Zero-offset prediction by polynomial interpolation

Paul Fowler

INTRODUCTION

Despite its shortcomings, the common-midpoint stack remains the workhorse of seismic processing. A stack is commonly treated as approximately representing a zero-offset section. Many assumptions are made in this approximation, since it is based on a model of point sources and detectors in an earth which is isotropic, homogeneous (or weakly stratified) and which supports only compressional waves. One type of error is primarily kinematic or geometric, and may be partly ameliorated by better velocity analysis, non-hyperbolic moveout, and dip-moveout methods. In this paper, I would like to address a different problem, namely that amplitudes of events in a stack will not in general be the same as those in a zero-offset experiment, since even if the kinematic effects of move-out are properly corrected, the amplitude of an event generally will vary with offset.

Stacking extracts the mean value in the offset direction from a series of normal-moveout corrected traces. It may be conceived of as a lateral low-pass filter; in fact, it passes exactly the d.c. component of the amplitude variation with offset. To allow for the variation which is actually observed, we need to design a filter which passes lateral wavenumber components above d.c. A filter which used the entire spectrum to laterally extrapolate, however, would probably be too sensitive to noise; one of the greatest virtues of the conventional stack is its ability to dramatically improve signal to noise ratio. The variation of amplitude with offset is ordinarily fairly smooth as long as one is not near a critical angle. We wish, therefore, to design an inexpensive lateral low-pass filter.

MOTIVATION FOR CHOOSING A POLYNOMIAL FITTING FILTER

The synthetic gather in figure 1a was constructed to show a variety of amplitude-offset effects. The modeling program used a compressional source, and the gather contains all mode converted events but no multiples. Figure 1b shows the same gather after normal moveout at the P-wave velocity, and 1c shows the moveout corrected gather without the converted events. Examining figure 1c, we can see that some events increase amplitude with offset, some decrease, and one event (at 1.42 sec) undergoes a complete polarity reversal. Figure 2 shows the values of the amplitudes for the event at 1.42 sec plotted against offset. We wish to treat such a sequence of values as samples from a continuous function, and predict the value at zero offset from its values at the non-zero offsets that are actually recorded. A conventional stack would use the mean value as the predicted zero-offset value, but for a polarity reversing event such as the one shown, the mean is near zero, and the event would effectively lost in the stack. Similarly, we can expect any event which shows substantial amplitude trend to be significantly under- or over-estimated by conventional stacking.

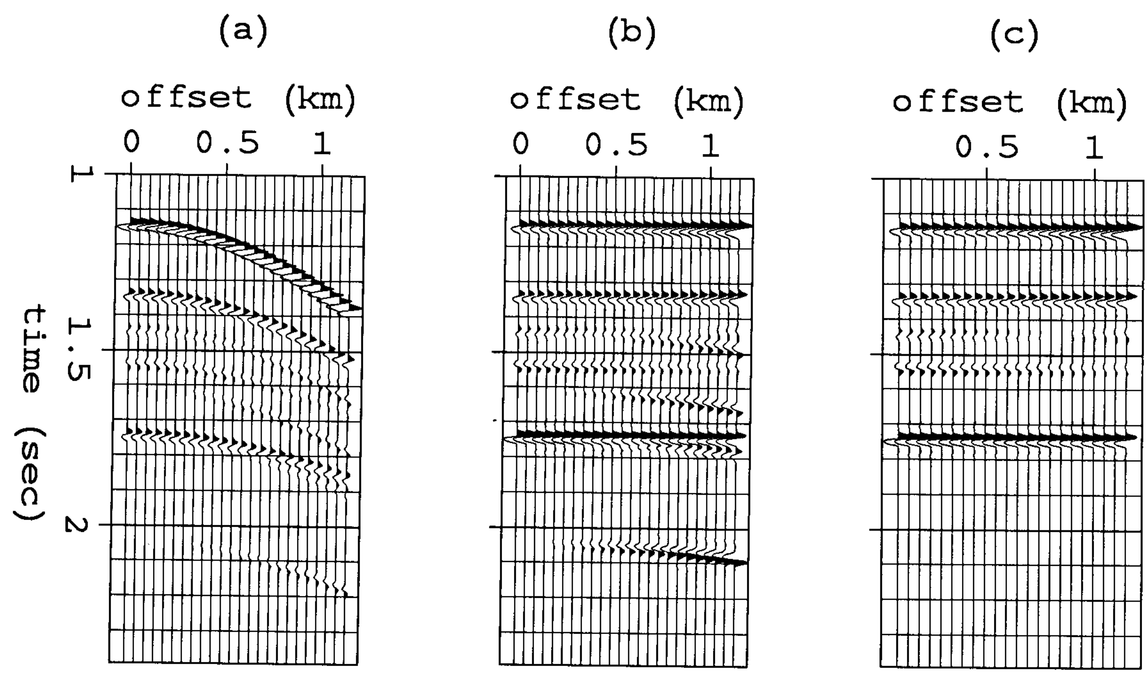


FIG. 1. (a) Synthetic model gather. The source is compressional, and all mode converted events are included, but no multiples. (b) Same model gather as (a) after NMO at compressional wave velocity. (c) Same as (b), but compressional primary arrivals only.

