Integral operator quality from low order interpolation

or

Sometimes nearest neighbor beats linear interpolation

Stewart A. Levin

ABSTRACT

In most discussions of interpolation methods, it is the worst-case behavior that dominates the analysis. From a systems point of view, one really should analyze how that interpolation is used in producing an end product in order to determine the interpolation's suitability. In this report I look at the summation operators slant stack, NMO and Kirchhoff migration as the "systems" and determine that their output quality can be significantly better than the traditional take on interpolation would suggest. In one scenario, I even found nearest neighbor interpolation did the job even better than linear interpolation.

INTRODUCTION

Many geophysical imaging and analysis tools involve summation over various trajectories within their input data. It is rare indeed that such summations do not involve intermediate interpolation among adjacent or nearby samples. Such interpolations are generally imperfect and so can degrade the final result—a subject of interest to me for some time (Levin, 1994). In a recent broad brush presentation of Kirchhoff time migration to an introductory C++ class at Stanford, I presented the slide:

Interpolation

- We chose linear for speed
  - best case is perfect (directly on a sample)
  - worst case is halfway; good to ≈ ½ Nyquist
  - average case good to ≈ ¾ Nyquist

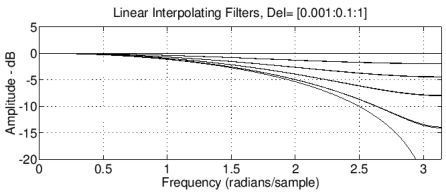


Figure by Julius Orion Smith III

Nyquist is highest frequency representable in discrete sampling. Seismic data has useful frequency content up to ≈ ⅔ Nyquist or less.

and asserted

> "Normally, an interpolator is selected based on its guaranteed-to-be-no-worse-than fidelity. However in the present case the distribution of many thousands or even millions of fractions of a sample to interpolate is uniform and so I focused in on the average rather than the worst case. From this perspective, looking at the amplitude behavior halfway between the 3rd and 4th curves plotted in the slide, the quite inexpensive linear interpolation does a good enough job in retaining fidelity. A similar plot of phase accuracy yields the same conclusion."

To understand this concept, let us take the very simple case of a single function replicated with linear delay and sampled with unit spacing as illustrated in Figure 1. Applying linear moveout and stacking would ideally reproduce that function. In order to apply the linear moveout we need to interpolate. Let us first consider what happens if we use simple nearest neighbor interpolation.

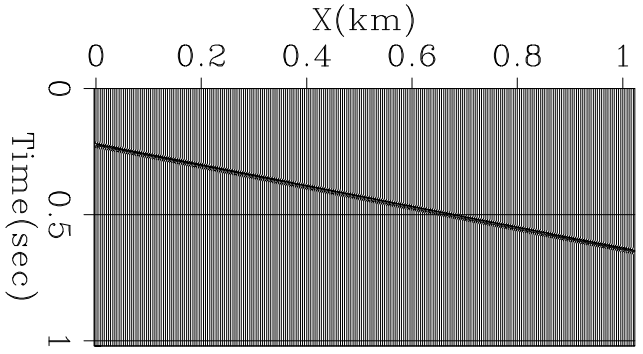


Figure 1: Simple dipping synthetic section. [**ER**]

We generally abhor nearest neighbor interpolation because it (a) is a discontinuous function of position and (b) can yield interpolated values that don't even have the correct sign. When stacking is included, though, to a good approximation we may assume the interpolation points are randomly distributed between samples following a uniform distribution. This means that the output values on our stack at x_0 are approximately

$$\int_{x_0-1/2}^{x_0+1/2} f(x)dx$$

which is convolutional filtering of $f(x)$ by a boxcar having Fourier transform

$$F(\omega) = \text{sinc}(\omega/2) \quad . \tag{1}$$

Figure 2 illustrates the filter and its spectrum. Note that within the Nyquist limits $-\pi$ to π, the spectrum (1) is positive, indeed greater than or equal to $2/\pi$. Therefore we can compensate for the nonflat spectrum by simple zero-phase spectral rebalancing after the stack, avoiding the expense of costly high quality sinc-like convolutions before stack.

