Preliminary Results on Synthesis and Migration of Transmission-Distorted Vertically-Stacked Sections

Philip S. Schultz

The following figures show the results of applying the concepts and formulas given in the preceding section on the treatment of transmission distortions to a synthetic earth model. In particular, equation (22) was used to generate the inversion (figures 6 through 10). The "surface data" (Figure 2) was generated by a relation derivable from equation (19).

Generation of the surface data from the reflection coefficient series is expressed (in the notation of the previous article) as obtaining $U(z_0)$ from $c'(x,t)$ (recall $c'(x,t) = c(x,t) * w(t)$) . So we have from before,

$$U(z_2) = c(x,t) *_t I(z_2) *_t w(t) \tag{19}$$

$$= c'(x,t) *_t I(z_2)$$

We now convert $U(z_2)$ and $I(z_2)$ to $U(z_0)$ and $I(z_0)$ by means of equations (13) in the previous article,

$$\overset{2}{\underset{0}{1}} M_- U(z_0) = c'(x,t) *_t \overset{2}{\underset{0}{1}} M_+ I(z_0) \tag{19a}$$

and recalling equations (14),

$$U(z_0) = \overset{0}{\underset{2}{1}} M_+ [c'(x,t) *_t \overset{2}{\underset{0}{1}} M_+ I(z_0)] \tag{19b}$$

We therefore have a method of generating transmission-distorted surface data, $U(z_0)$, from the reflection coefficients, $c'(x,t)$. Recall that $I(z_0)$ is trivially a delta-function plane wavefront. The equations above and those in the previous section imply a single reflecting horizon at z_2 . As was stated in the closing remarks of the previous section, generalization to many reflectors offers no conceptual difficulty, but leads to overly cumbersome notation.

The synthetics in this section were generated such that there are two subsurface reflectors of a graben-like geometry. The geometry was made to be similar to the model of Don C. Riley in the March 1974 SEP report (pp. 127 ff.) so that the contrast would be apparent between the two wave propagation effects.

Two potentially confusing aspects of the synthetics should be clarified. First, the entire seismic section is meant to encompass a full 2.34 seconds of data. The dominant wavelength is about nine data points, and since we assume a sampling interval of .004 seconds and the time domain only contains 200 data points, logical inconsistencies appear to abound. The obvious reason we decided to encompass 2.34 seconds of data in 200 data points is that the full 600 points needed would have involved needlessly long computation. The proper interpretation of the display is that the sampling interval is .004 seconds, the dominant frequency is about 28 Hz, and the reflectors are at depths in proper proportion to their position in the time scale. The migration was done with 13 Δz steps at $a = .064$. With an average velocity of 8000 ft/sec, and a receiver spacing of 100 ft, the vertically-stacked section should represent a depth of 2.34 seconds.

The second point of confusion is the upper "chirp" horizon, representing the sea floor. While it shows the proper form of the sea floor topography, it is not a representation of the reflected wave , rather of the transmitted wave but displayed at the position of the expected arrival of the reflected wave.

Notice that the wavefront relief of the sea floor transmission is about $\lambda/2$. This, coupled with the dominant frequency of 28 Hz implies a somewhat less absolute transmission shift than the one wavelength (40 Hz) shift discussed in "Modeling Diffractions of the Transmitted Wave". Figure 2 in that discussion is pertinent here, but the topographic relief inferred from the figure is greater than needed to produce the shift used here.

A Fast and Accurate Shifting Scheme

by Philip S. Schultz

Many situations require that a seismic trace be uniformly shifted up or down by a non-integral number of sample intervals. The most well-known example is in statics corrections, indeed, our treatment of migration in situations where the transmitted wave itself suffers diffractions also requires this type of shifting. We have looked at this problem with the suspicion that in some cases it might be desirable to differentially shift the wave field at each migration iteration (i.e., Δz step). One is then faced with the possibility of executing a shifting routine 50 or 100 times or more before migration is completed. (This subject is treated in detail in the sections of this report on transmission diffractions.) At this point we would like to be sure that the shifting routine used does not hopelessly distort the wave field after so many shifts.

Previously, we have used a simple two-point linear interpolator for all our shifting. It has the advantage of requiring only two MADS (multiply and adds) per point. But as we shall see, it can be substantially improved at the nominal cost of an extra MAD at each point.