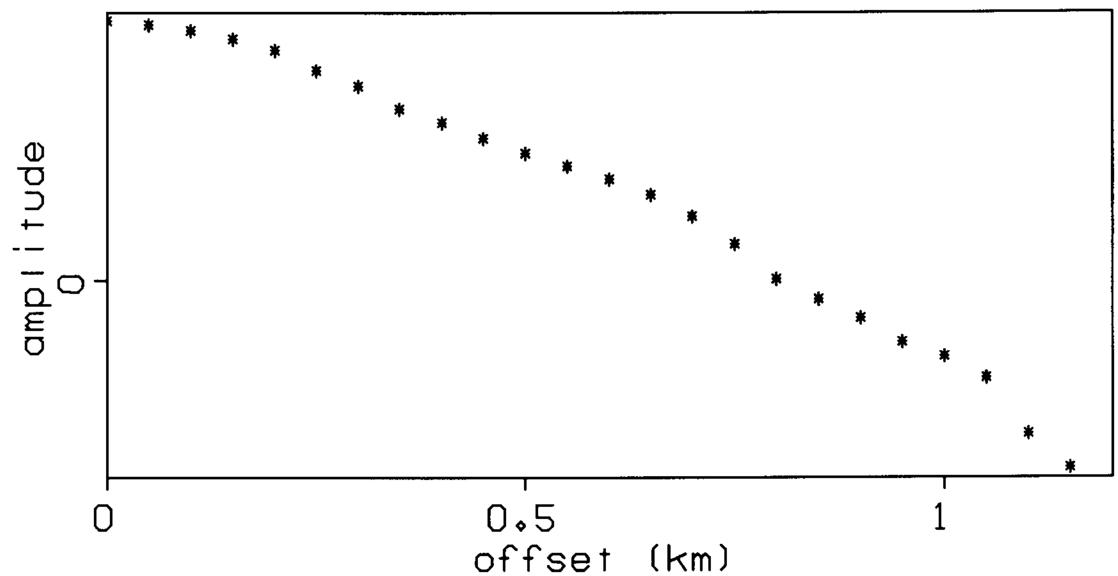


FIG. 2. Amplitude variation with offset of polarity reversing event at 1.42 seconds in model gather of figure 1(c).

Amplitude can vary with offset for a wide variety of reasons, some of which may be of considerable interest, such as the change of reflection coefficient with incident angle, and others, such as variability of geophone coupling to the ground, may contain little useful information. A full discussion of the many factors which may effect amplitude versus offset trends is given in Ostrander (1984). One approach to predicting a zero-offset trace would be to attempt to account for all of these effects by inverting for the best fitting parameters governing them, but this can be a difficult undertaking even if only a few factors are considered (Mora, 1984). Our approach here instead will be to feign ignorance about the causes of amplitude trends beyond distinguishing between low frequency trends and high frequency noise components. Rather than try to evaluate such factors as array response or reflectivity variation, we will simply extrapolate the observed values back to an estimate of what might have been recorded by a hypothetical zero-offset experiment.

Because we are not trying to deduce the causes underlying amplitude variation with offset, our primary criterion for extrapolating a value to zero-offset will be simplicity of method, rather than adherence to any underlying model. The trends in amplitude are generally fairly smooth, so we wish to limit the number of extra free parameters to be calculated. Perhaps the simplest functional form for a curve fitted to a fairly smooth set of data points is a polynomial. We describe here the results of experimenting with several different polynomials. A criterion also needs to be specified for fitting such a curve to the data. We have used least-squares here, principally because explicit solutions may be found, thus saving a great deal of computational effort. For some purposes, a different fitting norm such as L_1 might prove more desirable; for a detailed examination of this subject see Woodward (1985).

AVOIDING REPEATED SOLUTION OF THE NORMAL EQUATIONS

The simplest polynomial is just a constant. Least squares fitting of a constant to data is just calculating the mean, so the zero-order case of our proposed polynomial fitting reduces to normal stacking. Suppose now that we have N data points and wish to fit a polynomial through them. We will use the case of a parabola as a demonstration; other order polynomials are solved entirely analogously. Let y_i be the amplitudes recorded at the different offsets x_i, and let $p_2(x)=c_0+c_1x+c_2x^2$ be the parabola we are fitting to the data y_i. Then the normal equations to be solved can be written as (Claerbout, 1976):

$$\left\{ \sum_{i=1}^{N} \begin{bmatrix} -y_i \\ 1 \\ x_i \\ {x_i}^2 \end{bmatrix} \begin{bmatrix} -y_i & 1 & x_i & {x_i}^2 \end{bmatrix} \right\} \begin{bmatrix} 1 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

or

$$\sum_{i=1}^{N} \begin{bmatrix} [-y_i]^2 & -y_i & -x_i\, y_i & -{x_i}^2 y_i \\ -y_i & 1 & x_i & {x_i}^2 \\ -x_i\, y_i & x_i & {x_i}^2 & {x_i}^3 \\ -{x_i}^2 y_i & {x_i}^2 & {x_i}^3 & {x_i}^4 \end{bmatrix} \begin{bmatrix} 1 \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Such a set of normal equations can be solved by standard techniques for the polynomial coefficients c_0, c_1 and c_2. It is worth noting that because the polynomials $1, x, x^2$, etc are far from orthogonal, this system of equations is highly unstable. This instability and methods of avoiding it using a basis of Legendre polynomials are discussed in most numerical analysis texts (see, for example, Atkinson, 1978). In our case, however, we can avoid most of the numerical instability, and most of the computational effort as well, by explicitly solving the equations once and for all. Even for low order polynomials the algebra involved can be fearsome, but such problems are readily treated by computer symbolic algebra programs.† The expressions for the coefficients will contain terms of the

† I used the program MACSYMA, a powerful symbolic mathematical processor developed at M.I.T. and currently available from Symbolics, Inc. of Cambridge, Mass.

form $\sum x_i^j$ which depend only on the offset geometry and need be calculated once only, and weighted sums of the form $\sum x_i^n y_i$, which entail little cost above that of computing the unweighted sum, as is done in conventional stacking. It is important to note that only the leading coefficient in any polynomial needs to be calculated if only the value at zero offset is required. If we want to interpolate non-zero offsets, we will need to calculate and compute more constants, but as these constants depend only on the recording geometry, they need only be calculated once for an entire line, adding only trivially greater overhead. If a high order polynomial were desired, even the explicit algebraic solution might lead to numerical troubles, but here we used no terms higher than x_i^{12} and encountered little trouble.