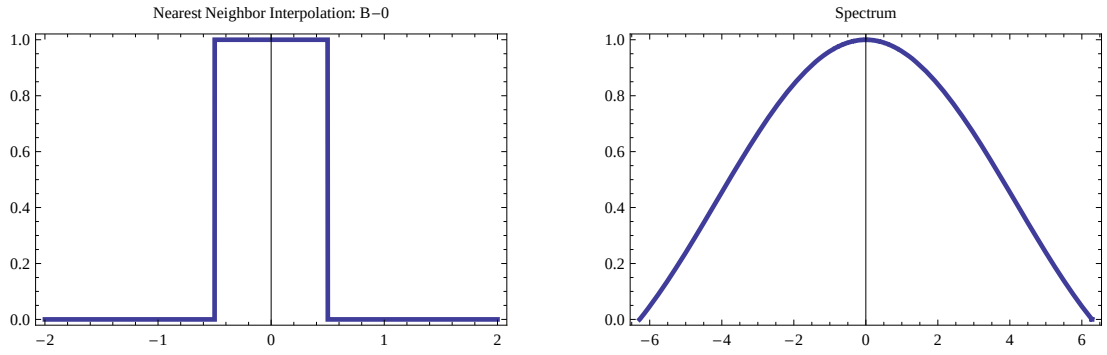


Figure 2: Nearest neighbor boxcar filter (left) and its spectrum (right). Figure from Fomel (2000). [**NR**]

Seeing that nearest neighbor interpolation works unexpectedly well, let us turn our attention to linear interpolation. Following the same line of reasoning leads to a triangular convolution illustrated in Figure 3 which is a convolution of the nearest neighbor boxcar with itself and hence a spectrum that is the square of (1). This linear interpolation result is *poorer* than nearest neighbor, requiring more post-stack spectral compensation and concomitant noise magnification at higher frequencies!

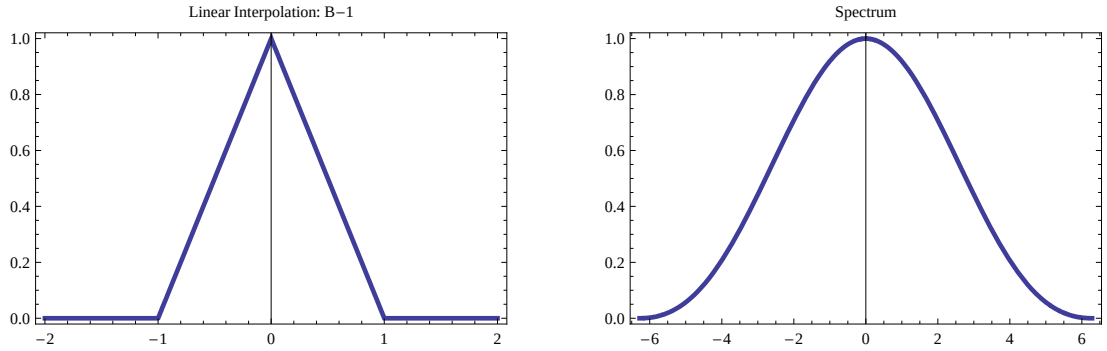


Figure 3: Nearest neighbor boxcar filter (left) and its spectrum (right). Figure from Fomel (2000). [**NR**]

EXAMPLES

For my first example I created a dipping planar synthetic with a Ricker wavelet on it. (Figure 4) This 26.556° dip is at an azimuth of 67.44° and the central frequency of the Ricker wavelet is 64.22 Hz, slightly more than half the Nyquist frequency. The wavelet was imposed by calculating the analytic Ricker formula directly onto the trace samples and not by post-convolving a spike trace synthetic. I then did an areal slant stack along the same dip and azimuth as the plane with the output at the location of

