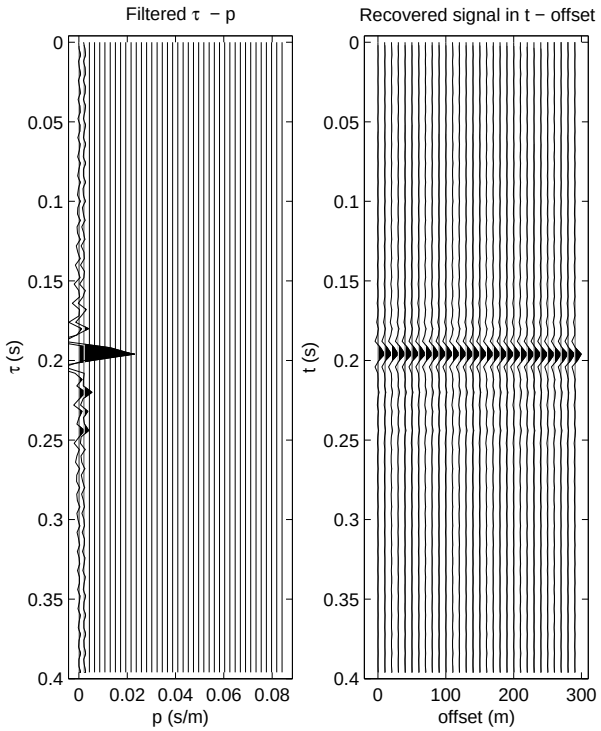

Figure 7.6: Left: Inverted $\tau - q$ panel after muting. Right: Data reconstructed by forward modeling the inverted/muted $\tau - q$ panel. In this example the multiple has been eliminated by muting in the $\tau - q$ domain.

gorithm has been proposed by Cary (1998). In this case the Radon panel is constrained to be sparse in both the Radon parameter and the intercept time. The high resolution parabolic Radon transform can be used to isolate multiples interferences with a few milliseconds of residual moveout at far offset. This is a problem frequently encountered when dealing with short period multiple reflections generated by carbonate targets in the Western Canadian Basin (Hunt et al., 1996).

One of the advantages of the high resolution parabolic Radon transform is that the focusing power of the transform is considerably increased with respect to the classical least squares parabolic Radon transform. Unfortunately, the high resolution parabolic Radon transform leads to the inversion of an operator that is Hermitian but does not exhibit a Toeplitz structure. The resulting Hermitian operator is inverted using Cholesky decomposition. The Cholesky method for solving Hermitian linear systems of equations requires a number of operations that is proportional to M^3, where M is the dimension of the Hermitian operator.

7.4.1 Least squares Parabolic Radon transform

Common mid point (CMP) gathers after normal moveout (NMO) correction can be modeled as a superposition of events with parabolic moveout:

$$d(x_j,t) = \sum_{k=1}^{M} m(q_k, \tau = t - q_k\, x_j^2)\,, j = 1, N\,, \tag{7.56}$$

where $d(x_j,t)$ denotes the CMP gather, x_j the offset, $m(q_k,\tau)$ is the Radon panel, q_k the discrete Radon parameter and τ the intercept time. The data con sist of N seismic traces which do not need to be regularly sampled. The Radon parameter is uniformly discretized according to $q_k = q_0 + \Delta q\,(k-1),\ k = 1,\ldots,M$.

Equation (7.56) is essentially a decomposition of the CMP gather in terms of parabolic events distributed in the plane τ, q. It is computational more convenient to rewrite the last equation in the frequency-offset domain. Taking Fourier transform with respect to the temporal variable t we arrive to the following expression

$$d(x_j,f) = \sum_{k=1}^{M} m(q_k,f)\, e^{i2\pi f q_k x_j^2}\,, j = 1,\ldots,N\,. \tag{7.57}$$

The calculations can be carried out independently for each frequency f. Equation (7.57) can be written in matrix form as follows:

$$\mathbf{d}(f) = \mathbf{L}(f)\,\mathbf{m}(f)\,. \tag{7.58}$$

To avoid notational clutter we will drop the frequency dependency in equation (7.58) and write $\mathbf{d} = \mathbf{L}\,\mathbf{m}$.

The least squares Radon operator is estimated by minimizing the following cost function.

$$J = ||\mathbf{d} - \mathbf{L}\,\mathbf{m}||^2 + \mu||\mathbf{m}||^2. \tag{7.59}$$

The regularization term $\mu||\mathbf{m}||^2$ is used to control the roughness of the solution. It can be shown that this term is one of the major sources of amplitude smearing in the Radon panel (Sacchi and Ulrych, 1995).

Taking derivatives of J with respect to $\mathbf{m}$ and equating them to zero yields

$$\begin{aligned}(\mathbf{L}^H\,\mathbf{L} + \mu\mathbf{I})\mathbf{m} &= \mathbf{L}^H\mathbf{d} \\ &= \mathbf{m}_{adj}\,.\end{aligned} \tag{7.60}$$

In the last equation $\mathbf{m}_{adj}$ denotes the low resolution Radon transform obtained using the adjoint or transpose operator $\mathbf{L}^H$. The least squares solution becomes

$$\begin{aligned} \mathbf{m} &= (\mathbf{L}^H \mathbf{L} + \mu \mathbf{I})^{-1} \mathbf{m}_{adj} \\ &= (\mathbf{R} + \mu \mathbf{I})^{-1} \mathbf{m}_{adj} \, . \end{aligned} \tag{7.61}$$

At this point some observations are in order. First it is clear that $\mathbf{R} = \mathbf{L}^H \mathbf{L} + \mu \mathbf{I}$ is a Toeplitz form (Kostov, 1990), with elements given by

$$\{\mathbf{R} + \mu \mathbf{I}\}_{l,m} = \sum_{k=1}^{N} e^{-i2\pi f \Delta q (l-m) x_k^2} + \mu \delta_{l,m} \, . \tag{7.62}$$

Solving this equation using the Levinson recursion requires approximately $4M^2 + 7M$ operations, and storage of only the first row of the Toeplitz matrix (Marple, 1987). This feature yields to a very efficient algorithm to compute the parabolic Radon transform.

7.4.2 High resolution parabolic Radon transform

In the high resolution parabolic Radon transform the vector $\mathbf{m}$ is retrieved by solving the following equation:

$$(\mathbf{R} + \mathbf{W}^H \mathbf{W}) \mathbf{m} = \mathbf{m}_{adj} \, . \tag{7.63}$$

The matrix $\mathbf{W}$ is a diagonal matrix with elements that depend on $\mathbf{m}$ (Sacchi and Ulrych, 1995). This leads to an iterative algorithm where $\mathbf{W}$ is bootstrapped from the result of a previous iteration. In general, the iterative procedure is not required if we are able to design $\mathbf{W}$ from a priori information. The matrix of weights $\mathbf{W}$ is a diagonal matrix with elements given by

$$\{\mathbf{W}\}_{l,m} = w_l \, \delta_{l,m}, \quad l, m = 1, \ldots, M \, . \tag{7.64}$$

The elements of the diagonal form $\mathbf{R} + \mathbf{W}^H \mathbf{W}$ become:

$$\{\mathbf{R} + \mathbf{W}^H \mathbf{W}\}_{l,m} = \sum_{k=1}^{N} e^{-i2\pi f \Delta q (l-m) x_k^2} + w_l^2 \delta_{l,m} \, . \tag{7.65}$$

It is clear that the addition of a diagonal matrix with non-constant e elements has destroyed the Toeplitz structure of the operator. The above matrix can be inverted by the

Cholesky method in a number of operations proportional to M^3. From the computational point of view it is more convenient to compute the Radon transform using a constant diagonal regularization (equation (7.60)). However, if we want to estimate a high resolution Radon operator, the regularization term must be a diagonal form with non-constant elements. The elements of $\mathbf{W}$ are used to emphasize the Radon parameters q_k that need to be constrained to be zero. In general, the matrix $\mathbf{W}$ is bootstrapped from the data in an iterative manner. The aforementioned procedure is described in Sacchi and Ulrych (1995).