Using interpolating polynomials to extrapolate off the end of the data can have give quite erroneous answers if no further constraints are included. Francis Muir suggested the following reasoning, which leads effectively to a strong constraint on the behavior at the missing inner offsets. By reciprocity, we can construct a symmetric, two-sided midpoint gather, where negative offsets are interpreted as reciprocal experiments with the shot and geophone reversed. The extrapolation problem then becomes one of interpolation, and our interpolating function must be an even function. If we additionally require that the interpolating function be differentiable at zero offset, which is certainly physically reasonable, then any interpolating polynomial must contain only even power terms. That is, perhaps we should use only polynomials in x^2 rather than in x .

Several polynomials were tried experimentally. General polynomials including odd order terms will be designated by p_j and polynomials containing only even terms will be q_j , where in both cases j is the power of the highest order term. Note that $p_0 = q_0$ is the conventional stack. Explicit solutions for the various forms of the leading coefficient c_0 which were used for calculating the zero-offset value are given in the appendix.

SYNTHETIC DATA EXAMPLES

Figure 3 again shows the amplitude values for the event at 1.42 seconds, as in figure 2. The best fitting curves for the polynomials p_0, p_1, p_2, p_3, q_2, and q_4 are also shown. The data values are shown for all offsets, but the inner two were not used in computing the fitting polynomials. The mean (p_0) does a poor job of predicting the correct zero-offset value; the other polynomials all do much better.

Figure 4 shows the interpolated zero-offset traces calculated with each of the polynomials from the p-wave data of figure 1c. For clarity, each trace is repeated 6 times. The modeled zero-offset trace is shown similarly at the left. Again, to better simulate

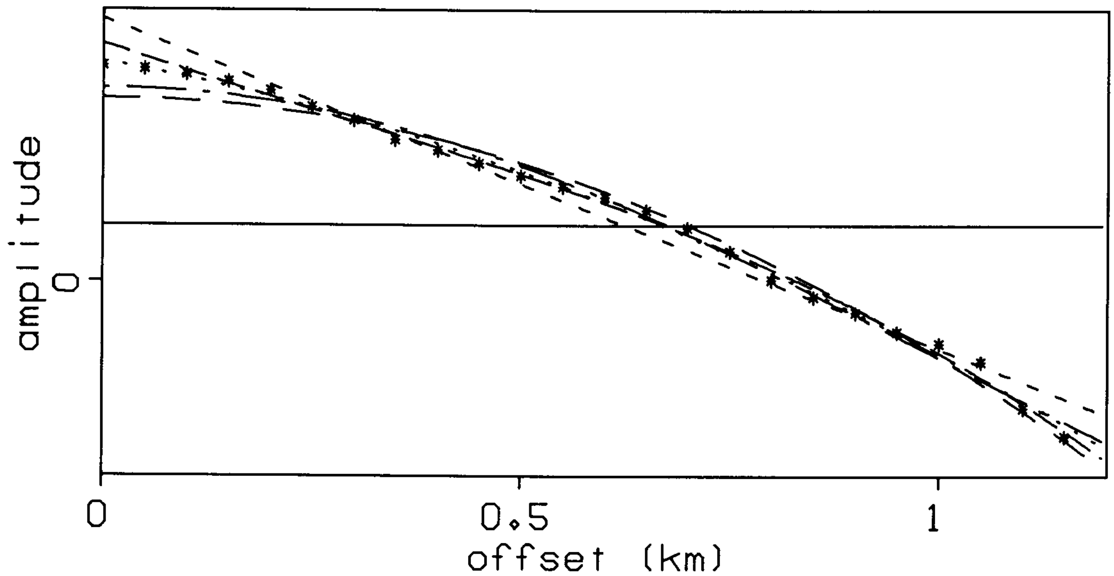


FIG. 3. Amplitude values at each offset for the event at 1.42 seconds in figure 1c. The stars represent the modeled amplitude values. The various lines represent different polynomial fitted by least squares criteria. The curves are, in descending order of their zero-offset intercept values, p_1, p_3, p_2, q_4, q_2, and p_0. The last polynomial discussed in the text, q_6, is not plotted because for this example it was virtually indistinguishable from q_4. The mean p_0 which is used for conventional stacking severely underestimates the zero-offset amplitude; all of the higher order polynomials produce better estimates.

field data, the inner 2 offsets were not used in computing the polynomial stacks. Figure 5 shows the error made by each polynomial. The modeled zero-offset trace is shown first, followed by the difference between it and each of the polynomial predictions. The mean stack p_0, as expected, makes noticeable errors in the amplitudes of most events. The linear trend fit, p_1, also makes visible errors. The higher order polynomials all make only insignificant errors.