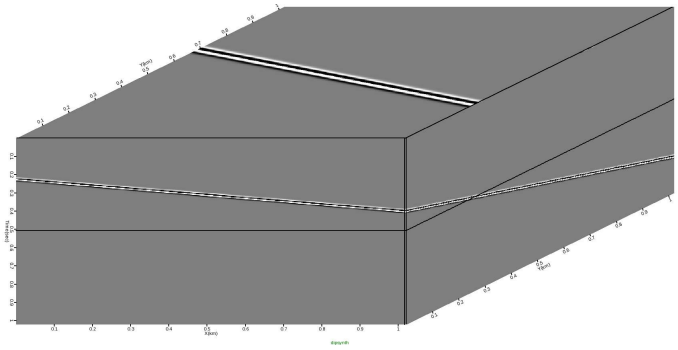


Figure 4: Dipping 3D synthetic used for slant stack test. [**NR**]

the first inline and crossline trace. I used nearest neighbor interpolation for this slant stack. Figure 5 shows the result overlain with what a perfect slant stack would have produced. Quite a respectable result even without spectral reshaping. (When spectral compensation is included, the result of nearest neighbor interpolation is nearly perfect as shown in Fig. 7.) Figure 6 shows the corresponding output for linear interpolation which, indeed, is not as accurate, agreeing with the analysis above.

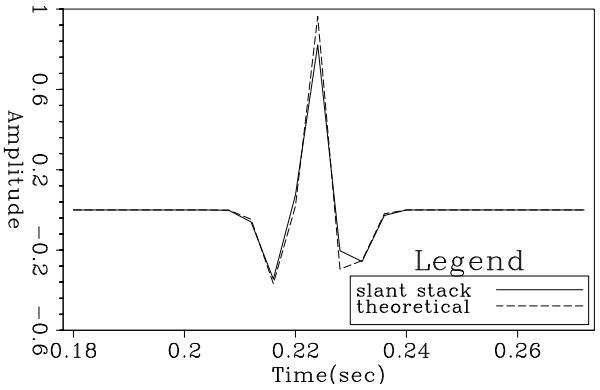


Figure 5: Comparision of the perfect theoretical result with the output of slant stacking the data in Fig. 4 parallel to the planar event using nearest-neighbor interpolation. [**ER**]

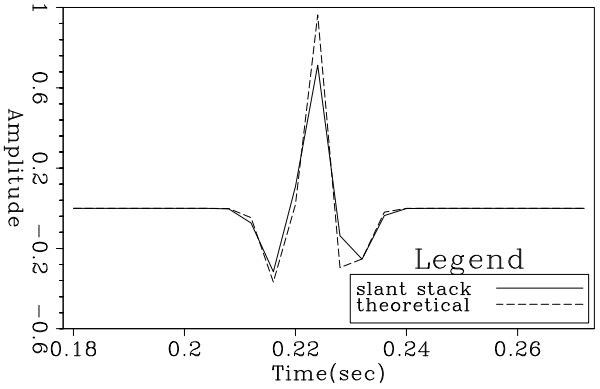


Figure 6: Comparision of the perfect theoretical result with the output of slant stacking the data in Fig. 4 parallel to the planar event using linear interpolation. [**ER**]

Of course, field data have many events, not just one, and summation operations will generally involve both constructive and destructive summation. Taking a first look at this topic, I modified the planar synthetic to have about 75 uniformly distributed individual parallel plane arrivals with amplitudes constructed according to

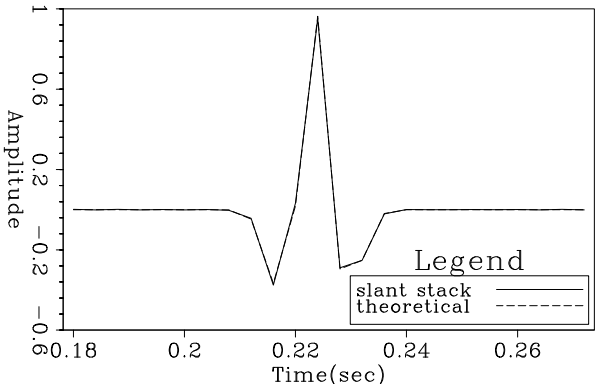


Figure 7: Comparision of the perfect theoretical result with the output of slant stacking the data in Fig. 4 parallel to the planar event using nearest-neighbor interpolation followed by spectral compensation with the reciprocal of formula 1. [**ER**]

a Gaussian distribution. (Figure 8.) Using this synthetic, I deliberately slant stacked horizontally (Fig. 10 which requires no interpolation) and again at a dip that was not parallel to the plane of the synthetic. (For Fig. 11 I opted to interchange the X and Y axis dips.) It is clear that nearest-neighbor interpolation created just about as much destructive cancellation as one would expect from a high order interpolation. Nearest