To begin, let us examine the properties of a two-point operator interpolating a midpoint. Given a time series (f_1 , f_2) , the midpoint, $f_{1.5}$, can be obtained by

$$f_{1.5} = \frac{1}{2} f_1 + \frac{1}{2} f_2 , \qquad (1)$$

i.e., convolution of the filter (0.5, 0.5) onto the time trace. This filter clearly has a d.c. response of unity. Figure 1 shows why it would be desirable to normalize this filter to a d.c. response of something somewhat larger than unity. In our case we have used 1.005. For midpoint interpolation the filter is zero phase and the distortion is described entirely in the Fourier amplitude. From the figure we see that a 1% amplitude attenuation occurs at 17 points per wavelength. This is a fairly dense sampling and is representative of only the very low end of the spectrum. Admittedly, we have taken the very worst case of two-point interpolation: the midpoint.

We would now like to design a filter to first very accurately interpolate the midpoint, and then use our two-point interpolator on the now more densely sampled data (an idea given me by John P. Burg). In this way, for example, we will effectively create a 16 point per wavelength sampling out of what was formerly 8 points per wavelength. The ultimate use of our two-point interpolator will then be accurate to within 1% for 17/2 or 8.5 points per wavelength.

We choose to interpolate midpoints with the zero phase filter (a, b, b, a) . We have chosen a and b so that the filter will be normalized to a d.c. response of unity and have its Fourier amplitude have minimal (approximately) deviation from unity out to 8 points per wavelength. (As noted in the caption to Figure 2, a slight amplification was acceptable since the two-point filter has a strong tendency to attenuate.) With the above criteria, we created the filter (-0.0682, 0.568, 0.568, -0.0682), described in Figure 2. So, for the time series (f_1, f_2, f_3, f_4), we have

$$f_{2.5} = (-0.0682)f_1 + (0.568)f_2 + (0.568)f_3 + (-0.0682)f_4 \qquad (2a)$$

or

$$f_{2.5} = a\,f_1 + b\,f_2 + b\,f_3 + a\,f_4 \quad . \qquad (2b)$$

Aiming at combining the four-point and two-point filters, we note

$$f_{2+\Delta t} = A(1-\Delta t)f_2 + A\,\Delta t\,f_3 \qquad (3)$$

with the two-point filter, where A is the d.c. normalization. But now if $0 \leq \Delta t \leq \frac{1}{2}$, we can write

$$f_{2+\Delta t} = 2A\,(0.5-\Delta t)f_2 + 2A\,\Delta t\,f_{2.5} \qquad (4)$$

But we have a linear expression for $f_{2.5}$ from equations (2). Combining (2) and (4) yields

$$f_{2+\Delta t} = [2A\,a\,\Delta t]f_1 + [2A(0.5-\Delta t) + 2A\,b\,\Delta t]f_2 + + [2A\,b\,\Delta t]f_3 + [2A\,a\Delta t]f_4 \qquad (5)$$

or

$$f_{2+\Delta t} = [2A\,a\,\Delta t](f_1 + f_4) + [2A(0.5-\Delta t) + 2A\,b\,\Delta t]f_2 + [2A\,b\,\Delta t]f_3 \quad . \qquad (6)$$

And a similar expression can be written for $\frac{1}{2} \leq \Delta t \leq 1$.

Now we have said that we are interested in shifting the entire trace uniformly. That means Δt is a constant. We have already decided on what permanent values to assign to A, a, b . Therefore, the operation given in equation (6) involves only three MADS per point. This should involve an increase in computation time by a factor of 1.5, a modest price.

Two examples of multiple shifting are given in Figures 3 through 6 where comparisons are made between the two-point shifter and the shifter using equation (6). Execution times are given in the second example, and as expected, a factor of about 1.5 is realized.

A subroutine, SHIFT4, is attached which shifts a wave field (or seismic section). An array SH(NX) is input which gives the shift for each of NX traces.