We have considered so far the zero-offset traces predicted by interpolating the compressional wave data. Large changes in amplitude with offset, however, are likely to be accompanied by significant conversion of energy into shear waves. Figures 6 and 7 show the zero-offset traces and their errors, just as in Figures 4 and 5, but now based on the data of Figure 1b, including all shear converted energy. All of the stacks incorrectly show the event at 2.05 seconds, which is strong at wide offsets but dies to zero at vertical incidence. Most obvious is that the p_j polynomials all severely overshoot the amplitude of this event; the quadratic polynomials q_j are comparable to the stack. The polynomials, especially the q_j, retain their better representation of the amplitudes of the

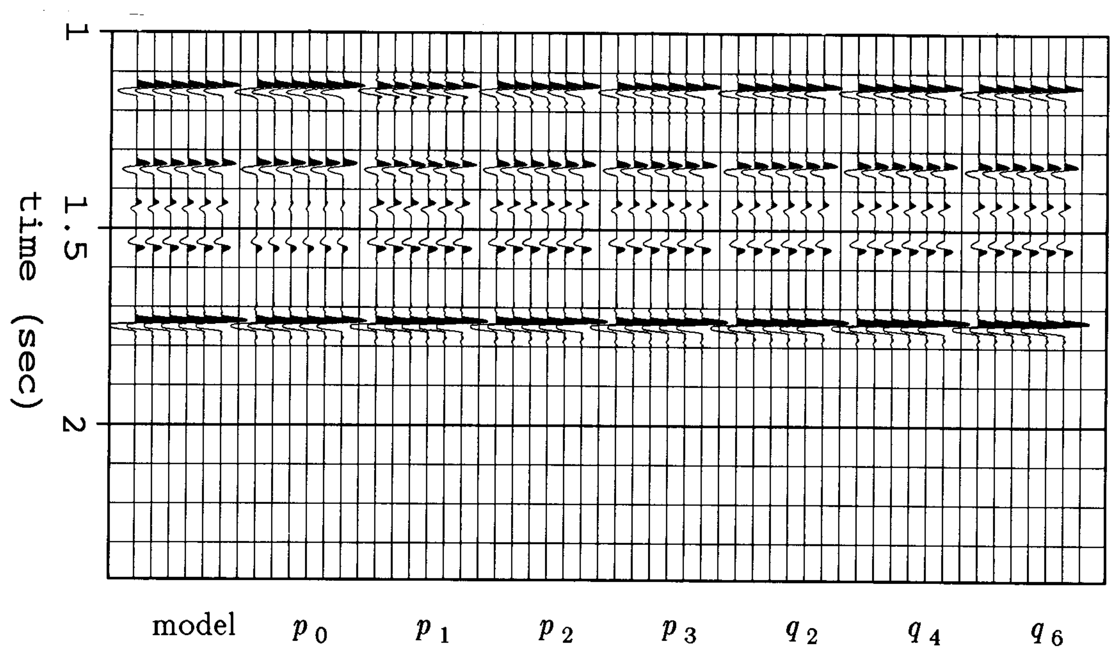


FIG. 4. Zero-offset traces for compressional data of figure 1c. The modeled zero-offset trace is at left, followed by the traces predicted by each polynomial interpolator. For clarity, each trace is repeated 6 times.

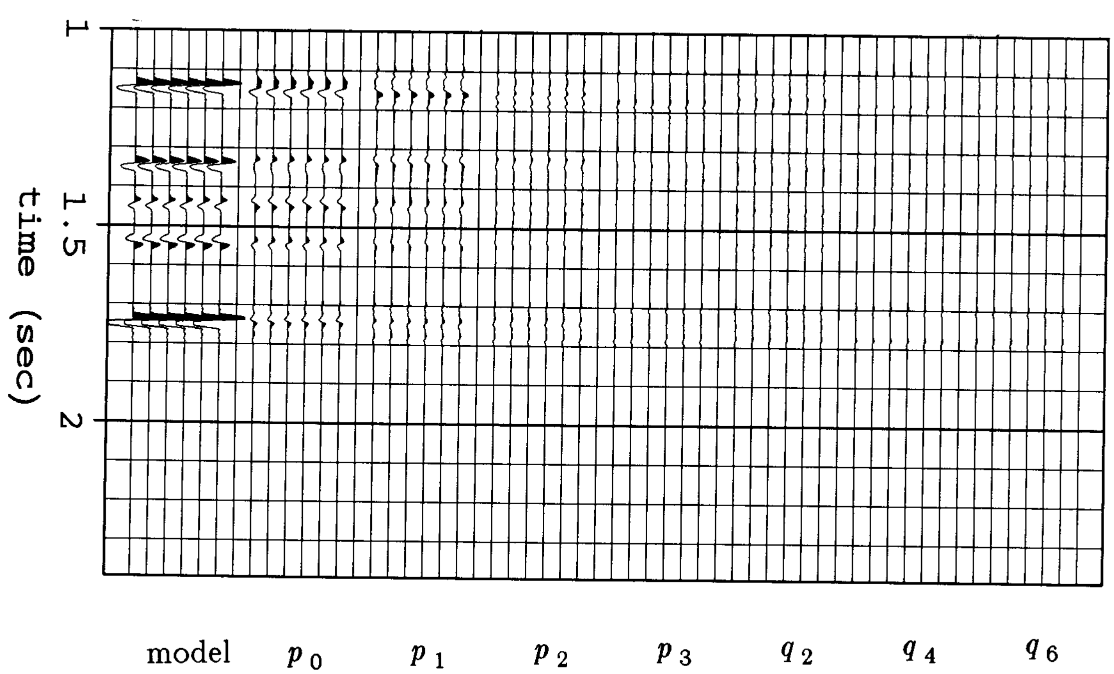


FIG. 5. Errors of zero-offset traces of figure 4. The modeled zero-offset trace is at left, followed by the difference between the model trace and each of the traces predicted by the polynomial interpolators.

earlier events.