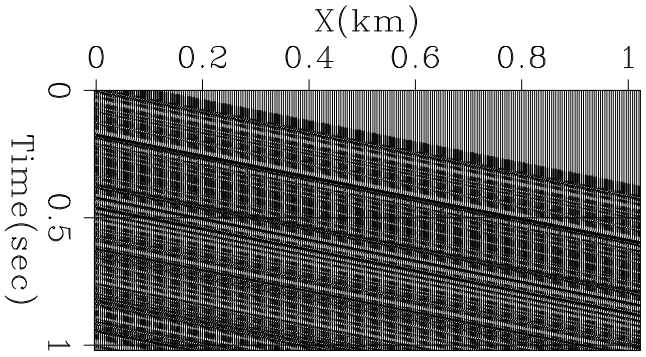


Figure 8: First slice of dipping 3D synthetic with sparse reflectors and Gaussian amplitude distribution used for slant stack test. [**ER**]

neighbor slant stack parallel to that dip yielded the expected high quality result show in Figure 9. Stacking horizontally, which requires no interpolation, or (destructive) slant stacking along a plane oriented with the X and Y axis interchanged resulted in equally small output amplitudes (a fraction of a percent) as expected.

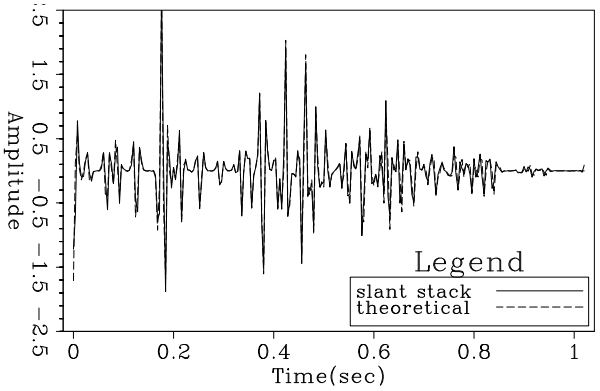


Figure 9: Comparision of nearest neighbor and high order interpolation slant stacking the data in Fig. 8 parallel to the planar event using nearest-neighbor interpolation followed by spectral compensation with the reciprocal of 1. [**ER**]

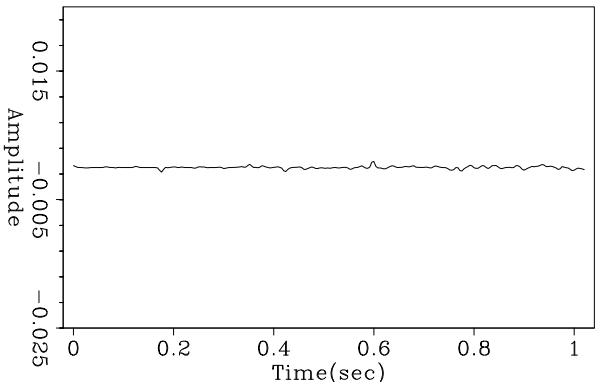


Figure 10: Horizontal stack of the data in Fig. 8 gained by a factor of 100 over that in Fig. 9. [**ER**]

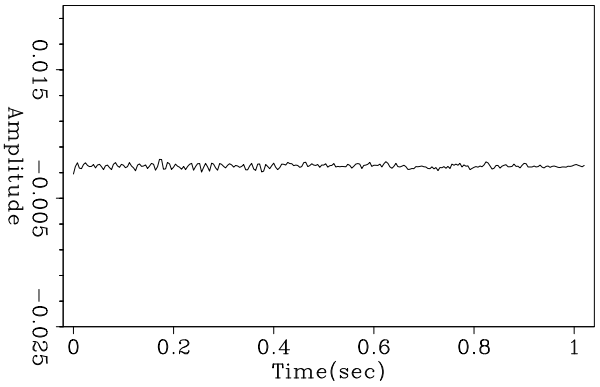


Figure 11: Slant stack over a non-parallel direction of the data in Fig. 8 gained by a factor of 100 over that in Fig. 9. [**ER**]

My second example is another situation where summation is expected to be almost exclusively constructive: 3D normal moveout. Figure 12 shows the result of applying NMO stack to a synthetic (flat dip) 3D CDP with maximum offsets of 12.8 and 25.6 km in the X and Y directions respectively. No spectral reshaping is shown because those results almost perfectly overlay their inputs. This anomaly arises because of NMO stretch which changes the shapes of the waveforms being stacked. In field data statics, deconvolution, and AVO effects will further modify the input waveforms, making spectral compensation analysis complicated and data dependent.

