Avoiding interpolation artifacts in Stolt migration

William Harlan

Introduction

Stolt migration requires a frequency-domain interpolation that can be the source of a great many numerical artifacts. Two common interpolators, linear and geometric, both generate strong, incorrectly migrated events which can entirely replace the correctly migrated events. Some programmers attempt to diminish the problem by padding their section again and again with zeros. One can improve on the results of endless padding by using an efficient algorithm based on the sinc function.

Again, these artifacts are not minor. Figure 1 shows the results of migrating an unpadded time section with linear, geometric, and sinc interpolators. The time section contains five "spikes," which should migrate into upward-turning semicircles. For linear and geometric interpolation, many additional semicircles appear, replacing approximately one half the energy of the correct events. The sinc-based algorithm, to be described later, takes little additional computation time and yet does not produce these foreign events. To appreciate best the assumptions implicit in the sinc interpolation, we shall first examine the sources of these bad events for the linear and geometric interpolators.

Linear Interpolation Errors

Linear interpolation in the frequency domain may be reduced to two operations. The frequency function has first been multiplied by a sampling function and has then been convolved by the triangle function. The effect of these operations is to introduce spurious time events in the time domain. To see the equivalence, assume that the point to be interpolated lies at a point $n+\delta n$, where $0 \le \delta n < 1$. Let C_n and C_{n+1} be the two adjacent samples. Then