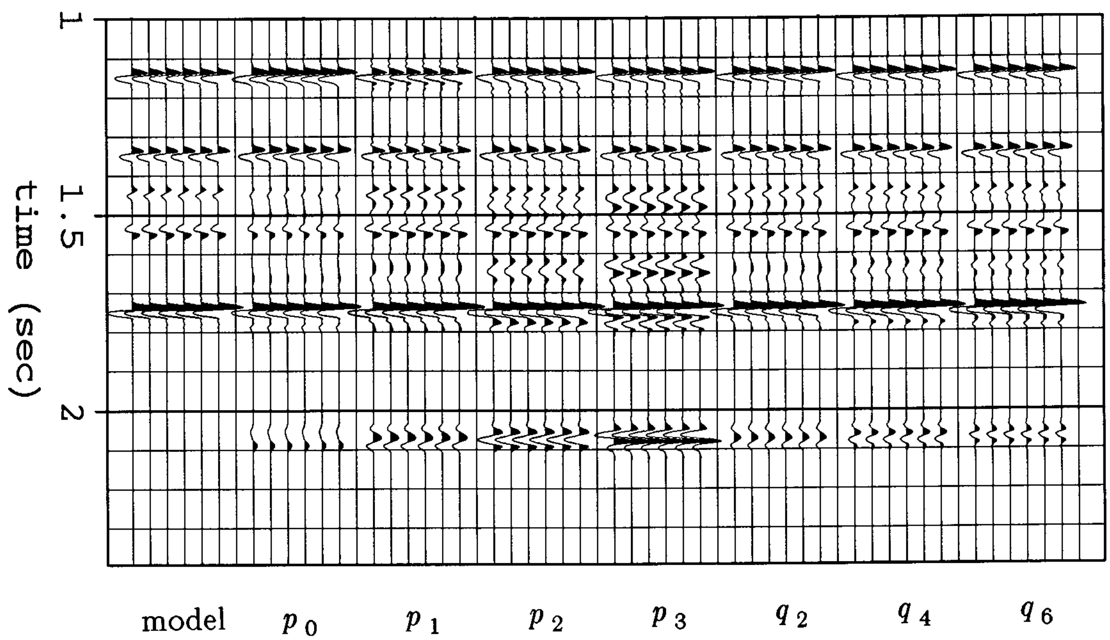


FIG. 6. Zero-offset traces for compressional and shear converted data of figure 1b. The modeled zero-offset trace is at left, followed by the traces predicted by each polynomial interpolator. For clarity, each trace is repeated 6 times. Compare with Figure 4, which shows only the compressional data.

One of the strongest virtues of the conventional stack is the dramatic improvement in signal-to-noise ratio it can produce. Since the polynomial interpolators discussed here also use an L_2 fit, we can expect them to be similarly successful at discriminating against noise. Because they pass a broader lateral bandwidth, we can expect them to degrade faster at high noise levels, however. Figure 8 shows the data of Figure 1c with a moderate amount of Gaussian noise added. Figures 9 and 10 are the counterparts of Figures 6 and 7, showing the zero-offset traces and their errors. As expected, the higher order polynomials are more sensitive to noise than the lower order ones. Note especially q_2, which retains low error level. Further tests with higher noise levels confirmed these results: the q_j consistently outperform the p_j, and the higher the order of the polynomial, the more sensitive to noise.

We have not explicitly demonstrated the effects of coherent noise, but it is easy to know what to expect. Noise such as ground roll which appears at high amplitudes at

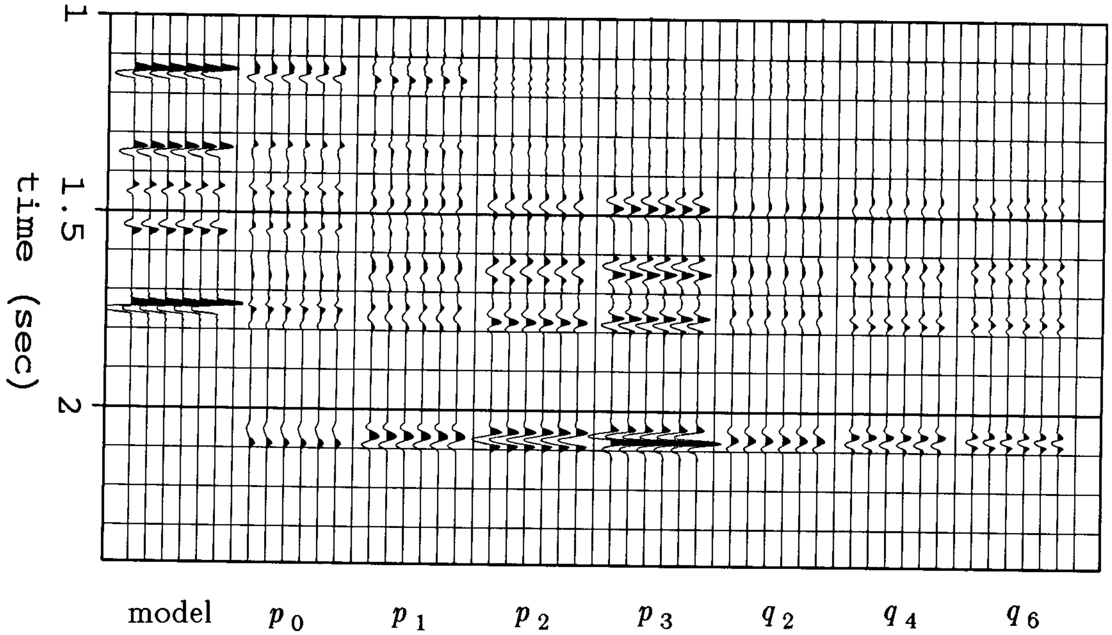


FIG. 7. Errors of zero-offset traces of figure 6. The modeled zero-offset trace is at left, followed by the difference between the model trace and each of the traces predicted by the polynomial interpolators.

