

# An algorithm for 3D residual static corrections

William S. Harlan (based on ideas from Greg Lazear)

1996

This algorithm is designed to obtain enough information in one pass through 3D surface seismic data to estimate reflection static corrections. During this pass, a running buffer retains traces whose sources and receivers are close to the next expected trace. Each new trace is cross-correlated with traces in the buffer, and the cross-correlations are saved in a disk file. The necessary static corrections are optimized iteratively from the saved correlation functions, and corrections are applied in a second and final pass through the data. This note uses a notation which might clarify the details.

Assume that surface seismic data amplitudes  $d(\mathbf{x}_i, \mathbf{x}^k, t)$  were recorded over time  $t$  for many pairs of 3D ground positions for sources  $\mathbf{x}_i$  and receivers  $\mathbf{x}^k$ . Sources are indexed with subscripts, and receivers with superscripts.

The time axis has been stretched according to some “normal moveout” which is a function of the source-receiver offset  $\|\mathbf{x}_i - \mathbf{x}^k\|$  and the midpoint  $\|\mathbf{x}_i + \mathbf{x}^k\|/2$ . After such adjustments, nearby traces are assumed to resemble each other within a time window, except for time-invariant, surface-consistent source shifts  $\Delta t_i$  and receiver shifts  $\Delta t^k$ :

$$d(\mathbf{x}_i, \mathbf{x}^k, t) \approx d(t - \Delta t_i - \Delta t^k) \text{ for } t_1 < t < t_2. \quad (1)$$

The approximation is expected to worsen as the collection of traces includes sources and receivers farther apart. The time window should be as deep as possible and should include strong reflections.

Define a cross correlation  $c_{ij}^{kl}(\tau)$  for specific pairs of nearby sources and receivers:

$$c_{ij}^{kl}(\tau) \equiv c_{ji}^{lk}(-\tau) \equiv \int_{t_1}^{t_2} d(\mathbf{x}_i, \mathbf{x}^k, t - \tau) d(\mathbf{x}_j, \mathbf{x}^l, t) dt. \quad (2)$$

These crosscorrelations are calculated only between traces whose sources, receivers, and/or midpoints are within a limited distance of each other:

$$\|\mathbf{x}_i - \mathbf{x}_j\| < x_{\max}, \quad \|\mathbf{x}^k - \mathbf{x}^l\| < x_{\max} \quad \text{and} \quad \|(\mathbf{x}_i + \mathbf{x}^k)/2 - (\mathbf{x}_j + \mathbf{x}^l)/2\| < x_{\max}. \quad (3)$$

To the extent that our data model (1) is correct, then

$$\text{the lags } \tau_{ij}^{kl} \approx \Delta t_i - \Delta t_j + \Delta t^k - \Delta t^l \text{ will maximize } c_{ij}^{kl}(\tau_{ij}^{kl}). \quad (4)$$

The following algorithm suggests itself. Pick lags  $\tau_{ij}^{kl}$  that do maximize the correlations, then solve a least-squares problem:

$$\text{find } \Delta t_i \text{ and } \Delta t^k \text{ to minimize } \sum_{ijkl} (\tau_{ij}^{kl} - \Delta t_i + \Delta t_j - \Delta t^k + \Delta t^l)^2. \quad (5)$$

The above algorithm requires that a great many noisy cross-correlations be picked for possibly unreliable peaks. Instead, we can prepare and model cross-correlations of different partial sums of the data. Some information will be lost, but the sums will improve the signal-to-noise ratio of the correlations and will improve the reliability of the picked lags at correlation peaks.

Several partial sums are possible. The most common is probably a cross-correlation of each trace with a stack of those traces that share a particular midpoint. (Remember the nearby-midpoint constraint in (3) applies to all crosscorrelations.)

$$C_i^k(\tau) \equiv \sum_{jl} c_{ij}^{kl}(\tau). \quad (6)$$

In addition we could calculate the following sums of all crosscorrelations sharing a trace with a particular source or receiver

$$C_i(\tau) \equiv \sum_{jkl} c_{ij}^{kl}(\tau) \quad \text{and} \quad C^k(\tau) \equiv \sum_{ijl} c_{ij}^{kl}(\tau) \quad (7)$$

or also calculate the sum of all crosscorrelations sharing traces with a particular pair of sources or receivers

$$C_{ij}(\tau) \equiv \sum_{kl} c_{ij}^{kl}(\tau) \quad \text{and} \quad C^{kl}(\tau) \equiv \sum_{ij} c_{ij}^{kl}(\tau). \quad (8)$$

Again we limit sums to nearby traces as in (3). We can examine the lags over a limited range  $|\tau| < \tau_{\max}$  and find the peaks of any of these summed crosscorrelations:

$$\{\tau_i^k, \tau_i, \tau^k, \tau_{ij}, \tau^{kl}\} \text{ maximize } C_i^k(\tau_i^k), C_i(\tau_i), C^k(\tau^k), C_{ij}(\tau_{ij}), \text{ and } C^{kl}(\tau^{kl}). \quad (9)$$

The peaks of the crosscorrelations should ideally coincide with sums of surface-consistent shifts. To fit any or all of these picked peaks, find the sum of source shifts  $\Delta t_i$  and receiver shifts  $\Delta t^k$  which fit the picked lags in a least-squares sense:

$$\begin{aligned} \min_{\Delta t_i \Delta t^k} & \sum_{i,k} (\tau_i^k - \Delta t_i - \Delta t^k)^2 + \sum_i (\tau_i - \Delta t_i)^2 + \sum_k (\tau^k - \Delta t^k)^2 + \\ & \sum_{i,j} (\tau_{ij} - \Delta t_i + \Delta t_j)^2 + \sum_{k,l} (\tau^{kl} - \Delta t^k + \Delta t^l)^2 \end{aligned} \quad (10)$$

The simplest algorithm would use only the first of these sums.

We could also maximize summed crosscorrelations more directly, but the non-linearity is more inconvenient:

$$\max_{\Delta t_i, \Delta t^k} \left[ C_i^k(\Delta t_i + \Delta t^k) + C_i(\Delta t_i) + C^k(\Delta t^k) + C_{ij}(\Delta t_i + \Delta t_j) + C^{kl}(\Delta t^k + \Delta t^l) \right] \quad (11)$$

The original data are corrected implicitly with these shifts:

$$d(\mathbf{x}_i, \mathbf{x}^k, t) \Leftarrow d(\mathbf{x}_i, \mathbf{x}^k, t - \Delta t_i - \Delta t^k). \quad (12)$$

We could begin the algorithm again with these corrected data, improving the ability to pick correlation peaks. In practice we need not correct the data explicitly. We only adjust the original crosscorrelations:

$$c_{ij}^{kl}(\tau) \Leftarrow c_{ij}^{kl}(\tau - \Delta t_i + \Delta t_j - \Delta t^k + \Delta t^l). \quad (13)$$

We can then recalculate the sums (6), (7), and (8) and repeat the rest of the algorithm. After convergence, the data are explicitly corrected with the accumulated shifts.