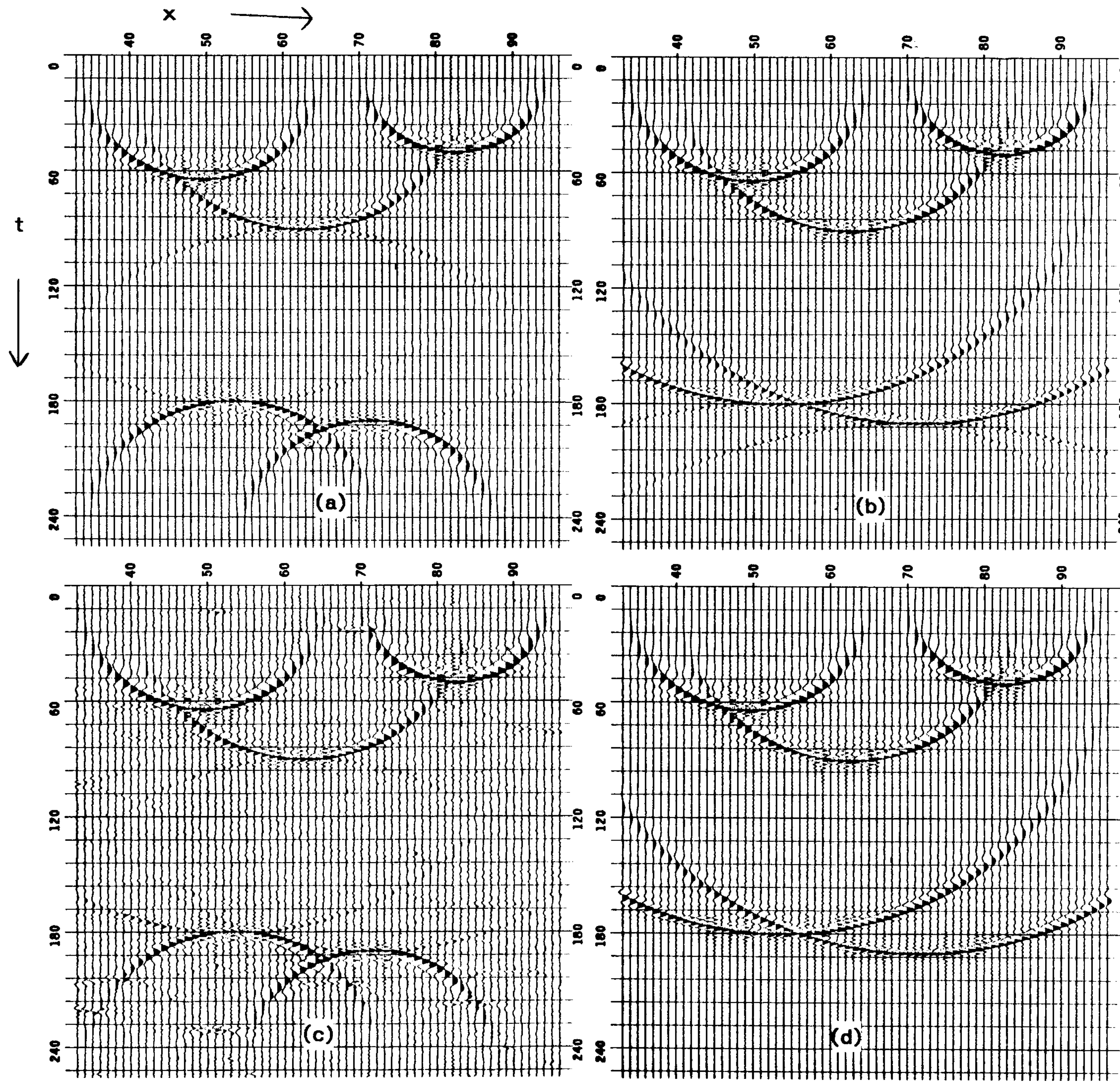

FIG. 1. Stolt migration of five "spikes" using various interpolation schemes: (a) Linear interpolation, unpadded section; (b) Linear interpolation, 100% padding; (c) Geometric interpolation, unpadded section; (d) Sinc-based interpolation, unpadded section. One expects only five upward-turning semicircles such as seen in (d). But (a), (b), and (c) include additional, incorrect, downward-turning events, with a loss in the energy of the lower, correct events. Padding with zeros reduces the energy of these incorrect events in (b) but does not remove them. The sinc algorithm (d), with no visible artifacts, still runs 25% faster than the padded algorithm (b).

linear interpolation gives

$$C_{n+\delta n} = (1-\delta n)C_n + (\delta n)C_{n+1} \tag{1}$$

Now consider the triangle function:

$$\Lambda(x) \equiv \begin{cases} 1-|x| & |x| \le 1 \\ 0 & |x| > 1 \end{cases}$$

Lay the triangle function $\Lambda(f/\Delta f)$ over the sampled function with the triangle's peak over $f=(n+\delta n)\Delta f$ as shown in Figure 2. The height of the triangle over C_n and C_{n+1} is $1-\delta n$ and δn, respectively, the same as the linear weighting factors in (1). This operation may be expressed as

$$p'(f) = \Lambda(\frac{f}{\Delta f}) * \left[\text{Ш}(\frac{f}{\Delta f}) \cdot p(f) \right] \quad \text{where } \text{Ш}(x) \equiv \sum_{n=-\infty}^{+\infty} \delta(x-n)$$

The FFT gives us the sampled function. With the above operation we are constructing an approximation of the original continuous $p(f)$. In Stolt migration we simultaneously resample along a new stretched grid $k_\tau = vk_z = (4\pi^2 f^2 - v^2 k_x^2)^{1/2}$.

But what are we doing to the time section by changing $p(f)$ into $p'(f)$? We are adding a great many time events, which will be migrated just as the original ones. Taking the Fourier transforms of $\text{Ш}(f/\Delta f)$ and $\Lambda(f/\Delta f)$ and using the convolution theorem, we find that we are altering each time axis to

$$p'(t) = sinc^2(\frac{t}{T}) \cdot \left[\text{Ш}(\frac{t}{T}) * p(t) \right] \quad \text{where } sinc(x) \equiv \begin{cases} \frac{sin(\pi x)}{\pi x} & x \ne 0 \\ 1 & x = 0 \end{cases}$$

$T = 1/\Delta f$ is the length of the time axis. Figure 3 illustrates what happens. The convolution by $\text{Ш}(t/T)$ replicates the time section at time intervals of T. The waveform values for times $t_0, t_1, \ldots, t_N$ are duplicated at times $nT+t_0$, $nT+t_1, \ldots, nT+t_n$ for all integer n. The multiplication by $sinc^2(t/T)$ attenuates most of these extra events, but a very poor selection of them. Good events falling in the interval $T/2 \le t \le T$ are actually attenuated more than the unwanted events at negative times $-T/2 \le t \le 0$. When we return to the τ domain, we find that these unwanted events lie over our correct events but with incorrect curvatures and orientations. The downward-turning events near the bottom of the section are stronger than the correct upward-turning events. At higher gain, many other semicircles corresponding to greater times and depths can be seen.