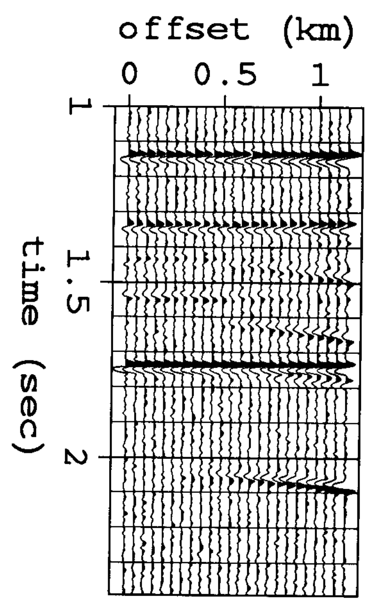


FIG. 8. The NMO-corrected data of Figure 1c with the addition of Gaussian noise.

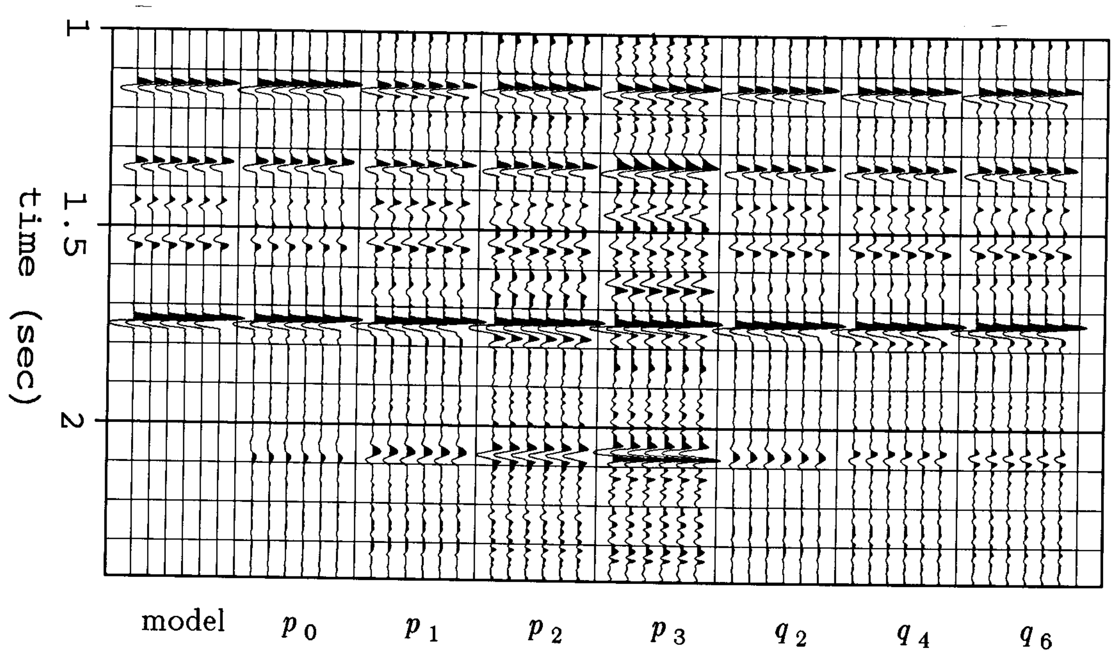


FIG. 9. Zero-offset traces for the noisy compressional and shear converted data of figure 8. The modeled zero-offset trace is at left, followed by the traces predicted by each polynomial interpolator. For clarity, each trace is repeated 6 times. Compare with Figure 6, which shows the same results in the noise-free case.