In our synthetic example, the elements of the diagonal matrix $\mathbf{W}^H\mathbf{W}$ are given by

$$w_k^2 = \begin{cases} 100. & if \;\; q_k \notin Q \\ 0.0001 & if \;\; q_k \in Q\,, \end{cases} \tag{7.66}$$

where Q indicates the set of parameters q_k where the reflections are localized. These weights can be interpreted as the inverse of a variance in model space. If w_l^2 is large, $1/w_l^2$ is small and therefore, the algorithm will constraint the areas of no reflections in the τ, q space to be zero. It is clear that the resolution is enhanced by inhibiting the creation of smearing in the Radon panel.

7.4.3 Conjugate gradients and circulant matrices

To solve equation (7.63) we adopt the method of conjugate gradients, which is summarized below.

We want to solve $(\mathbf{R}+\mathbf{D})\mathbf{m} = \mathbf{m}_{adj}$, where $\mathbf{D} = \mathbf{W}^H\mathbf{W}$.

Start with an initial solution $\mathbf{m_0}$, set $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{m}_{adj} - (\mathbf{R}+\mathbf{D})\mathbf{m}_0$,

$\alpha_{i+1} = (\mathbf{r}_i, \mathbf{r}_i)/(\mathbf{p}_i, (\mathbf{R}+\mathbf{D})\mathbf{p}_i)$

$\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_{i+1}\mathbf{p}_i \;\; \mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_{i+1}(\mathbf{R}+\mathbf{D})\mathbf{p}_i$

$\beta_{i+1} = (\mathbf{r}_{i+1}, \mathbf{r}_{i+1})/(\mathbf{r}_i, \mathbf{r}_i)$

$\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1}\mathbf{p}_i$

where $i = 0, 1, 2, \ldots K$ denotes the iteration number.

The cost of the conjugate gradients algorithm is dominated by the cost of multiplying a matrix by a vector. In general, matrix times vector multiplication is an $O(M^2)$ process. In our problem we will use the Toeplitz structure of $\mathbf{R}$ to find a fast manner to compute the aforementioned operation.

The product $(\mathbf{R}+\mathbf{D})\mathbf{x}$ can be decomposed into two products: $\mathbf{Rx}+\mathbf{Dx}$. The first product

can be efficiently computed using the Fast Fourier Transform (FFT), the second product involves only $2M$ operations (M products plus M additions) and does not substantially increase the computational cost of the inversion.

The first product, $\mathbf{y} = \mathbf{R}\mathbf{x}$, is evaluated by augmenting the system as follows:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}' \end{bmatrix} = \mathbf{R}_{aug} \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix}, \tag{7.68}$$

where $\mathbf{R}_{aug}$ is the original Toeplitz matrix after being properly folded to become a circulant matrix (Strang, 1986; Schonewille and Duijndam, 1998). The right hand side can be computed by multiplying the Fourier transform of the first row of $\mathbf{R}_{aug}$ by the Fourier transform of vector $[\mathbf{x}, \mathbf{0}]^T$, and taking the inverse Fourier transform of this product. Now our matrix times vector operation takes $O(M' \log M')$ operations where M' is the size of augmented matrix ($M' = 2M$). We have found that the conjugate gradients algorithm convergences after a few iterations ($K \approx M/5$). Therefore, the inversion becomes an $O(K\, M'\, Log(M'))$ process. This is more efficient than the direct inversion of equation (7.63) by the Cholesky method.

7.4.4 Example

In Table 7.1 we present a comparison of CPU times in seconds for 3 different algorithms. The times in Table 1 correspond to the total computational cost for 512 frequencies. These simulations were performed on a SGI Origin 2000.

In both cases we have 4 parabolic events which were mapped to the Radon domain u sing the following algorithms:

1. **Lev**: Classical least squares parabolic Radon transform implemented via the Levinson recursion (valid for a constant damping).

2. **Chol**: High resolution Radon transform implemented via the Cholesky decomposition.

3. **CG+FFT**: High resolution parabolic Radon transform implemented via conjugate gradients plus matrix times vector multiplication using the FFT.

Is is clear that the new algorithm can achieved high resolution at a computational cost comparable to the one of the classical least squares Radon transform computed with the Levinson recursive solution.

In Figure (7.1) we portray the results obtained for the 256×256 simulation. Note that the differences between the high resolution Radon transform computed with the Cholesky decomposition and the proposed algorithm are minimal.

$N \times M$	**Lev**	**Chol**	**CG+FFT**
128×128	2	6	3
256×256	8	42	12

Table 7.1: CPU times in seconds for the 3 algorithms tested in this study. N denotes the number of traces and M the number of q parameters.

7.5 Programs for Slant Stack and Parabolic Radon Transforms

The following two programs are a MATLAB implementation of the Radon transform in $f-x$ and $f-p$. The inverse transform is solved using Least squares. The high resolution implementation using circulant matrices is a little bit more tricky and requires more than a few lines of Matlab.

Forward Transform

Operator to compute the forward linear and parabolic Radon transform.

```
function [d]=for_taup(m,dt,h,q,N,flow,fhigh);
%INV_TAUP    An inverse Radon transforms. Given the seismic data, this
%            function computes the Radon panel by inverting the Radon operator
%
%  [d] = for_taup(m,dt,h,q,N);
%
%
%  IN   m: the Radon panel (d(nt,nq)
```