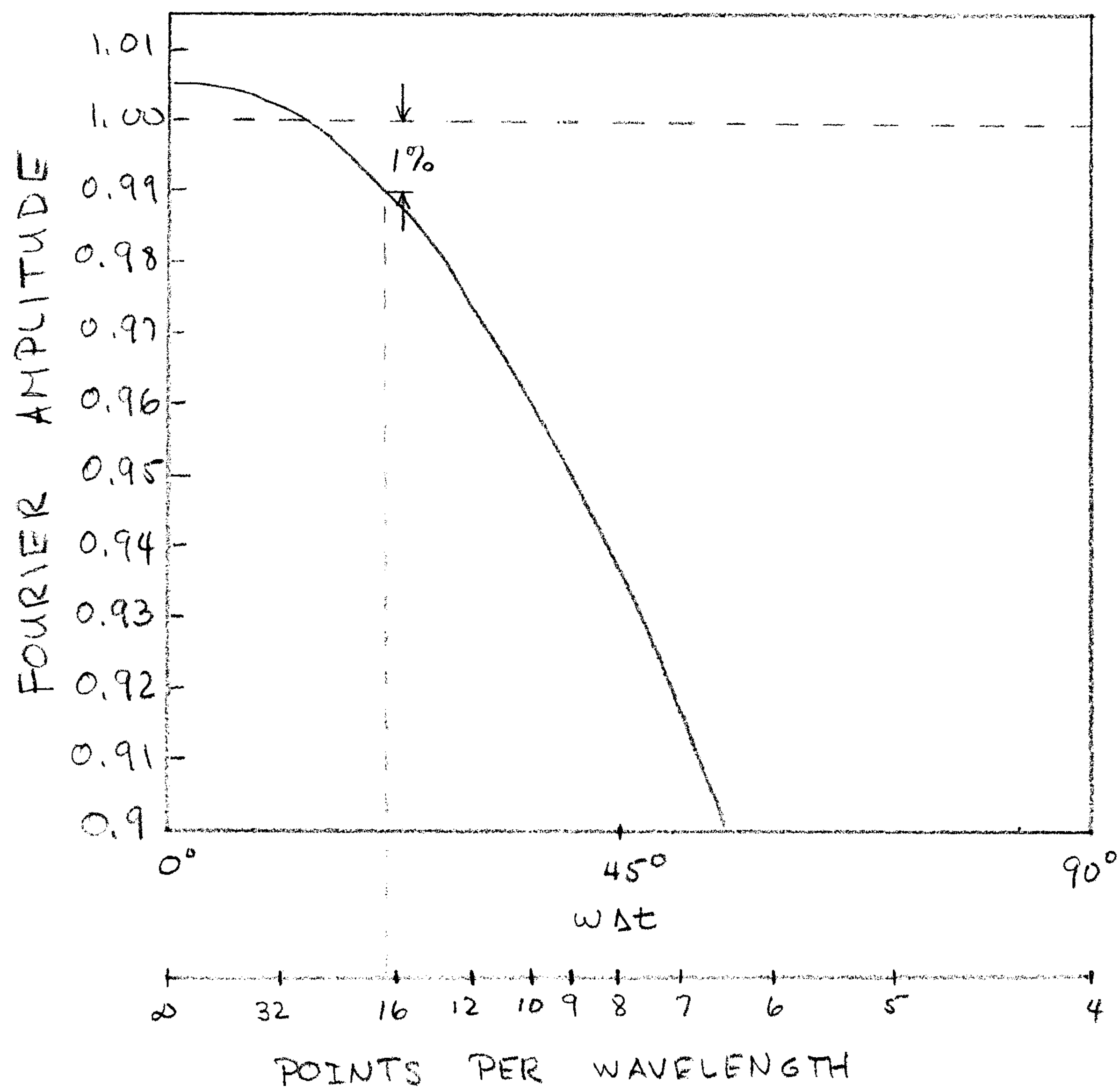

Figure 1. The Fourier amplitude of a simple linear midpoint interpolator normalized to a d.c. response of 1.005, i.e., (0.5025, 0.5025). The curve is a segment of a cosine function which would reach zero at 2 points per wavelength (as expected). At this particular normalization a 1% amplitude attenuation is reached at 17 points per wavelength (this could be extended by renormalization if further d.c. amplification were acceptable).

Midpoint interpolation is the most inaccurate in simple two-point linear interpolation since the interpolated point is not close to either data point, and it can be seen here that at as many as 8 points per wavelength almost 7% of the amplitude is lost <u>for each interpolation</u>.