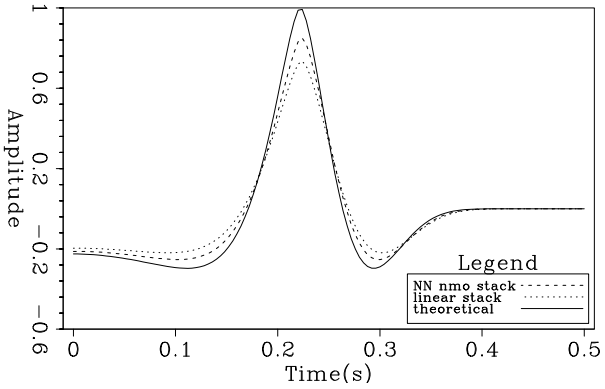


Figure 12: NMO stack comparison of nearest-neighbor, linear and high order interpolaton. Here no spectral compensation has been applied. [**ER**]

Closely related to NMO stack is common reflection surface (CRS) imaging which estimates and sums over prestack specular reflection surfaces (Jäger et al., 2001;

Hertweck et al., 2004). I generated and CRS processed a prestack synthetic for a dipping plane and got the result in Figure 13. With the exception of an apparent one sample shift, probably a small coding bug, again nearest-neighbor did a fine job and outperformed linear linear interpolation.

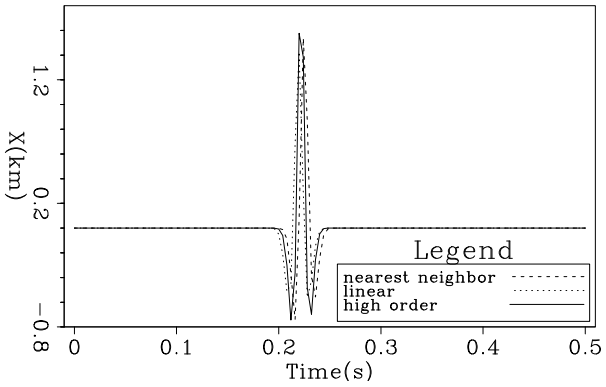


Figure 13: CRS zero-offset trace for a dipping planar synthetic using nearest neighbor, linear and high-order interpolation. [**ER**]

Transitioning further into the balance between constructive and destructive interference, for a fourth example, I poststack migrated the synthetic of Fig. 4 with the same constant velocity used to create it. Using nearest-neighbor interpolation, the output at the first inline and crossline trace location is shown in Figure 14. Linear interpolation produces Figure 15 and highly accurate interpolation produces 16. In this case, increasing the order of the interpolant does increase accuracy, however the level of residual artifact is disconcerting.

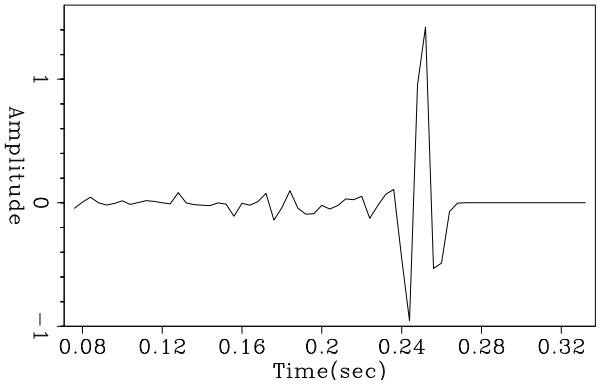


Figure 14: Poststack 3D Kirchhoff time migration of the dipping planar synthetic using just nearest-neighbor interpolation. [**ER**]

The fundamental difference between the last example and the previous ones is that poststack Kirchhoff migration sums only a few samples coherently to form an image point sample. With prestack Kirchhoff, however, the number of coherent samples forming an image point sample is much larger. Figures 17 and 18 show 3D prestack Kirchhoff migration of a horizontal planar event. (Couldn't get the dipping plane version working in time.) There is no longer annoying precursor noise in these results.