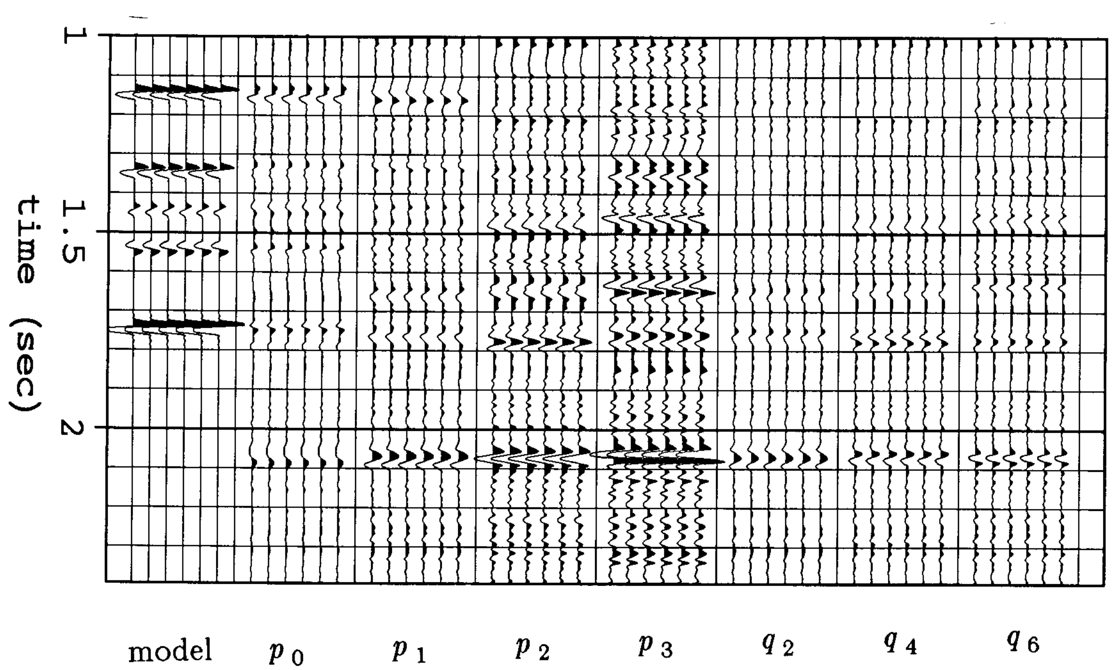


FIG. 10. Errors of zero-offset traces of figure 9. The modeled zero-offset trace is at left, followed by the difference between the model trace and each of the traces predicted by the polynomial interpolators.

the inner offsets will be fitted better by the polynomials, and suppressed less well, than by a conventional stack. Similarly, multiples will be suppressed poorly, because the effects of residual moveout are small at inner offsets. In one sense, the problem here is one of definition. We define some kinds of coherent events as noise, simply because they arise from waves we do not want (or know how) to use. The interpolator methods discussed here are designed to base their prediction upon any coherent trends in the data, and extrapolate them back to zero offset. But often what we really want out of a stack is not what would have been recorded by a zero-offset experiment, but rather the record of an exploding reflectors experiment. The poor suppression of multiples by the interpolator methods is in this sense a mark of their honesty to the data, but it means that in many conventional applications they will be inferior, since the "dishonest" aspect of multiple suppression by conventional stacking is often one of its most important features. However, for comparing stack data with VSP's or synthetic seismograms based on well logs, it may prove better to try to estimate all amplitudes as accurately as possible, multiples included, rather than try to match incompletely suppressed multiples and too strong or too weak primaries.

CONCLUSIONS

The estimation of amplitudes by stacking can be improved by including higher order polynomial terms in the calculation of the stack. The cheapest and most robust way to do this appears to be fitting a polynomial of the form $q_2(x)=c_0+c_2x^2$ to the amplitudes as a function of offset. The cost of doing so is only slightly more than that of conventional stacking. Zero-offset amplitudes calculated this way can be substantially more accurate than those in a conventional stack for events whose amplitudes vary significantly with offset. Multiple suppression may be substantially worse. A possible niche for this method would be in improving the match between synthetic seismograms and stacked reflection data.

ACKNOWLEDGMENTS

I thank Peter Mora for the use of his elastic modeling program, and Francis Muir for useful discussion.

REFERENCES

Atkinson, K.E., 1978, An introduction to numerical analysis: John Wiley and Sons, Inc.

Claerbout, J.F., 1976, Fundamentals of geophysical data processing: McGraw-Hill Book Co.

Mora, P., 1984, Elastic inversion using ray theory: SEP-41

Ostrander, W.J., 1984, Plane-wave coefficients for gas sands at nonnormal angles of incidence: Geophysics, 49, 1637-1648

Woodward, M., 1985, Statistical averages for velocity analysis and stack: Median vs. mean: SEP-42

APPENDIX

Explicit solutions for the leading term c_0 which were used for calculating the zero-offset values are listed here. The variables x and y are assumed to be indexed and summed from 1 to N, and the range of each summation is only over the variables immediately to the right of the summation sign, with no multiple sums intended.

$$p_0: \ c_0 = \frac{1}{N}\sum y$$

$$p_1: \ c_0 = \frac{\sum x \sum xy - \sum x^2 \sum y}{\left(\sum x\right)^2 - N\sum x^2}$$

$$p_2: \ c_0 = \frac{\alpha_2\sum y \ + \beta_2\sum xy \ + \gamma_2\sum x^2 y}{d_2}$$

where

$$\alpha_2 = \left(\sum x^3\right)^2 - \sum x^2 \sum x^4$$

$$\beta_2 = \sum x \sum x^4 - \sum x^2 \sum x^3$$

$$\gamma_2 = \left(\sum x^2\right)^2 - \sum x \sum x^3$$

$$d_2 = \sum x^2\left(\sum x \sum x^3 - N\sum x^4\right) + \left(\sum x\right)^2 \sum x^4 + N\left(\sum x^3\right)^2 + \left(\sum x^2\right)^3$$

$$p_3: \ c_0 = \frac{\alpha_3\sum y \ + \beta_3\sum xy \ + \gamma_3\sum x^2 y \ + \delta_3\sum x^3 y}{d_3}$$

where

$$\alpha_3 = \sum x^2 \left[\left(\sum x^5\right)^2 - \sum x^4 \sum x^6\right] + \left(\sum x^3\right)^2 \sum x^6 - 2\sum x^3 \sum x^4 \sum x^5 + \left(\sum x^4\right)^3$$