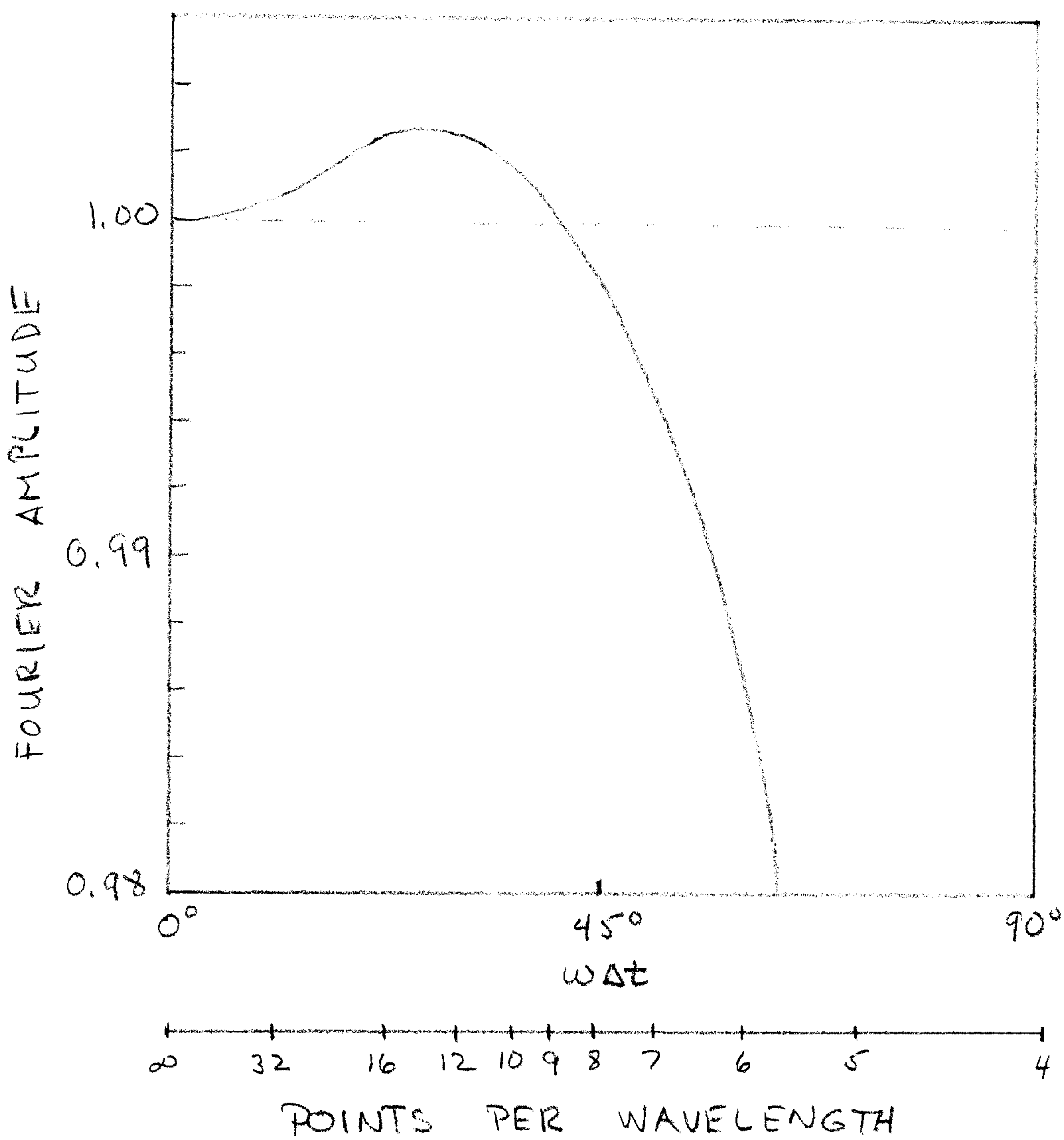

Figure 2. The Fourier amplitude of the four-point midpoint interpolator (-0.0682, 0.568, 0.568, -0.0682). This interpolator, after normalization to a d.c. response of unity, and after recognizing its symmetrical (zero phase) character, allows one degree of freedom in its design. This was chosen so that a sampling of 8 or more points per wavelength would result in an amplitude distortion of no more than approximately 0.2%. In addition, a slight amplification was preferred to a slight attenuation because the eventual two-point variable interpolator has a greater tendency to attenuate.