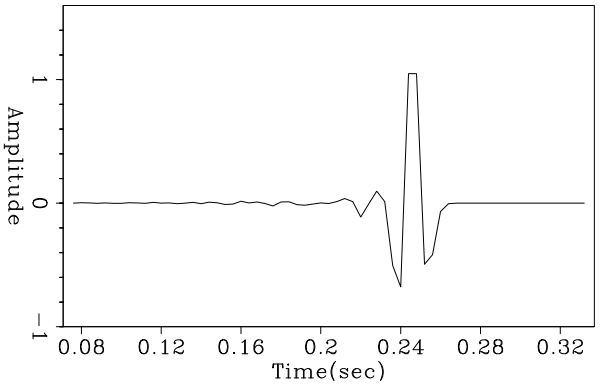


Figure 15: Poststack 3D Kirchhoff time migration of the dipping planar synthetic using linear interpolation. [**ER**]

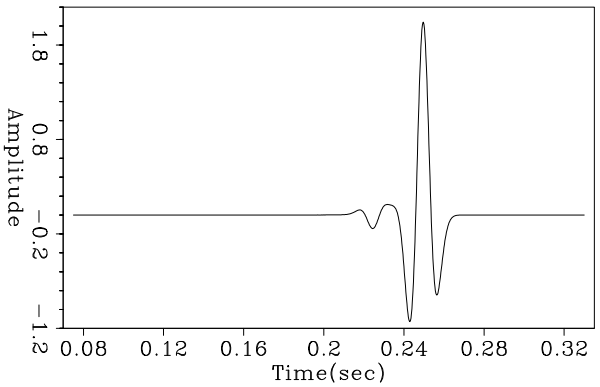


Figure 16: Poststack 3D Kirchhoff time migration of the dipping planar synthetic using high-order interpolation. [**ER**]

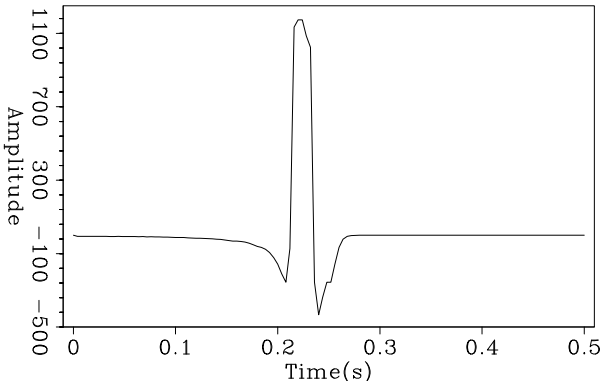


Figure 17: Prestack 3D Kirchhoff time migration of a horizontal planar synthetic using just nearest-neighbor interpolation. No attempt was made to taper aperture or apply phase corrections. [**CR**]

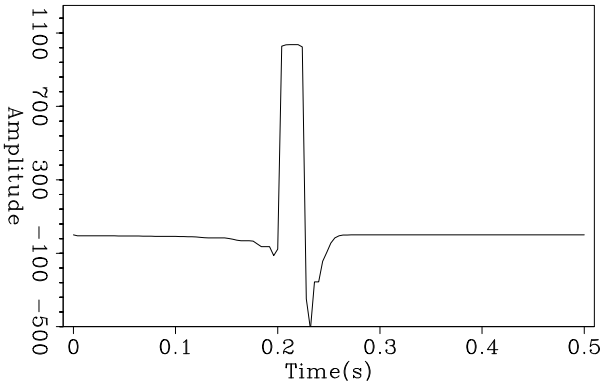


Figure 18: Poststack 3D Kirchhoff time migration of a horizontal planar synthetic using linear interpolation. No attempt was made to taper aperture or apply phase corrections. [**CR**]

DISCUSSION AND CONCLUSIONS

We have clearly seen a significant difference between slant stacking (or NMO stacking) and migration in how interpolation affects the result. This is a reflection (no pun intended) on the fact that the former is adding signals in phase while the latter is relying on massive destructive interference to suppress those data that shouldn't contribute to the output. (See, e.g., Levin (2004).) The constructive interference in the 3D poststack migration example is over a fairly small patch and has led to some unwanted noise and other artifacts. However full prestack Kirchhoff migration suppresses the noise and confirms the original conclusion that it is the average case, not the worst case, of interpolation that should be the focus of integral operator system analysis.