One may improve the attenuation of these unwanted events by a "demodulation" before interpolation. Multiplying the frequency axis by $e^{-i\pi f/\Delta f}$ before linear interpolation effectively shifts the peak of the $sinc^2$ over the time events at $T/2$: the greatest migrated

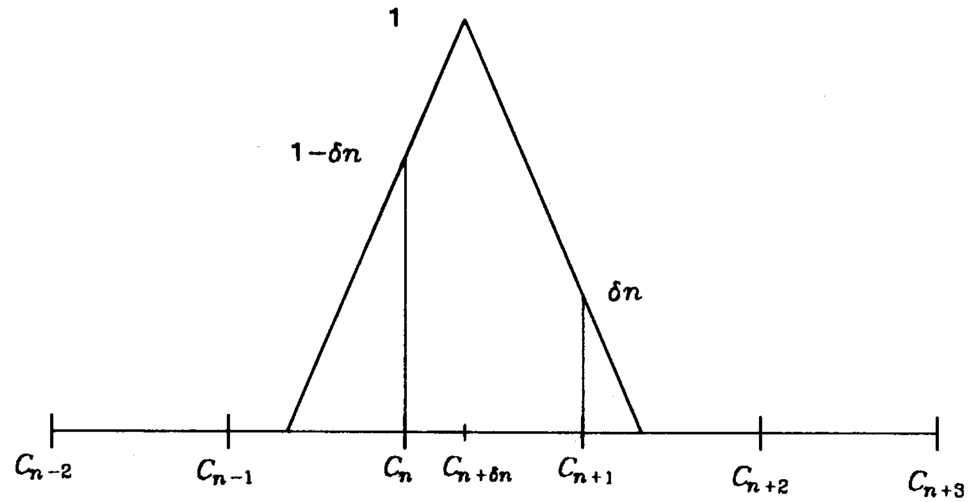

FIG. 2. Linear interpolation of sampled data is equivalent to convolution by a triangle function. The samples are indicated by C_n's, and the interpolated point by $C_{n+\delta n}$. The two adjacent points, C_n and C_{n+1}, receive the linear weighting factors of $1-\delta n$ and δn, the respective heights of the triangle over the sample points.

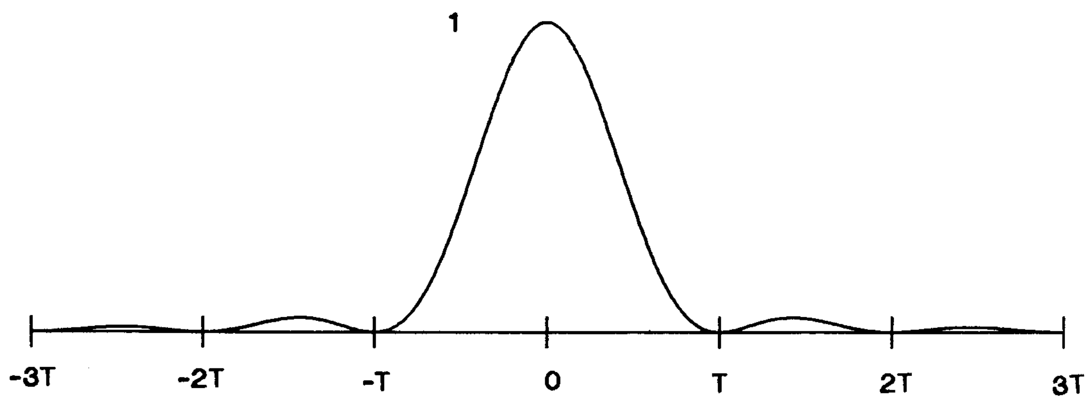

FIG. 3. Linearly interpolating in the frequency domain equivalently multiplies the time domain by the above $sinc^2(t/T)$ function. Sampling in the frequency domain first replicates the time axis, $0 \le t \le T$, at intervals of T. The multiplication by the $sinc^2(t/T)$ leaves much of this replicated information, which appears after migration as artifacts. In addition, much of the original, correct information is greatly reduced. One prefers to multiply by a near rectangle such as in Figure 5.