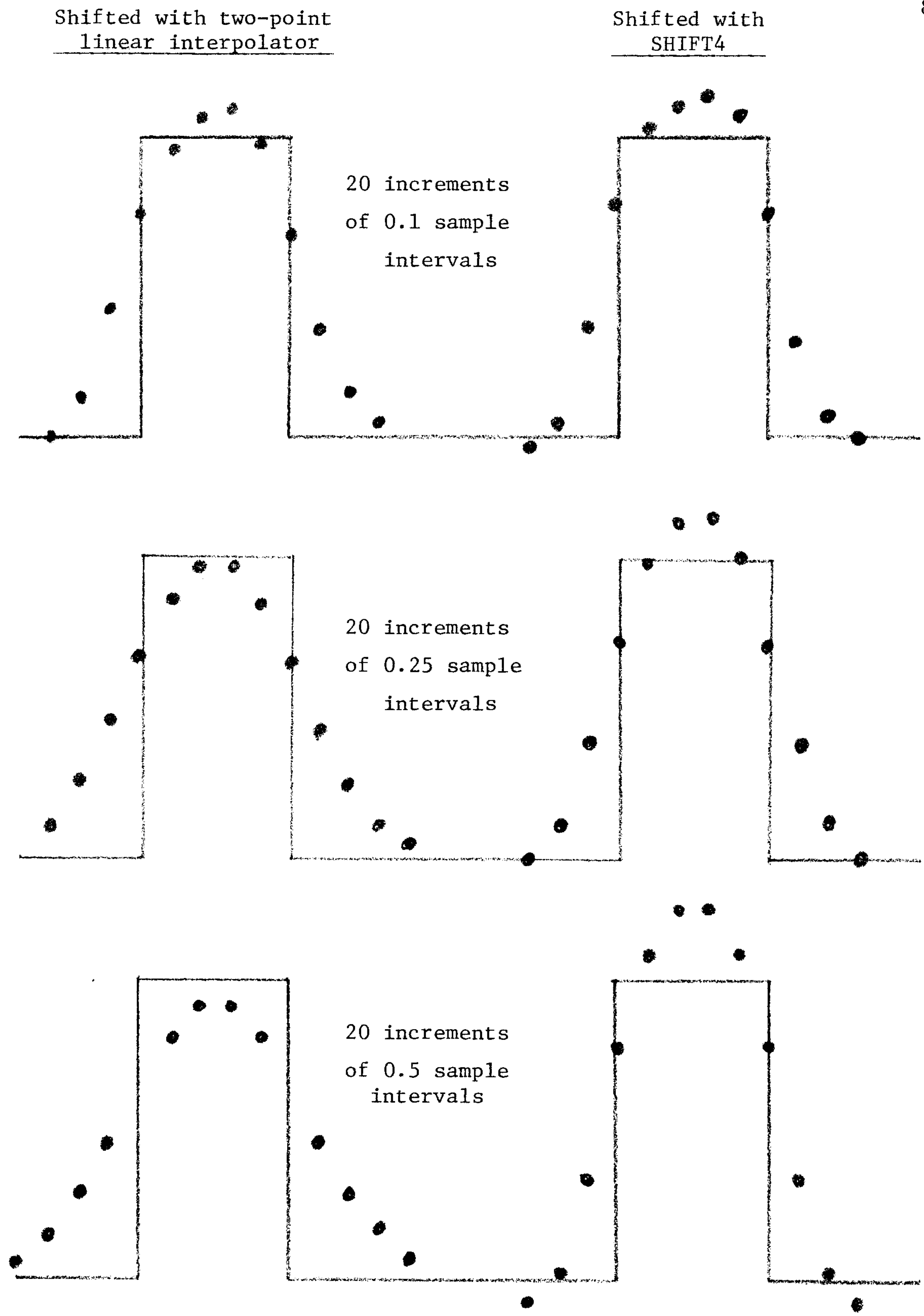

Figure 3. Example showing the comparison of a two-point interpolator with the algorithm, SHIFT4, described in the text. The original waveforms are shown as solid lines, while the actual data points are plotted for the resultant waveforms (after twenty consecutive shifts). Note that SHIFT4 gives consistently less diffusion and attenuation, and also tends to give more uniform results over the range of shifts depicted. The example of a square waveform is deliberately severe in that it contains much energy in low and high frequencies. The real strength of the SHIFT4 routine is in the 8 to 20 points/wavelength range.