REFERENCES

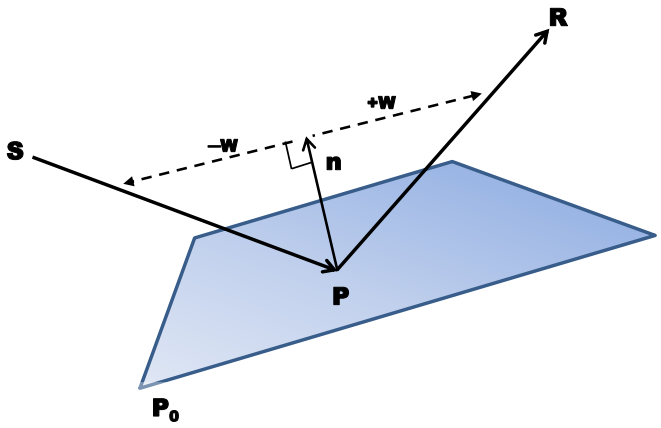
Fomel, S., 2000, Inverse B-spline interpolation: SEP-Report, **105**, 79–108.

Hertweck, T., C. Jäger, J. Mann, E. Duveneck, and Z. Heilmann, 2004, A seismic reflection imaging workflow based on the Common-Reflection-Surface (CRS) stack: Theoretical background and case study: SEG Technical Program Expanded Abstracts, **23**, 2032–2035.

Jäger, R., J. Mann, G. Höcht, and P. Hubral, 2001, Common-reflection-surface stack: Image and attributes: Geophysics, **66**, 97–109.

Levin, S. A., 1994, Stolt without artifacts? — dropping the Jacobian: SEP-Report, **80**, 513–532.

———, 2004, Numerical precision in 3D prestack Kirchhoff migration: SEG Technical Program Expanded Abstracts, **23**, 1119–1122.



Given a source location $\boldsymbol{S}$, a receiver location $\boldsymbol{R}$ and the plane $\boldsymbol{n}\cdot(\boldsymbol{P}-\boldsymbol{P_0})$=0, where $\boldsymbol{n}$ is a unit normal, to find the reflection point $\boldsymbol{P}$, drop a perpendicular $\boldsymbol{w}$ from $\boldsymbol{n}$ to the line connecting $\boldsymbol{P}$ to $\boldsymbol{R}$. Snell's Law says that running $\boldsymbol{w}$ in the other direction connects to the line between $\boldsymbol{P}$ and $\boldsymbol{S}$. So we have:

$$\begin{array}{rcl} (\boldsymbol{R}-\boldsymbol{P}) & = & \alpha(\boldsymbol{n}+\boldsymbol{w}) \\ (\boldsymbol{S}-\boldsymbol{P}) & = & \beta(\boldsymbol{n}-\boldsymbol{w}) \\ \boldsymbol{n}\cdot\boldsymbol{w} & = & 0 \\ \boldsymbol{n}\cdot\boldsymbol{P} & = & \boldsymbol{n}\cdot\boldsymbol{P_0} \end{array} .$$

Dotting $\boldsymbol{n}$ onto the first two equations gives

$$\begin{array}{rcccl} \boldsymbol{n}\cdot(\boldsymbol{R}-\boldsymbol{P}) & = & \boldsymbol{n}\cdot(\boldsymbol{R}-\boldsymbol{P_0}) & = & \alpha \\ \boldsymbol{n}\cdot(\boldsymbol{S}-\boldsymbol{P}) & = & \boldsymbol{n}\cdot(\boldsymbol{S}-\boldsymbol{P_0}) & = & \beta \end{array} ,$$

and subtracting the first two equations produces

$$(\boldsymbol{R}-\boldsymbol{S}) \quad = \quad (\alpha-\beta)\boldsymbol{n}+(\alpha+\beta)\boldsymbol{w} \ ,$$

which we can solve directly for $\boldsymbol{w}$. Given this $\boldsymbol{w}$, the first equation immediately yields

$$\boldsymbol{P} \quad = \quad \boldsymbol{R}-\alpha(\boldsymbol{n}+\boldsymbol{w}) \ ,$$

the desired reflection point.