# SPECTRAL WHITENING

## OF SEISMIC DATA

Ralph Knapp

KGS Open-file Report 91-31

August 9, 1991

## SPECTRAL WHITENING

Spectral whitening is a process designed to improve the resolution of data at a stage near completion. It performs about half the function of a spiking deconvolution which has the same purpose. The difference is that spiking deconvolution affects both amplitude and phase of the wavelet. For instance, it assumes, and tries to maintain, that the wavelet is minimum phase. Spiking deconvolution also assumes that the reflectivity series of the earth is random. This assumption is necessary if the phase portion of spiking deconvolution is not to adversely affect the results by affecting the reflectivity series of the earth. Unfortunately, the assumption is probably more often wrong than right, and spiking deconvolution can transform data inappropriately, affecting the accuracy of the interpretation. Spectral whitening is more general and it is zero phase. Rather than using the specific values of the amplitude spectrum, it uses a smoothed version, a closer approximation to the wavelet itself, and it does not affect phase. Thus, it does not change the reflectivity series of the earth and does not adversely affect interpretation.

The resolution of seismic data is directly proportional to the bandwidth of the data. Further, for relatively broad band data, resolution depends on the highest frequency content (Knapp, 1990). Basically, resolution improves as the time width of the wavelet decreases, resolution and wavelet time width are inversely proportional:

$$R \approx {}^{1}/_{\Delta t}.$$

The Scaling Property of Fourier transform theory states that wavelet time width is restricted according to frequency bandwidth (see, e.g., Brigham, 1974):

$$\Delta t \geq {}^{1}/_{\Delta f}.$$

The uncertainty in this equation is due to the influence of the phase spectrum, the more complex the phase spectrum, the broader the wavelet. For instance, the minimum possible duration of a wavelet occurs for zero-phase, the least complicated phase. To the other extreme, random phase results in noise that has an infinite duration.

The purpose of whitening seismic data is to broaden its amplitude spectrum and improve resolution in its final stages. Figure 1 shows a well-processed seismic line in its final stages. During processing, attempts were made to maintain a broad amplitude spectrum by using common filtering techniques. However, these filtering techniques are not specifically designed for the data and only do an approximate broadening, and one of the final stages of processing is stacking or averaging of the CDP traces. Stacking has the inherent characteristic that it stacks low frequencies better than high frequencies. This is because high frequencies are more sensitive to small errors than low frequencies.

The average amplitude spectrum for all the data of Figure 1 is shown in Figure 2. The plot is the output of program SPEC (Appendix 1). In principle, the seismic wavelet has a smooth amplitude spectrum. The influence of the geological section causes peaks and valleys within this smooth section as some frequencies are enhanced by constructive interference and others are reduced by destructive interference. The wavelet spectrum is estimated by the "Average Smoothed Spectrum." This curve is best determined by the processor. automated methods have typically not had good results. The trapezoid, "Desired Average Spectrum," represents the spectrum that the processor would like the data to have. It is important that the frequency band chosen have reflection data at reduced amplitude and not just noise. This also is determined by the processor based on experience and experimentation. Addition of high frequencies to the data improves resolution, but it also increases noise and it reduces the smoothness of the data. Low-frequency, low-resolution data is smoother and more pleasant looking than high-frequency, high-resolution data. Even experienced interpreters will frequently choose low-resolution data in preference to high-resolution data because it is smoother, less complex, and easier to work with.

Using the curves of Figure 2, a custom filter is designed to boost the amplitudes of the "Average Smoothed Spectrum" to the levels of the "Desired Average Spectrum." The values determined are input into the program WHITE (Appendix 2), which applies a zero-phase filter of the given amplitude spectrum. It is important that the filter input be smooth and not too outrageous in boosting amplitudes. Factors of 10 are probably a maximum. Sample input for program WHITE is given in Appendix 3. The values given are actually the square-root of the multipliers needed to do the task. Thus the program was run twice to achieve the task. This procedure was followed so that the intermediate step of a partial boost could be monitored.

The results of spectral whitening are seen on Figure 3. The spectrum is Figure 4. Results are not real dramatic because the data was fairly broad band in the first place, but several important improvements have occurred. For instance, the strong reflector at about 65 ms, CMP 240, that wedges upward at about CMP 500 has strong side-lobe signal on Figure 1. This is essentially gone on Figure 3, revealing more information in the region directly above this event. Events in the region of 70 to 90 ms are clarified by spectral whitening. The focus of the events at about 100 ms and 130 ms are clearer. Predictably, the whitened section (Figure 3) is more complex and less smooth, but that is a necessary consequence of improving resolution or focus.