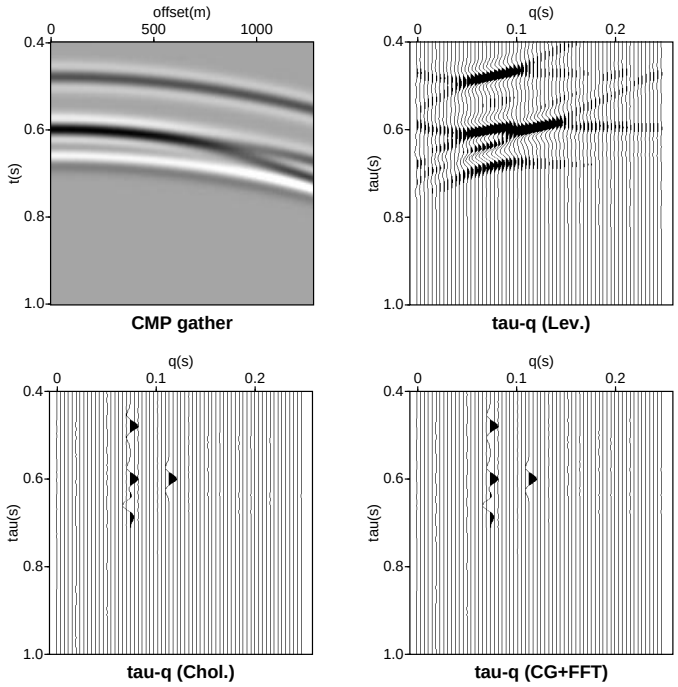

Figure 7.7: A synthetic CMP gather composed of 4 parabolic events is used to test 3 different algorithms to compute the Radon transform. **Lev.** indicates the classical solution using least squares with a constant damping term; the Levinson algorithm is used to invert the resulting Toeplitz form. **Chol.** indicates the high resolution solution using non-constant damping (8)), this solution is computed by means of the Cholesky decomposition. **CG+FFT** indicates the proposed fast algorithm to compute the high resolution Radon transform. In this example the size of the Radon operator is 256×256. CPU times in seconds are given in Table 7.1

```
%       dt: sampling in sec
%       h(nh) offset or position of traces in mts
%       q(nq) ray parameters to retrieve or curvature
%             of the parabola if N=2
%       N:1 Linear tau-p
%        :2 Parabolic tau-p
%       flow, fhig: min and max freq. in Hz
%
%  OUT  d: the data
%
```

```
%
%  SeismicLab
%  Version 1
%
%  written by M.D.Sacchi, last modified December 10,  1998.
%  sacchi@phys.ualberta.ca
%
%  Copyright (C) 1998 Signal Analysis and Imaging Group
%                     Department of Physics
%                     The University of Alberta
%

nt= max(size(m));
nh = max(size(h));


M = fft(m,[],1);
D = zeros(nt,nh);
i = sqrt(-1);

ilow  = floor(flow*dt*nt)+1; if ilow<1; ilow=1;end;
ihigh = floor(fhigh*dt*nt)+1;
if ihigh>floor(nt/2)+1; ihigh=floor(nt/2)+1;end

for if=ilow:ihigh
f = 2.*pi*(if-1)/nt/dt;
L = exp(i*f*(h.^N)'*q);
x = M(if,:)';
y = L * x;
D(if,:) = y';
D(nt+2-if,:) = conj(y)';
end
D(nt/2+1,:) = zeros(1,nh);
d = real(ifft(D,[],1));

return;
```

Inverse transform

Operator to compute the LS inverse Radon transform. Notice that this is an "academic" implementation. A fast implementation involves replacing `inv` by a fast solver (i.e., Levinson's recursion).

```
function [m] = inv_taup(d,dt,h,q,N,flow,fhigh,mu);
%INV_TAUP   An inverse Radon transform. Given the seismic data,
%           this subroutine computes
%           the Radon panel by inverting the Radon operator
%
%  [m] = inv_taup(d,dt,h,q,N,flow,fhigh,mu)
%
%  IN   d: seismic traces  (d(nt,nh)
%       dt: sampling in sec
%       h(nh) offset or position of traces in mts
%       q(nq) ray parameters to retrieve or curvature
%             of the parabola if N=2
%       N:1 Linear tau-p
%        :2 Parabolic tau-p
%       flow:  freq.  where the inversion starts in HZ (> 0Hz)
%       fhigh: freq. where the inversion ends  in HZ   (> Nyquist)
%       mu: regularization parameter
%
%  OUT  m: the linear or parabolic tau-p panel
%
%
%  SeismicLab
%  Version 1
%
%  written by M.D.Sacchi, last modified December 10,  1998.
%  sacchi@phys.ualberta.ca
%
%  Copyright (C) 1998 Signal Analysis and Imaging Group
%                     Department of Physics
%                     The University of Alberta
%

nt= max(size(d));
nq = max(size(q));
```

```
nh = max(size(h));

D = fft(d,[],1);
M = zeros(nt,nq);
i = sqrt(-1);

ilow  = floor(flow*dt*nt)+1; if ilow<1; ilow=1;end;
ihigh = floor(fhigh*dt*nt)+1;
if ihigh>floor(nt/2)+1; ihigh=floor(nt/2)+1;end

for if=ilow:ihigh
f = 2.*pi*(if-1)/nt/dt;
L = exp(i*f*(h.^N)'*q);
y = D(if,:)';
x = L'*y;

MATRIX = L'*L;
tr=real(trace(MATRIX));
   Q =mu*tr*eye(nq);
   x = inv(MATRIX+Q) *L'* y;
M(if,:) = x';
M(nt+2-if,:) = conj(x)';
end
M(nt/2+1,:) = zeros(1,nq);
m = real(ifft(M,[],1));
return
```

7.6 Time variant velocity stacks

We will discuss in this section the computation of time variant operators that can be used as an alternative to the parabolic Radon transform.

The parabolic Radon transform is a time invariant operator, therefore it can be implemented in the frequency domain. This trick permits one to solve several small problems, one at each frequency, instead of a large problem involving all the time-offset-velocity samples at the same time.

It is clear that in the case of the parabolic Radon transform time invariance is achieved by means of an approximation. It might happen that the parabolic approximation is not properly satisfied and consequently, travel-times (especially at far offsets) are not properly modeled.

In this part of the course, we will focus our attention of the computational aspects of Hyperbolic Radon operators.

We have already mentioned that the data in the CDP domain can be modeled as a superposition of hyperbolas. User this assumption a hyperbolic stack operator can be used to map hyperbolas (reflections) into $time - velocity$ pairs. In other words, our operator is used to map data from $offset - time$ to $velocity - time$ space. In the new space we can identify multiple reflection and filter them out. We can also use this type of operators to reduce random noise and to enhance the overall aspect of the seismic reflection which might be hidden by strong ground-roll (in a CSG) or any other type of deterministic noise.

The time-variant velocity-stack operator is defined in terms of summation along Dix hyperbolas, $m(\tau, v)$ is used to designate the velocity-stack and $d(t, h)$ the CMP gather:

$$d(t,h) = \int m(\tau = \sqrt{t^2 - h^2/v^2}, v)\, dv\,, \tag{7.69}$$

where h is source-receiver offset, t is two-way travel-time, v is the rms velocity, and τ is two-way vertical travel-time. After discretization and lexicographic arrangement, equation (7.69) can be written as

$$d = Lm\,. \tag{7.70}$$

The vectors d and m have $nt \times nh$ and $n\tau \times nv$ elements, respectively. The dimension of the operator L is $(nt \times nh) \times (nt \times n\tau)$. The forward or modeling operator, L, picks a

wavelet in velocity space and produces a hyperbola in data space. The transpose operator L^T is a simple NMO followed by stacking operator.

To find the inverse operator we consider the problem

$$Minimize\ \{\phi = ||Lm - d||_2^2\}$$

Differentiating ϕ with respect to m yield the least-squares solution

$$\hat{m} = (L^T L)^{-1} L^T d = (L^T L)^{-1} m_0\,, \tag{7.71}$$

where for simplicity we have assumed that L is full rank. In equation (7.71) m_0 is the low resolution velocity-stack computed by means of the adjoint or transpose operator (Sacchi and Ulrych, 1995). The velocity stack computed after inversion, $\hat{m}$, possesses more resolution that m_0. Unfortunately, the computation of $\hat{m}$ involves the inversion of $L^T\,L$. If we assume a typical CMP gather of 48 channels and 1000 samples per trace. In addition, suppose that 48 traces and 1000 samples were used to discretize the velocity, v, and the intercept time, τ, respectively. In this case $L^T L$ has dimension 24000×24000. It is evident that direct methods cannot be applied in this type of problems.