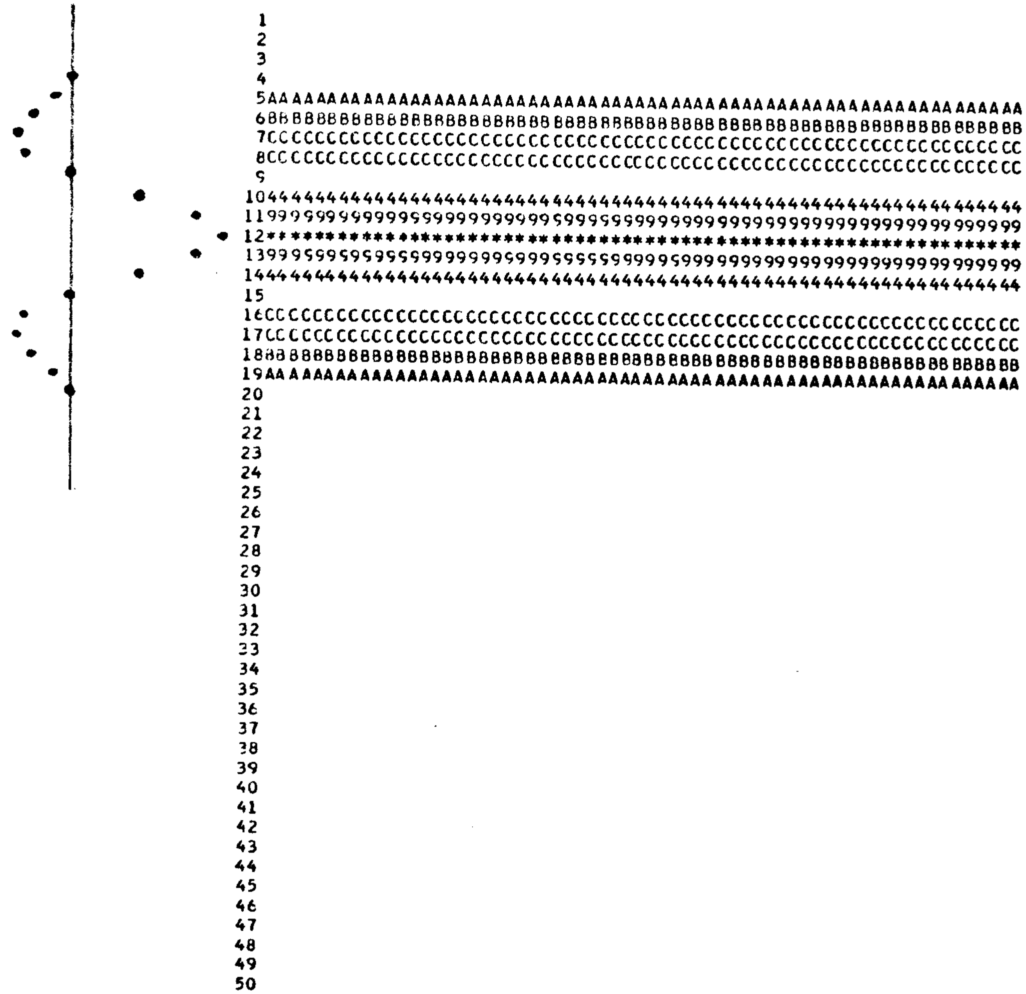

Figure 4.

The initial wave field of a more realistic shifting example. The numeric digits signify positive quantities (saturation: "*"), while the alphabetic digits signify negatives (saturation: "H"). The waveform has a dominant frequency of 9 points/wavelength. A simple two-point interpolator and SHIFT4 were applied, each 100 consecutive times for a total shift of 30 downward (i.e., each shift being 0.3 of a sample interval). The results, along with execution times, are shown in Figures 5 and 6.