energy will now be from the correct events. However, correct events away from $T/2$ and toward 0 and T now will be strongly attenuated. One desires instead of the $sinc^2$ an attenuating function which is as flat as possible in the region $0 \le t \le T$. The demodulated linear interpolator poorly satisfies this condition. In fact, a two-point interpolator based on the sinc greatly improves the results.

Geometric Interpolation Errors

The geometric interpolator is defined by

$$\ln(C_{n+\delta n}) = (1-\delta n)\ln(C_n) + (\delta n)\ln(C_{n+1})$$

By taking logarithms, we unwrap circles in the complex f domain into vertical lines. If $p(f)$ can be approximated well by successive points on a complex circle, then we may interpolate the unwrapped $\ln p(f)$ linearly. Specifically, the sampled values of $p(f)$ must allow the approximation

$$C_n = Ce^{-in\Delta\varphi} \text{ for } 0 \le \Delta\varphi \le \pi \tag{2}$$

The trouble is that for an arbitrary $p(t)$, $p(f)$ may behave nothing like this exponential function. If $p(w)$ poorly resembles a circle, then we are linearly interpolating a non-linear $\ln p(f)$, adding additional events in the time domain just as for linear interpolation.

Geometric interpolation was originally proposed because the above relationship exactly suits the transform of a single spike at time t_0:

$$p(t) = \delta(t-t_0) \supset p(f) = e^{-i2\pi f t_0}$$

When the upper half of the section contains more than one spike, however, $p(f)$ begins to look very unlike a complex circle. Assume two spikes are at times t_1 and t_2. Then

$$p(t) = \delta(t-t_1)+\delta(t-t_2) \supset$$

$$p(f) = e^{-i2\pi f t_1} + e^{-i2\pi f t_2} = 2\cos\left[\pi f(t_1-t_2)\right]\cdot e^{-i\pi f(t_1+t_2)} \tag{3}$$

Here we have our desired exponential, as for (2), but with an important cosine modulation added. As long as t_1-t_2 is small, this modulation will have a much lower frequency than the exponential cycle, and the circular approximation will be fairly good. But assume that t_1 is substantially smaller than t_2. Then this cosine oscillates nearly as rapidly as the exponential, and the error of approximating a circle will be very substantial. Figure 4 shows the effect of increasing the number of spikes from one to two in the input time section of the geometric-interpolation migration.

If we look for a general relationship, we find that the exponential of equation (3) appears because our time section is centered away from zero. We can shift a function centered at zero time to one centered at positive time by convolving with a delta function. That is,