7.6.1 The conjugate gradients algorithm

The trick here is to use a semi-iterative technique to find an approximate solution to our problem. The advantage of the CG algorithm is the the matrix L does not need to be stored. In fact, L is not even a matrix but an operation perform on a vector. To apply the CG algorithm we need first to define the operations L and L^T.

It is clear that L is an operator that picks a wavelet in the $\tau - v$ and produces a hyperbola in $t-h$. The operator L^T (the adjoint or transpose operator) does the opposite, it gathers information in $t - h$ along a hyperbolic path and collapses this information into a point in $\tau - v$.

Let us assume that we have a code capable of performing the following operations (as I have already mentioned L and L' do not need to be matrices)

$$y = Lx \quad x' = L^T y \, .$$

To solve the problem $||Lx - y||^2$ with an initial solution x_0, we use the following Conjugate Gradients (CG) algorithm:

Set initial values: $r = y - L\, x_0$, $g = L^T\, r$, $s = g$

for i = 1:$ITERMAX$ {

$ss = Ls$, $\delta = ||ss||^2$

$\alpha = \gamma / \Delta$

$x = x + \alpha\, s$

$r = r - \alpha\, ss$

$g = L^T\, r$

$s = g + \beta\, s$

- }

The CG algorithm will find the least squares solution for the over-determined problem in N iterations where N is the total number of observations. In the under-determined problem, the CG converges to the minimum norm solution. The technique gives the exact answer for exact arithmetics, but of course round-off errors will affect the convergence of the algorithm. This is why the CG is often referred as a semi-iterative technique.

In the computation of the velocity stacks, we will use on a few iterations. How many iterations?. We can say, that we will use enough iterations to properly model the hyperbolic events. In fact, the CG method allows us to explore efficiently our solution by stopping the algorithm at any number of iteration and then, if the solution is not optimal, we can re-start the algorithm until a satisfactory misfit is obtained.

7.6.2 Example

We will analyze the performance of the CG with synthetic and real data examples. In Figure (7.8a) we portray a synthetic CMP gather. The model is composed of 2 primaries

reflections of 1500m/s (water column) and a primary of 1700 m/s at 0.65 s. In Figure (7.8b) we portray the velocity gather obtained using the adjoint operator. It is clear that this gather does not offer enough resolution to properly identify and separete the multiple event at 0.65 s from the primary. In Figure (7.8c) we portray the velocity gather obtained after inverting the data using the CH algorithm. Figure (7.8d) is the primary obtained after muting the velocy gather.

In Figure (7.9) I displayed the velocty gather obtained via the CG algorithm after amplitude clipping. In this panel we also portray the artifacts that arise from finite aperture and sampling (alias).

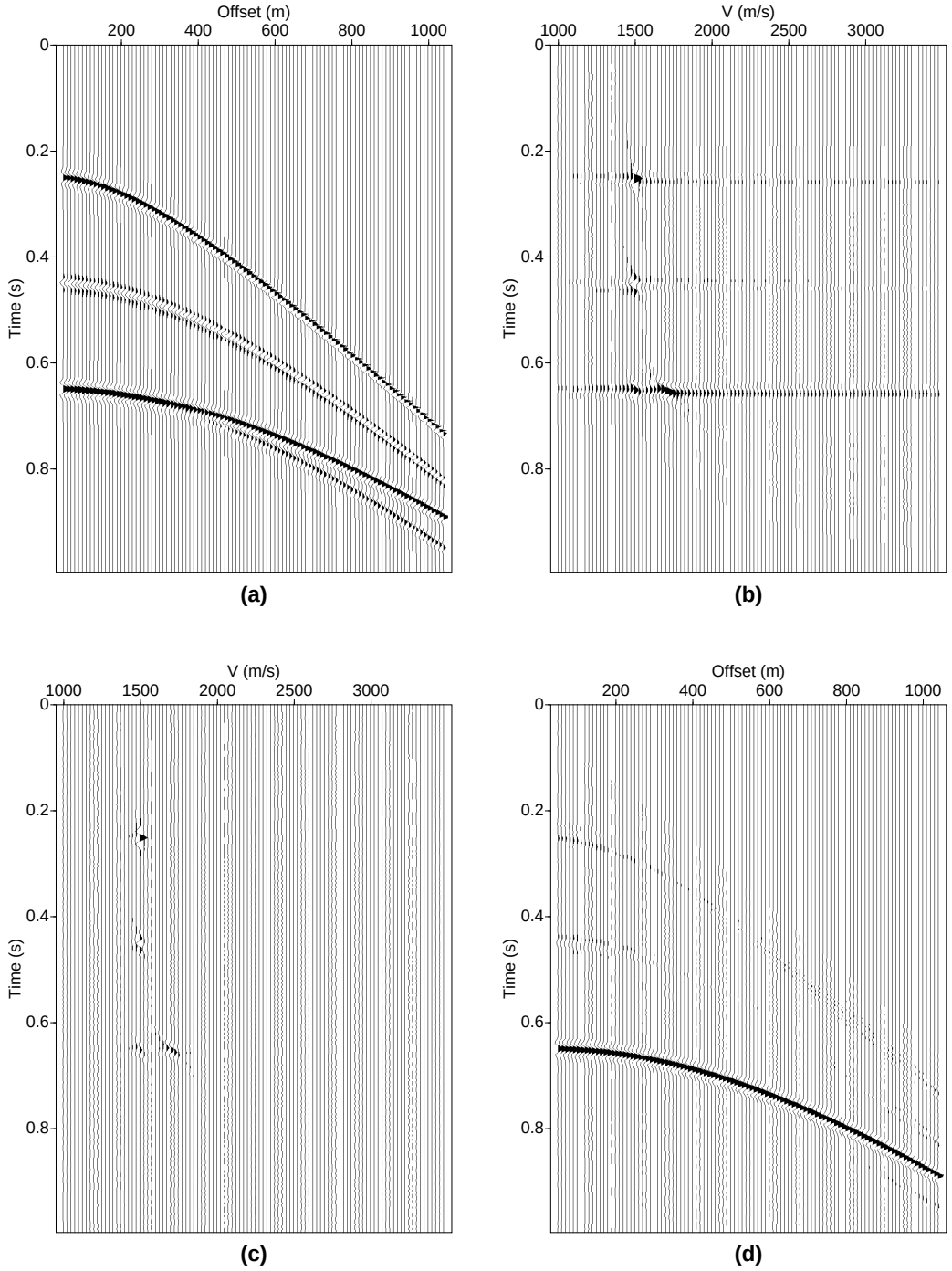

Figure 7.8: (a) Synthetic data. (b) Velocity gather obtained using the adjoint operator. (c) Velocity gather computed using the least-squares inversion. (d) Recovered data (primary) obtained after the de-multiple process.