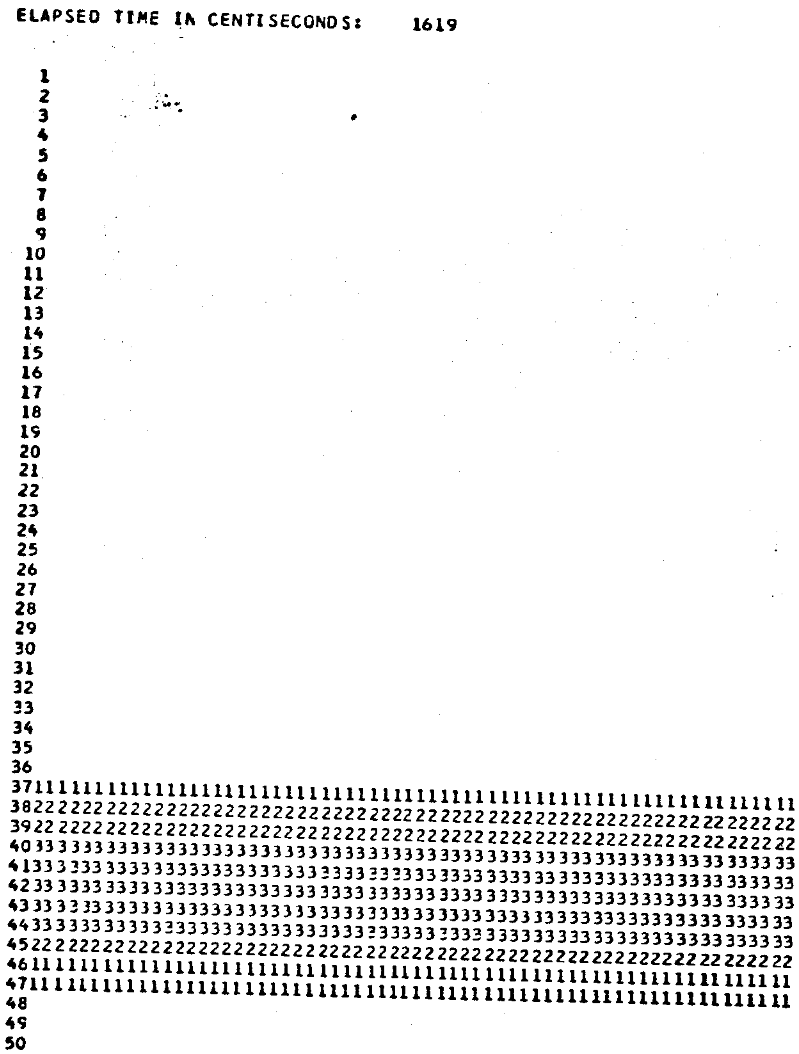

Figure 5. The result of shifting the wave field of Figure 4 downward with 100 shifts of 0.3 each by means of a simple two-point interpolator. The diffusion and attenuation are severe. Execution time is shown.