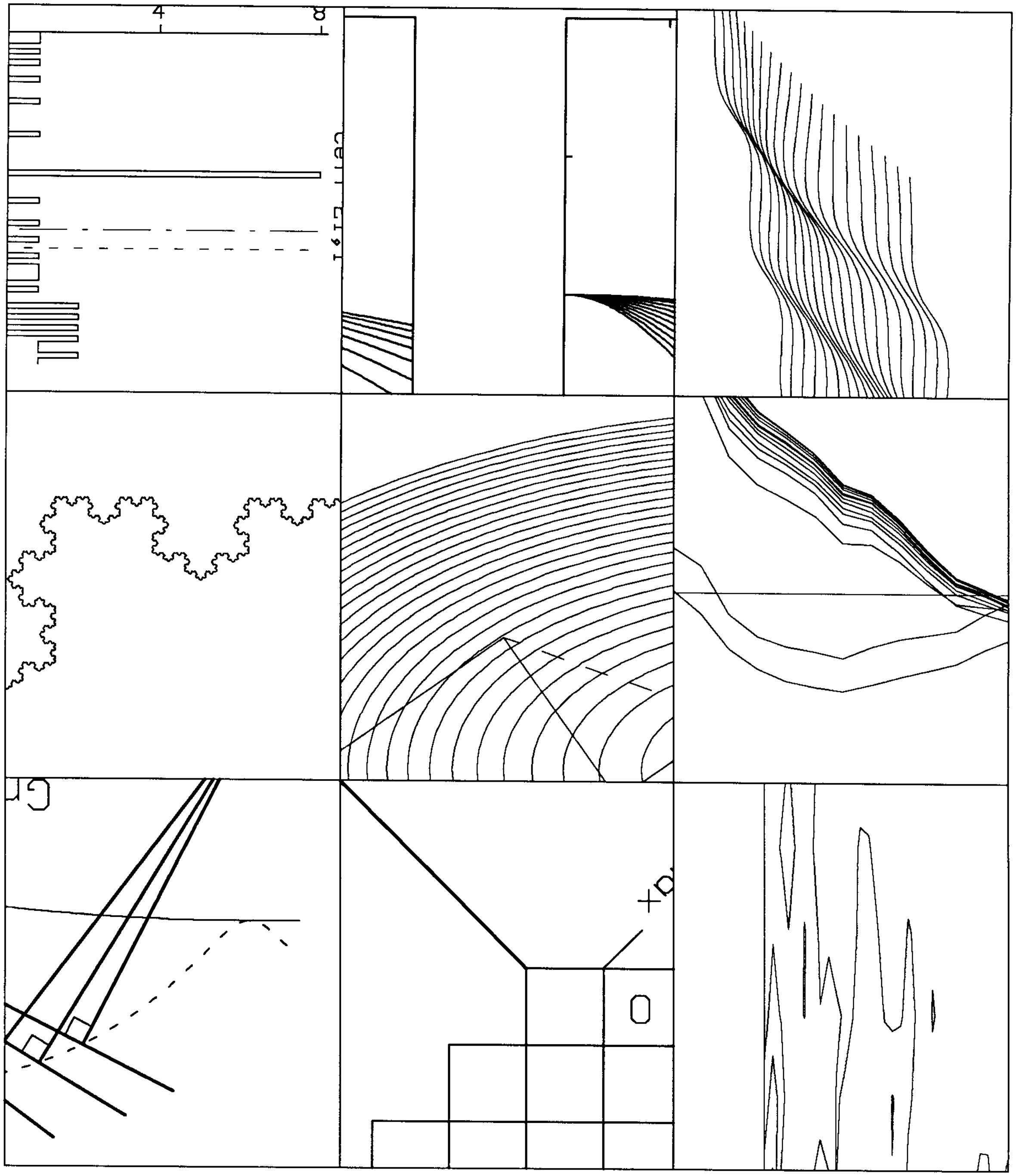
$$\beta_3 = \sum x \left[\sum x^4 \sum x^6 - \left(\sum x^5\right)^2\right] + \sum x^3 \left[\sum x^3 \sum x^5 - \sum x^2 \sum x^6 - \left(\sum x^4\right)^2\right] + \sum x^2 \sum x^4 \sum x^5$$

$$\gamma_3 = \sum x^3\left(\sum x \sum x + \sum x^2 \sum x^4\right) - \left(\sum x^2\right)^2 \sum x^5 - \sum x\left(\sum x^4\right)^2 - \left(\sum x^3\right)^3$$

$$d_3 = \left[N\sum x^2 - \left(\sum x\right)^2\right]\left[\sum x^4\sum x^6 - \left(\sum x^5\right)^2\right] - 2\sum x^2\sum x\sum x^4\sum x^5$$
$$+ 2\sum x^3\left[\sum x\sum x^2\sum x^6 + N\sum x^4\sum x^5 + \left(\sum x^2\right)^2\sum x^5 + \sum x\left(\sum x^4\right)^2\right]$$
$$- \left(\sum x^3\right)^2\left(N\sum x^6 + 2\sum x\sum x^6 + 3\sum x^2\sum x^4\right) - \left(\sum x^2\right)^3\sum x^6$$
$$- N\left(\sum x^4\right)^3 + \left(\sum x^2\sum x^4\right)^2 + \left(\sum x^3\right)^4$$

The corresponding values for the polynomial q_j can be found simply by substituting x^2 for x in all the above expressions. Similar expressions are required to compute c_1, etc, but these are only needed for interpolating offsets other than zero. The denominator d will be the same for all terms.

These quantities look exceedingly complicated, but are wonderfully easy to compute and use on a well equipped computer. One types the normal equations into a program such as MACSYMA, asks it to solve them, then requests that the answer be converted to Fortran code, which it also knows how to do. Then just insert the output code into your program, remembering that the constants α, β, etc. need be computed only once. Finally, have the same MACSYMA program convert the output to a format suitable for input to the typesetting software and insert it into this paper! (Actually, this last feature is buggy on our installation.) For circumstances such as this, in which the same set of equations needs to be solved many, many times but the number of equations is small, I would recommend this approach as the fastest and easiest (but don't try to do the algebra by hand!)



Where have you seen these before?