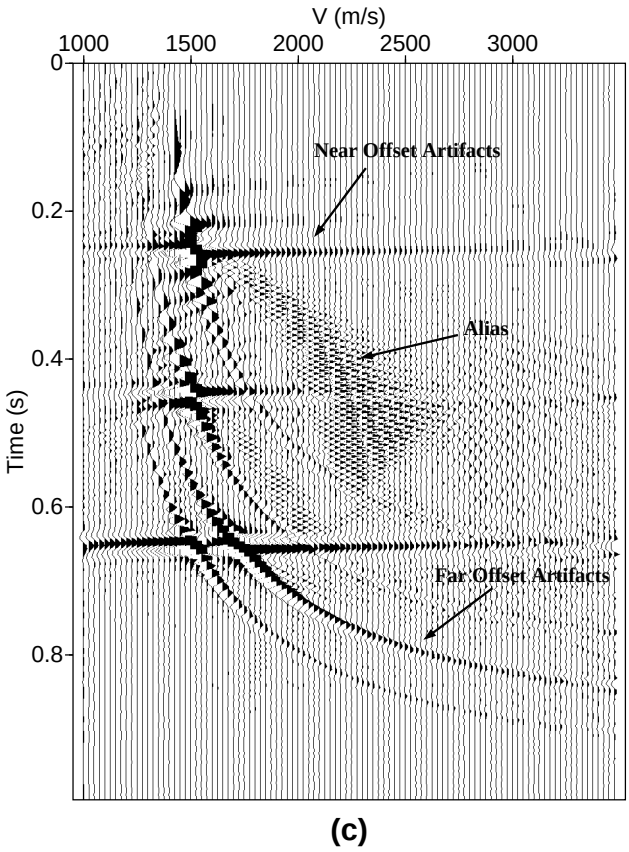

Figure 7.9: Clipped version of Figure (7.8)b showing finite aperture and sampling (alias) artifacts.

The following subroutine permits to compute forward and adjoint Hyperbolic Radon operators. I have included a linear interpolation step. See how do I define the adjoint of interpolation. It is important to stress that the forward and adjoint pairs must pass the dot product test. Otherwise, the CG inversion will not work. Once you have your forward and adjoint operators encapsulated in a subroutine it's quite simple to put together a CG inversion code.

```
      subroutine Hyperbolic_Radon(dt,nt,v,nv,h,nh,m,d,c)
c
c  Compute velocity panels  when c = 'a' (Adjoint Hyper. Radon)
c  Compute CMP gathers when      c = 'f' (Forward Hyper. Radon)
c                                c is character * 1
c INPUT
c  dt : sampling in sec
c  nt : number of time samples (also number of tau samples)
c  v(nv) : axis of the Radon panel (velocity in m/s). It can be
c          changed by 1/vel^2 or moveout at far offset
c  h(nh) : offset in meters
c          h(1) is offset of trace 1, h(2) is offset of trace 2....
c
c INPUT/OUTPUT
c  d(nh,nt) : cmp or super-cmp  input if c = 'a'
c  m(nv,nt) : Radon panel       output if c = 'a'
c
c  d(nh,nt) : cmp or super-cmp  output if c = 'f'
c  m(nv,nt) : Radon panel       input if c = 'f'
c

      real        d(300,2000), m(300,2000),h(300)
      real        v(300)

      character * 1 c

      if(c.eq.'a') call clean(m,nv,nt)  ! initialize m with zeros
      if(c.eq.'f') call clean(d,nh,nt)  ! initialize d with zeros

      do ih =1,nh
       do iv=1,nv
        do itau=1,nt
```

```
   ttt=(itau-1)*dt
   time=sqrt(ttt**2+(h(ih)/v(iv))**2)
   it1 = int(time/dt)
   a = time/dt - float(it1)       !Coeff. of the linear interp.
   it2 = it1 + 1

if(it1.lt.nt.and.it1.ge.1) then
 if(c.eq.'a') m(iv,itau) = m(iv,itau)+(1.-a)*d(ih,it1)+a*d(ih,it2)
 if(c.eq.'f') d(ih,it1)  = d(ih,it1)+(1.-a)*m(iv,itau)
 if(c.eq.'f') d(ih,it2)  = d(ih,it2)+    a *m(iv,itau)
endif

  enddo    ! end offset loop
 enddo    ! end velocity loop
enddo    ! end tau loop

return
end
```

7.7 High Resolution Radon Transform

We can contruct a solution m that consists on a few isolated spikes in velocity space. This is what we often call a sparse solution. We have already outlined a procedure to compute sparse solutions using the Parabolic Radon transform. In that case the spareness constraint was used to invert the Radon operator in the frequency domain. When using hyperbolic Radon transforms, the sparseness constraint has to be imposed in the $\tau - v$ domain. In general one can use any measure of sparseness (we have seen various norms that can be used to retrieve sparse models when dealing with impedance inversion in Chapter 4). Let's assume that we use a Cauchy-like norm (Sacchi and Ulrych, 1995). In this case we minimize

$$J = ||Lm - d||_2^2 + \mu \sum_k \ln(1 + m_k^2/b) \tag{7.72}$$

where m_k indicates an element of $m(\tau, v)$ after lexicographic arrangement (transformation of a matrix into a vector). The parameters μ and b are the hyper-parameters of the problem.

Taking derivatives of J with respect to m_k and equation them to zero leads to the following system

$$L^T\, L\, m - L^T\, d + Q\, m = 0 \tag{7.73}$$

where Q is a diagonal matrix with elements given by

$$Q_i = \frac{2\,\mu}{b + m_i^2}$$

It is clear that the system needs to be solved in an iterative manner (Q depends on the unknown model m). We can rewrite our solution as follows:

$$m^k = (L^T\, L + Q^{k-1})^{-1}\, L^T d\,. \tag{7.74}$$

where k indicates the iteration. The matrix of weights Q is computed from the result of the previous iteration. In general, one solve the problem for a given matrix of weights Q using CG, then after enough iteration to reach convergence, Q is updated and a CG is run again to solve the linear problem. The procedure is continues until we find the minimum of the cost function J. This algorithm can be very expensive, and in general a good felling for the μ and b is required to reach a sparse solution. When working with real data the parameters needed for the inversion (μ, b, number of iterations) are estimated

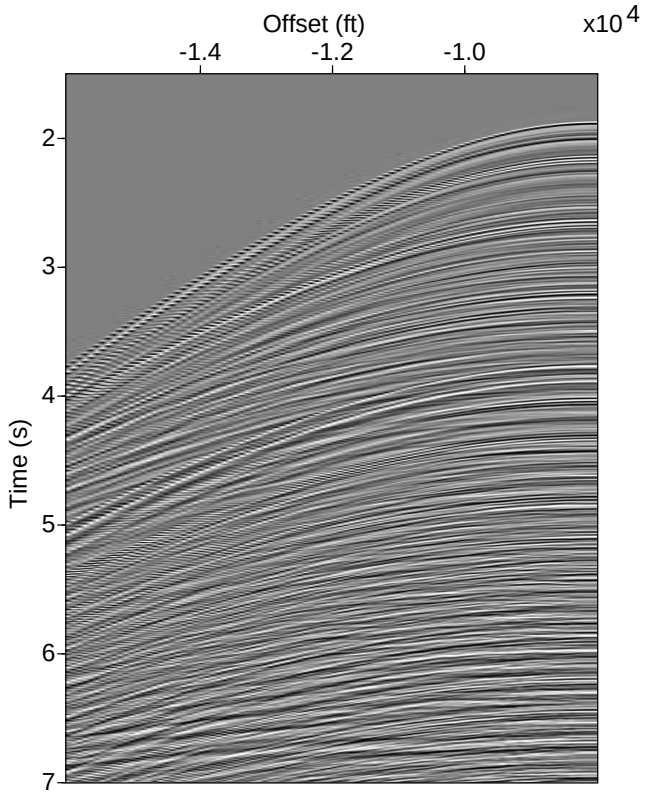

Figure 7.10: CMP gather # 1000 from a data set from the Gulf of Mexico.

by trial an error from a single CMP gather, the same parameters are used to invert the rest of the CMPs in the seismic volume.