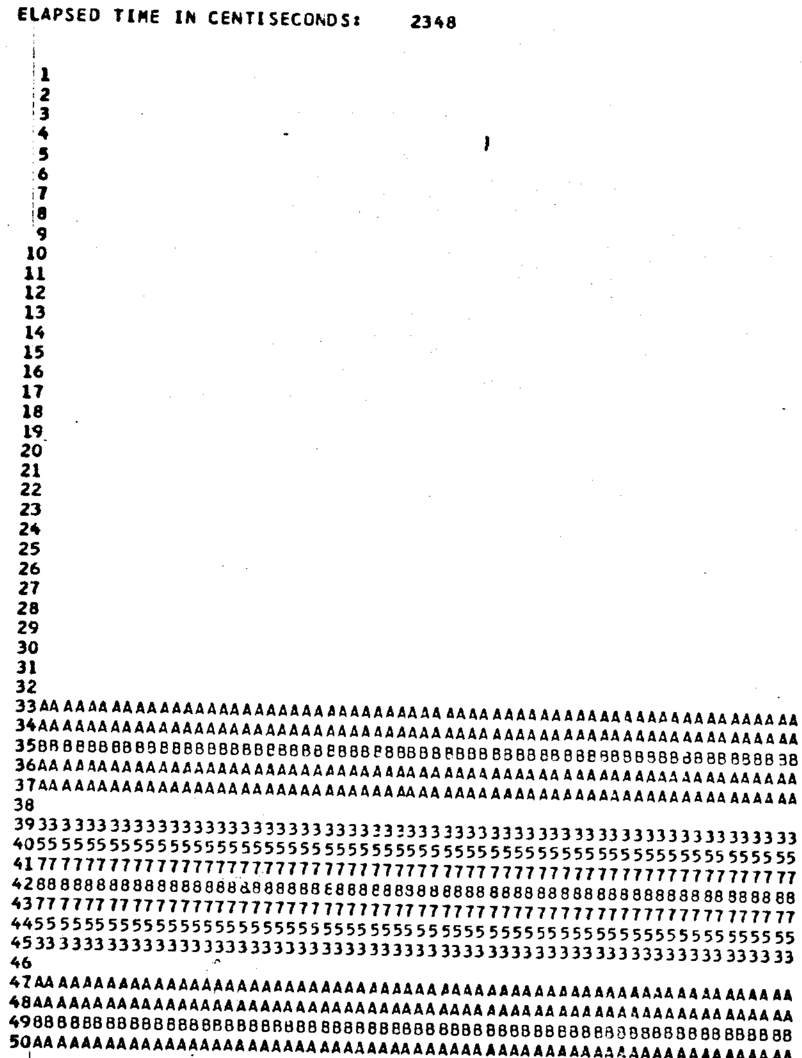

Figure 6. Same as Figure 5 but shifted with SHIFT4 The diffusion and attenuation are much less severe. Execution time is 1.45 times that of the two-point interpolator example of Figure 5.

```
      SUBROUTINE SHIFT4(WAVE,NX,NT,SH)
C-------4 POINT FILTER (-C.0682,0.568,0.568,-0.0682) FIRST TO
C-------INTERPOLATE MIDPOINT; FOLLOWED BY LINEAR
C-------INTERPOLATION. HAS <=1.0% ERROR FOR FREQUENCIES WITH
C-------AT LEAST 9 POINTS/WAVELENGTH.
      DIMENSION WAVE(NX,NT),SH(NX),BUFF(4)
C-------SH(X) IS THE SHIFT (POSITIVE DOWN) IN DATA POINTS
C-------(& FRACTIONS THEREOF) AS A FCN OF X-COORDINATE.
C
      NTM1=NT-1
      NTM2=NT-2
      DO 100 IX=1,NX
C
C-------PROVIDE FOR SH>1
      ISH=AINT(SH(IX))

      IF(ISH.EQ.0)GO TO 86
C
C-------DO SHIFTS OF MULTIPLES OF THE SAMPLE INTERVAL FIRST
      DO 87 IT=1,NT
      IARG=NT+1-IT-ISH
      IF(ISH.LT.0) IARG=IT-ISH
      IF((IARG.LT.1).OR.(IARG.GT.NT)) GO TO 89
      WAVE(IX,IARG+ISH)=WAVE(IX,IARG)
      GO TO 87
   89 WAVE(IX,IARG+ISH)=0.0
   87 CONTINUE
   86 CONTINUE
C
C-------DO FRACTIONS OF SAMPLE INTERVAL
      FSH=SH(IX)-AINT(SH(IX))
      DELX=ABS(FSH)
      IF(FSH.EQ.0.) GO TO 100
      IF(FSH.LT.-0.5) GO TO 74
      IF((FSH.GE.-0.5).AND.(FSH.LT.0.)) GO TO 73
      IF((FSH.GT.0.).AND.(FSH.LT.0.5)) GO TO 72
C     IF FSH >= 0.5 CONTINUE
C-------FILTER COEFFICIENTS
      A14=1.CC5*(-C.0682)*2.0*(1.0-DELX)
      A2=1.005*(2.0*(DELX-0.5)+0.568*2.C*(1.0-DELX))
      A3=1.005*0.568*2.0*(1.0-DELX)
      GO TO 70
   72 CONTINUE
      A14=1.005*(-0.0682)*2.0*DELX
      A2=1.005*0.568*2.0*DELX
      A3=1.005*(2.0*(0.5-DELX)+0.568*2.0*DELX)
      GO TO 70
   73 CONTINUE
      A14=1.005*(-0.0682)*2.0*DELX
      A2=1.005*((1.0-2.0*DELX)+0.568*2.C*DELX)
      A3=1.0C5*0.568*2.0*DELX
      GO TO 70
   74 CONTINUE
      A14=1.005*(-C.0682)*2.0*(1.0-DELX)
      A2=1.005*0.568*2.C*(1.0-DELX)
```

```
      A3=1.005*(2.0*(DELX-0.5)+0.568*2.0*(1.0-DELX))
C
   70 CONTINUE
      IF(FSH.LE.0.) GO TO 99
      DO 77 I=1,3
   77 BUFF(I)=WAVE(IX,NT-3+I)
      WAVE(IX,NT)=A14*BUFF(1)+A2*BUFF(2)+A3*BUFF(3)
      DO 79 IT=3,NTM1
      DO 78 IB=1,3
   78 BUFF(5-IB)=BUFF(4-IB)
      BUFF(1)=WAVE(IX,NT-IT)
      WAVE(IX,NT+2-IT)=A14*(BUFF(1)+BUFF(4))+A2*BUFF(2)
     *                 +A3*BUFF(3)
   79 CONTINUE
      WAVE(IX,2)=A2*BUFF(1)+A3*BUFF(2)+A14*BUFF(3)
      WAVE(IX,1)=A3*BUFF(1)+A14*BUFF(2)
      GO TO 100
   99 CONTINUE
C
      DO 97 I=2,4
   97 BUFF(I)=WAVE(IX,I-1)
      WAVE(IX,1)=A2*BUFF(2)+A3*BUFF(3)+A14*BUFF(4)
      DO 98 IT=2,NTM2
      DO 96 IB=1,3
   96 BUFF(IB)=BUFF(IB+1)
      BUFF(4)=WAVE(IX,IT+2)
   98 WAVE(IX,IT)=A14*(BUFF(1)+BUFF(4))+A2*BUFF(2)+A3*BUFF(3)
      WAVE(IX,NTM1)=A14*BUFF(2)+A2*BUFF(3)+A3*BUFF(4)
      WAVE(IX,NT)=A14*BUFF(3)+A2*BUFF(4)
  100 CONTINUE
      RETURN
      END
```