$$p(t) = p(t+T_{center}) * \delta(t-T_{center})$$

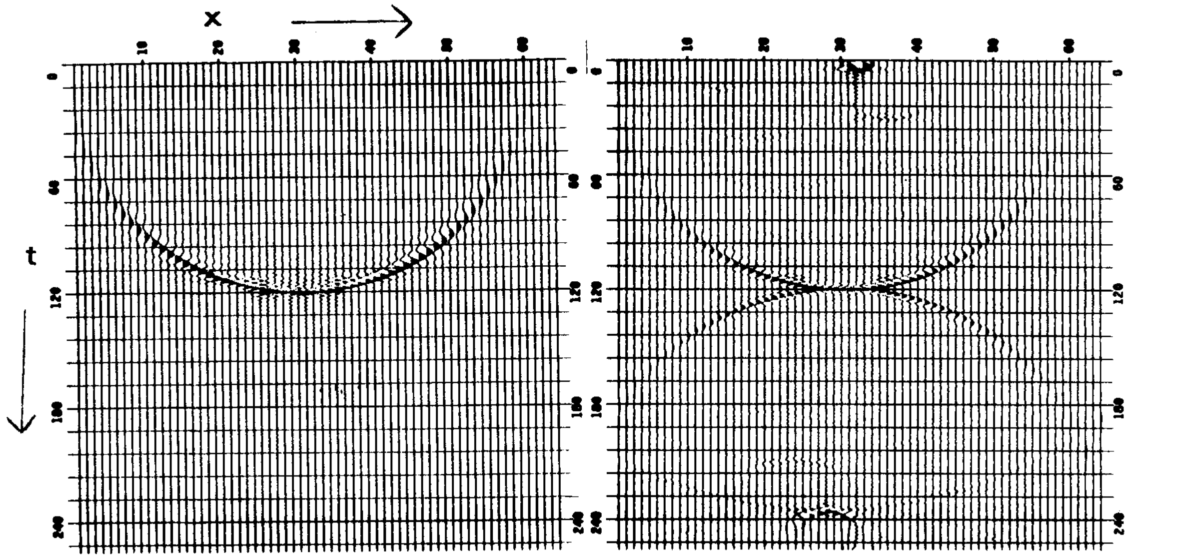

FIG. 4. Geometric interpolation, though ideal for migrating a single "spike" (left), produces strong downward turning artifacts when a second "spike" is added near the top of the right section. Geometric interpolation must fail for an arbitrary superposition of spikes, that is, for an arbitrary sampled function.

where $p(t+T_{center})$ is our time function centered about zero. In the frequency domain this convolution is the equivalent of multiplying by $\exp(-i2\pi f\, T_{center})$, the exponential we were looking for. The previous cosine modulation of (3) is just the Fourier transform of two spikes centered about zero time. Since $p(t)$ can be very arbitrary, so can the modulation of the exponential. Even if one were to pad indefinitely, strong artifacts such as appear in Figure 4 could not be avoided.

Sinc Interpolation

By examining the problems of the linear and geometric interpolators, we understand more clearly what an ideal interpolator should accomplish. Because the frequency domain is sampled, an event at time t_0 in our time section is replicated at an infinite number of positions $nT+t_0$ for all integers n. The replicated events begin with the same strength as the original ones and must be attenuated.

The act of interpolation amounts to a convolution by some chosen function and then a resampling in a new distorted grid -- k_τ in our case. This chosen function should have an inverse Fourier transform which will remove as many of the unwanted events as possible. In addition, one should not attempt to predict the functional form of $p(f)$, if one desires $p(t)$ to be as arbitrary as possible.

If we multiply the replicated time function by the following rectangle function

$$\Pi\left[\frac{t-T/2}{T}\right] \quad \text{where } \Pi(x) = \begin{cases} 1 & |t| < 1/2 \\ 0 & |t| > 1/2 \end{cases}$$

then we suppress all the unwanted events and leave the correct ones, for $0 \le t \le T$, at full strength. In the frequency domain this operation amounts to convolution by the following:

$$\frac{1}{\Delta f} e^{-i\pi \frac{f}{\Delta f}} sinc\left(\frac{f}{\Delta f}\right) \equiv \frac{1}{\Delta f} e^{-i\pi n} \, sinc(n)$$

Expressing this convolution in terms of the interpolated point, we get

$$C'_{n+\delta n} = \sum_{m=0}^{N-1} C_m e^{-i\pi[(n+\delta n)-m]} sinc[(n+\delta n)-m]$$

$$= \frac{1}{\pi} e^{-i\pi\delta n} \sin(\pi\delta n) \sum_{m=0}^{N-1} C_m / [(n+\delta n)-m] \quad \text{(cf. Rosenbaum)}$$

In this form all samples contribute something to the interpolated value. Assuming that the programmer would prefer to use fewer samples for each interpolated value, we may taper this sequence to a few terms about $n+\delta n$. An easily applied taper is the triangle function. A ten point sinc interpolator would have the form

$$C'_{n+\delta n} = \frac{1}{6\pi} e^{-i\pi\delta n} \sin(\pi\delta n) \sum_{m=-4}^{5} C_{n+m} \cdot (6-|\delta n - m|)/(\delta n - m) \qquad (4)$$