In Figures (7.10), (7.11), (7.12), and (7.13) we test the high resolution hyperbolic Radon transform with a data set from the Gulf of Mexico (data provided by Western Geophysical to test multiple attenuation codes).

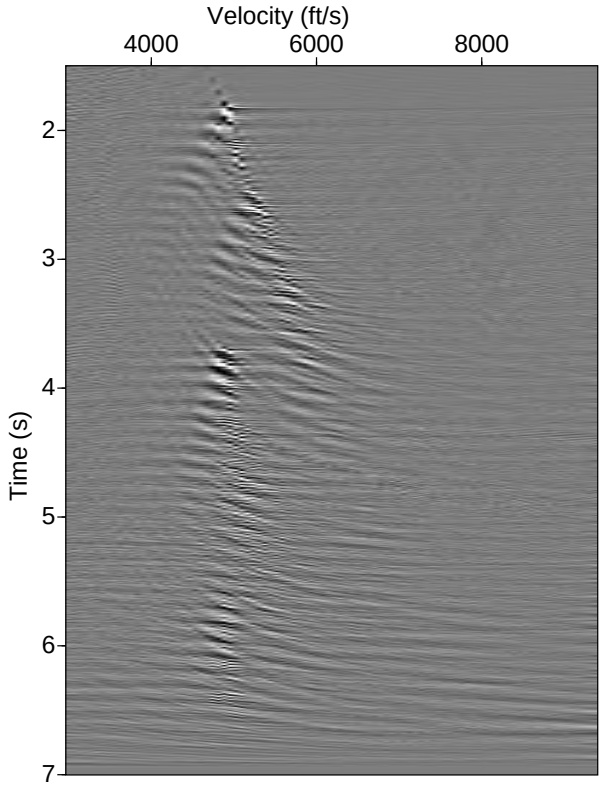

Figure 7.11: Velocity panel obtained by inversion of the Hyperbolic Radon transform using least-squares. CMP gather # 1000 from a data set from the Gulf of Mexico.

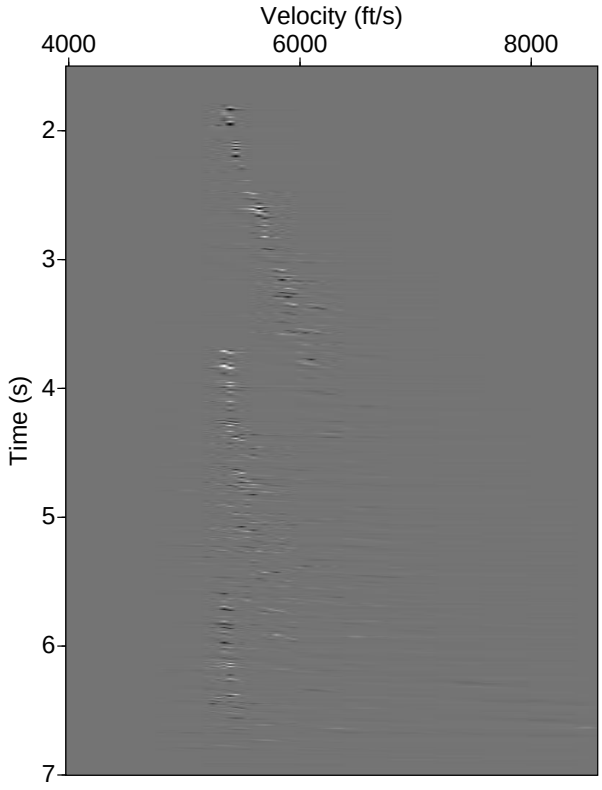

Figure 7.12: Velocity panel obtained by inversion of the Hyperbolic Radon transform using sparse inversion. CMP gather # 1000 from a data set from the Gulf of Mexico.

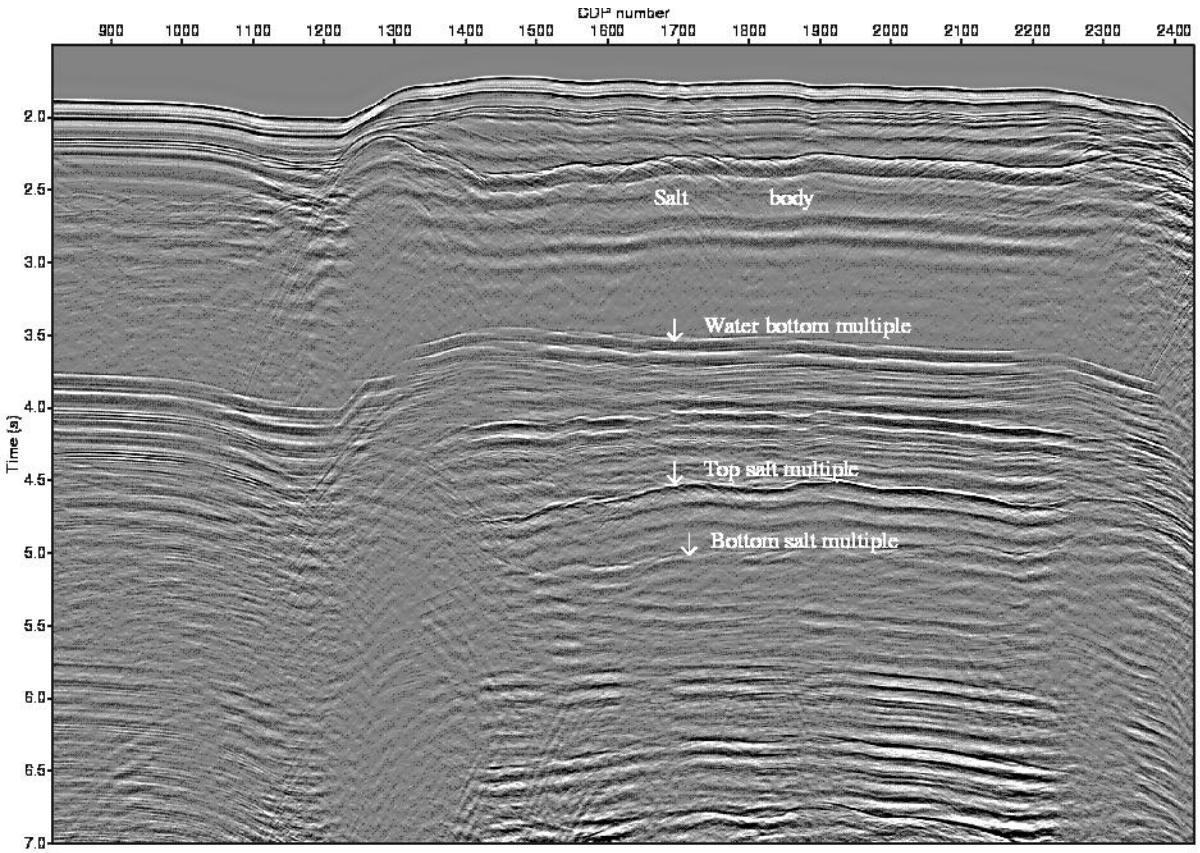

Figure 7.13: Stack section of the Gulf of Mexico data set before multiple removal.

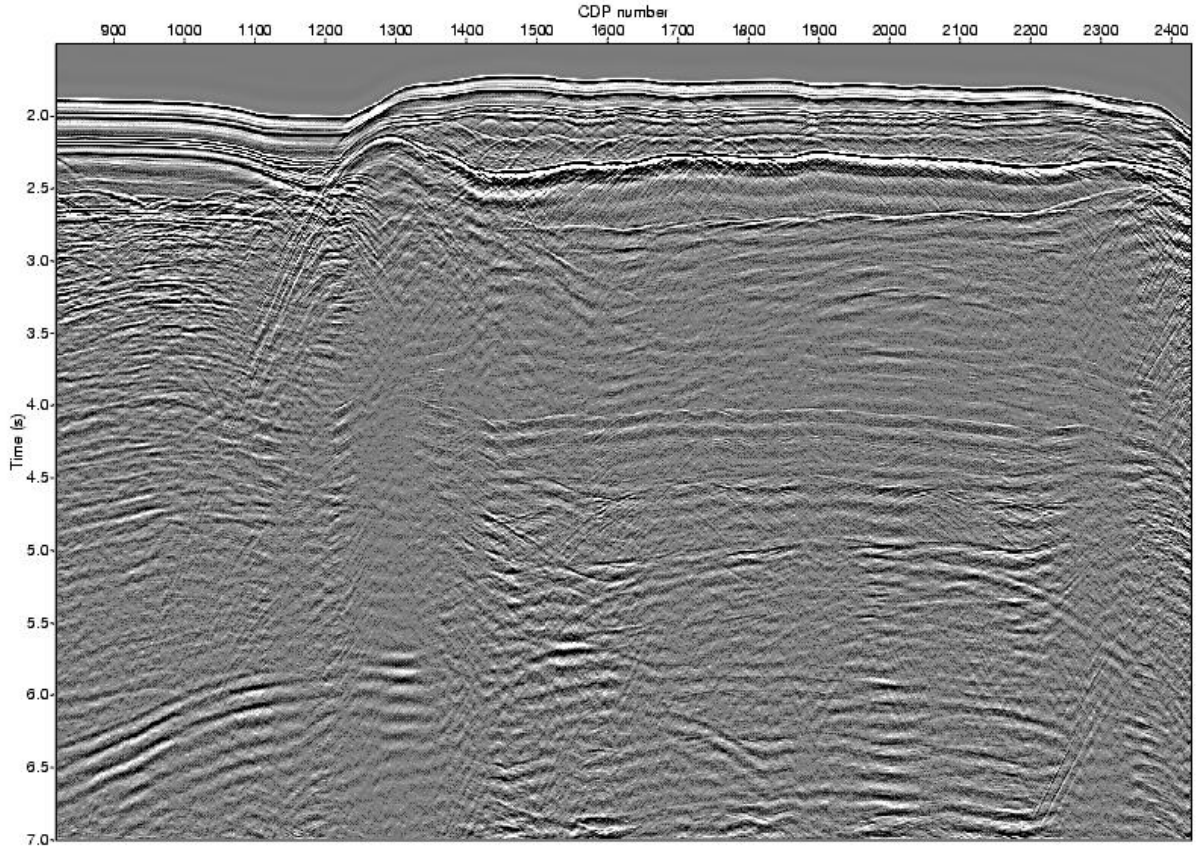

Figure 7.14: Stack section after multiple removal.

7.8 Interpolation problems

The parabolic and hyperbolic Radon transform can be used to interpolate CMP gathers. This is quite simple and, basically, entails mapping back the velocity stack to a data space using a new geometry. To interpolate pre-stack data in receiver-source space (or midpoint-offset) a more sophysicated approach is required. Various research group in the area of signal analysis have proposed algorithms to interpolate 1D data. These algorithms assume that the data are band-limited. One can extend these ideas to the problem of reconstructing pre-stack data. In geophysics Duijndam et. al (1999) and Hindriks et. al (1997) have introduced a least-squares algorithm to invert the fourier transform of the data. We will review some basic features of these algorithms and introduce a regularization term that enables us to recover large gaps in our pre-stack data set.

We define the discrete 2-D inverse Fourier transformation in source and receiver coordinates as

$$u(x_s, x_r, \omega) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} U(k_s(m), k_r(n), \omega) e^{jk_s(m)x_s} e^{jk_r(n)x_r}, \tag{7.75}$$

where x_s and x_r are the spatial variables along source and receiver coordinates, k_s and k_r are the corresponding wave-numbers and ω is the temporal frequency. Equation (7.75) gives rise to a linear system equations

$$u = AU \tag{7.76}$$

where

$$A_{mn} = \frac{1}{MN} e^{jk_s(m)x_s} e^{jk_r(n)x_r}, \tag{7.77}$$

u and U denote the known data and unknown coefficients of the DFT, respectively.

Therefore, the interpolation problem can be posed as finding, from the incomplete data, the 2D-DFT (U) by solving

$$u = AU + n \tag{7.78}$$

where n denotes the noise in the data. A unique solution may be obtained by minimizing the following expression

$$J = \| AU - u \|_2^2 + \epsilon \| U \|_2^2 \tag{7.79}$$

and the solution can be shown to take the form:

$$\hat{U} = (A^T A + \epsilon I)^{-1} A^T u, \tag{7.80}$$

where T denotes the transpose of a matrix.

Next, we derive a similar result but using a weighted DFT-domain norm introduced in the previous section. In this case the function to be minimized is

$$J = \| AU - u \|_2^2 + \epsilon \| U \|_P^2 \tag{7.81}$$

The solution takes the form:

$$\hat{U} = (A^T A + DI)^{-1} A^T u, \tag{7.82}$$

where D is a diagonal matrix with diagonal elements corresponding to $\frac{\epsilon}{|P(k_s,k_r)|^2}$ and $|P(k_s,k_r)|^2$ is a vector that contains the amplitude spectrum of U in lexicographic form . Ideally, one should know the amplitude spectrum of the data. Unfortunately, U is the unknown of our problem. The latter can be overcome by defining an initial D in terms of the DFT of the irregularly sampled data $A^T u$ and smoothing the result to attenuate the artifacts introduced by the irregularity of u (Ning and Nikias, 1990).

The scheme can be summarized as follows:

- Start with an initial $\hat{U}$.
- Compute $D = S(\hat{U}^* \hat{U})$, where S is a smoothing filter.
- Solve $\hat{U} = (A^T A + DI)^{-1} A^T u$ using Conjugate Gradients.
- Iterate until convergence.

An example of reconstruction is demonstrated on 15 synthetic shot gathers. Figure 7.15 shows six of the shot gathers with the shots #3 and #7 removed. The reconstruction is performed using the minimum weighted norm method with adaptive weights. Figure 7.16 shows the reconstructed shot gathers (only six shots are shown). The missing shots have been completely reconstructed.