This tapering by a triangle function means that we are no longer attenuating with a perfect rectangle in the time domain. Instead, the rectangle is first rounded by convolution with a $sinc^2$ function whose narrowness depends directly on the broadness of the triangle function. Figure 5 displays the appropriately rounded rectangle for a triangle enclosing ten sample points in the frequency domain. This function remains very flat over the correct time events and drops sharply to small values over the incorrect events; it preserves the relative strengths of the correct events but virtually eliminates replications. A slight amount of padding, say 5%, will move data from under the rounded corners of the rectangle, reducing the number of points necessary for interpolation.

Padding the data of a linearly interpolating program 100% with zeros more than doubles our run time, whereas using a ten point sinc interpolator increases the run time by only a factor of 1.5. Even after padding, the linearly interpolating program still produces strong artifacts on the lower events (Figure 1). The artifacts of the ten-point sinc interpolator are visible only under a very high gain.

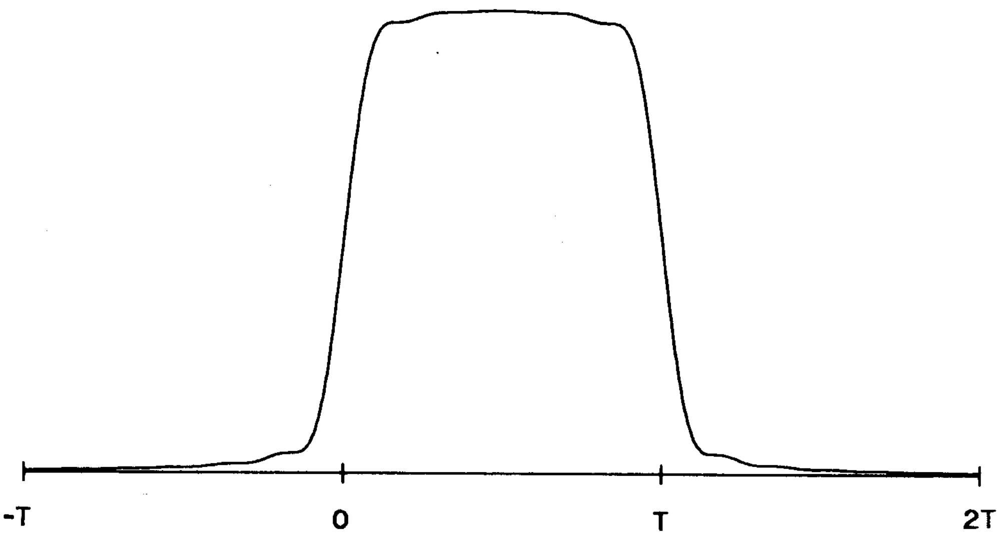

FIG. 5. Interpolating the frequency axis with the 10 point sinc interpolator of equation (4) equivalently multiplies the time domain by the above rounded rectangle function. Because the frequency axis is sampled, the time axis $0 \leq t \leq T$ is replicated at the adjacent intervals. The above function largely removes the effect of these replications and leaves all correct events at nearly equal strength. This selection greatly improves the results of linear interpolation (Figure 3) which leaves many incorrect events at greater strength than correct ones.

Conclusions

Linear and geometric interpolation both produce serious numerical artifacts when used in Stolt migration. Linear interpolation allows substantial improvement only by expensive padding of the time section. Geometric interpolation makes incorrect assumptions about the form of the time function, so even endless padding cannot help. The sinc-based interpolation reduces these artifacts to negligible magnitudes with a minimum of expense.

REFERENCES

Bracewell, R.N., 1978, The Fourier transform and its applications: New York, McGraw-Hill Book Company.

Lynn, W., 1977, Implementing f-k migration and diffraction: Stanford Exploration Project Report 11, p. 18-28.

Rosenbaum, J.H. and G.F. Boudreaux, 1981, Rapid convergence of some seismic processing algorithms: Geophysics, v. 46, p. 1667-1672.