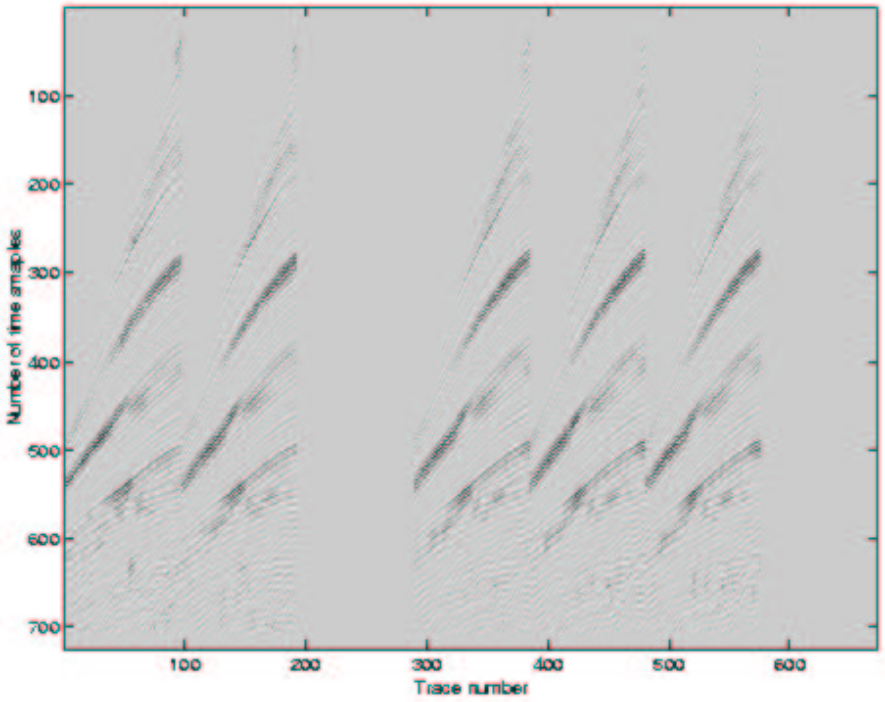

Figure 7.15: Six shot gathers with shots #3 and #7 removed.

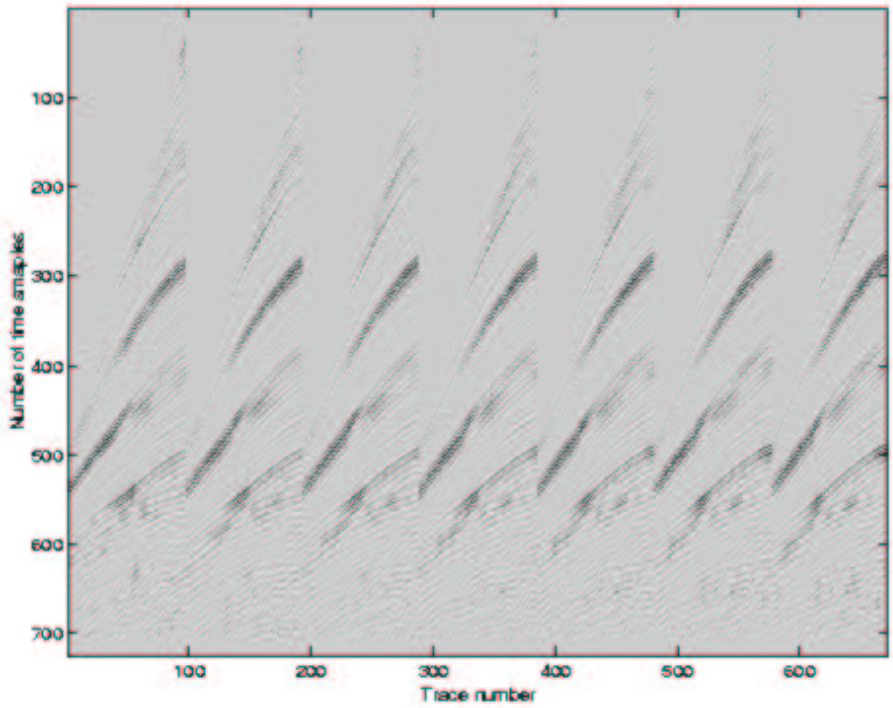

Figure 7.16: Reconstructed shot gathers using the minimum weighted norm algorithm.

7.9 References

Radon

Beylkin, G., 1987, Discrete radon transform: IEEE Trans. Acoust., Speech, Signal Processing., **ASSP-35**, 162-172.

Cary, P., W., 1998, The simplest discrete Radon transform, 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1999-20 02.

Chapman, C. H., 1981, Generalized Radon transforms and slant stacks: Geophys. J. Roy. Astr. Soc., **54**, 481-518.

Deans, S. R., 1983, The Radon transform and some of its applications: J. Wil ey and Sons, Inc.

Durrani, T. S., and Bisset, D., 1984, The Radon transform and its properties : Geophysics, **49**, 1180-1187.

Foster, J. D., and Mosher, C. C., 1992, Suppression of multiples reflection using the Radon transform: Geophysics, **57**, 386-395.

Hampson, D., 1986, Inverse velocity stacking for multiple elimination: J. Can. Soc. Expl. Geophys., **22**, 44-55.

Hunt, L., Cary, P., and Upham, W., 1996, An improved Radon Transform for short period multiple attenuation: CSEG 23rd Annual Mtg., Expanded Abstracts, 58-59.

Kostov, C., 1990, Toeplitz structure in slant-stack inversion: 60th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1618-1621.

Phinney, R. A., Chowdhury, K. R., and Frazer, L. N., 1981 Transformation and analysis of record sections: J. Geophys. Res., **86**, 359-377.

Pratt, W. K., 1991, Digital image processing: John Wiley and Sons, Inc

Sacchi, M.. D. and Ulrych, T., J., 1995, High-resolution velocity gathers and offset space reconstruction: Geophysics, **60**, 4, 1169-1177.

Thorson, J. R., and Claerbout, J. F., 1985, Velocity-stack and slant stack stochastic inversion: Geophysics, **50**, 2727-2741.

Yilmaz, Ö., 1989, Velocity-stack processing: Geophys. Prosp., **37**, 357-382.

Yilmaz, O., and Taner, M. T., 1994, Discrete plane-wave decomposition by least-mean-square-error method: Geophysics, **59**, 973-982.

Zhou, B., and Greenhalgh, S. A., 1994, Linear and parabolic $\tau - p$ revisited: Geophysics, **59**, 1133-1149.

FT Interpolation

Duijndam, A.J.W., Schonewille, M., and Hindriks, K., 1999, Reconstruction of seismic signals, irregularly sampled along on spatial coordinate: Geophysics, 64, 524-538.

Hindriks, K. O. H., Duijndam, A. J. W. and Schonewille, M. A., 1997, Reconstruction of two-dimensional irregularly sampled wavefields: Annual Meeting Abstracts, Society Of Exploration Geophysicists, 1163-1166.

Ning, T. and Nikias, C.L., 1990, Power Spectrum Estimation with Randomly Spaced Correlation Samples: IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 6, 991-997.

Sacchi, M.D. and Ulrych, T.J., 1998, Interpolation and extrapolation using a high-resolution discrete Fourier transform: IEEE Trans. Signal Processing, vol. 46, no. 1, 31-38.