Figure 5 shows zero-phase equivalent wavelets of computer-smoothed versions of Figures 2 and 4, respectively. This shows the expected improvement of the data. As seen on the sections and described above, the side-lobe noise of the wavelet is reduced substantially. After whitening, the central peak is sharper, and the associated troughs are smaller. In principle, addition of low frequencies reduces the side-lobe noise and addition of high frequencies improves the sharpness of the central peak.

## REFERENCE

Brigam, E.O., 1974, The Fast Fourier Transform: Prentice-Hall.

Knapp, R.W., 1990, Seismic resolution of thick beds, thin beds, and thin-bed cyclothems: Geophysics **55** (9), p. 1183-1190.

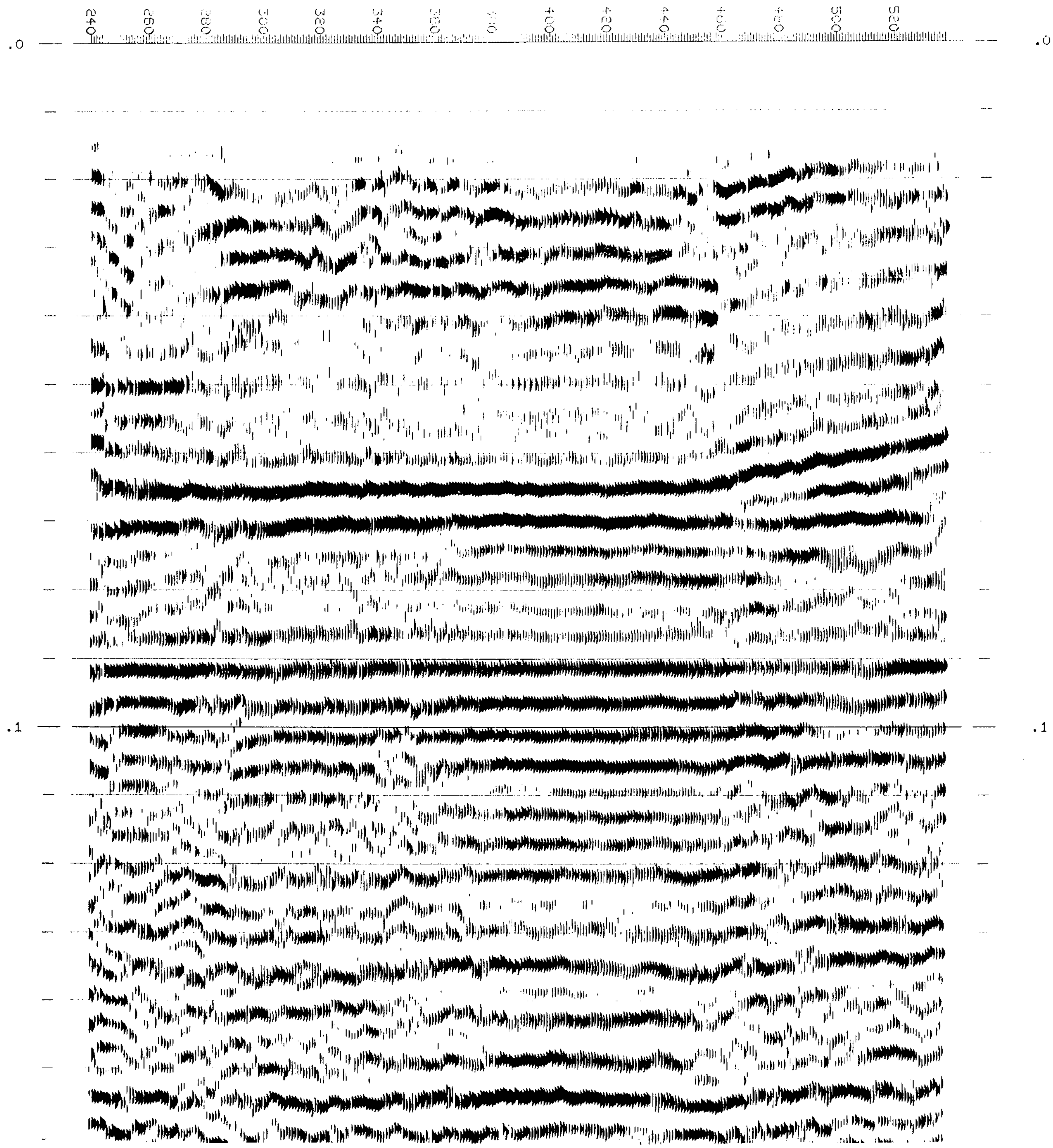


**FIGURE 1.** Seismic data processed to the final step of spectral whitening.

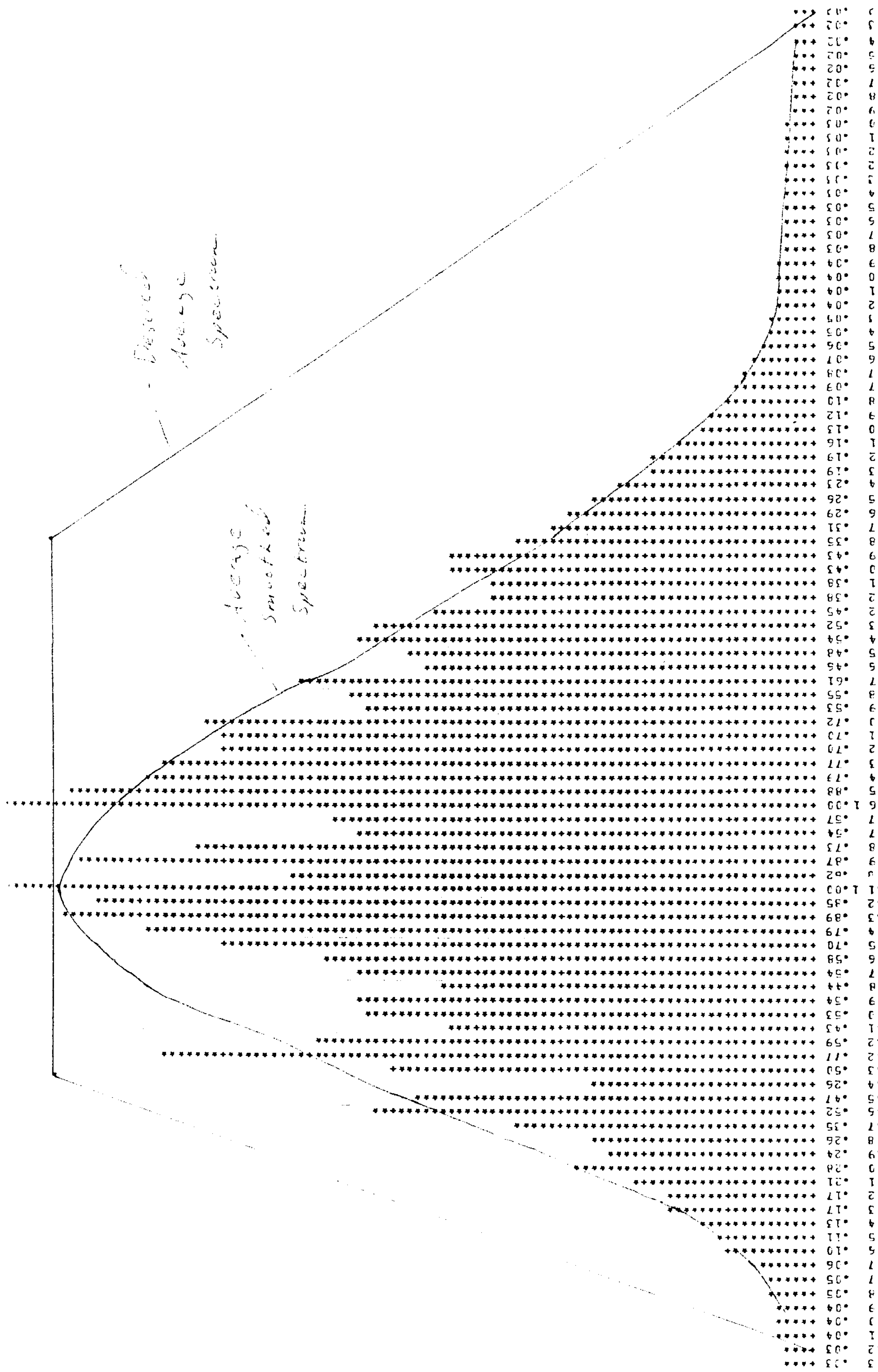


**FIGURE 2.** Amplitude spectrum of Figure 1, average of all traces.

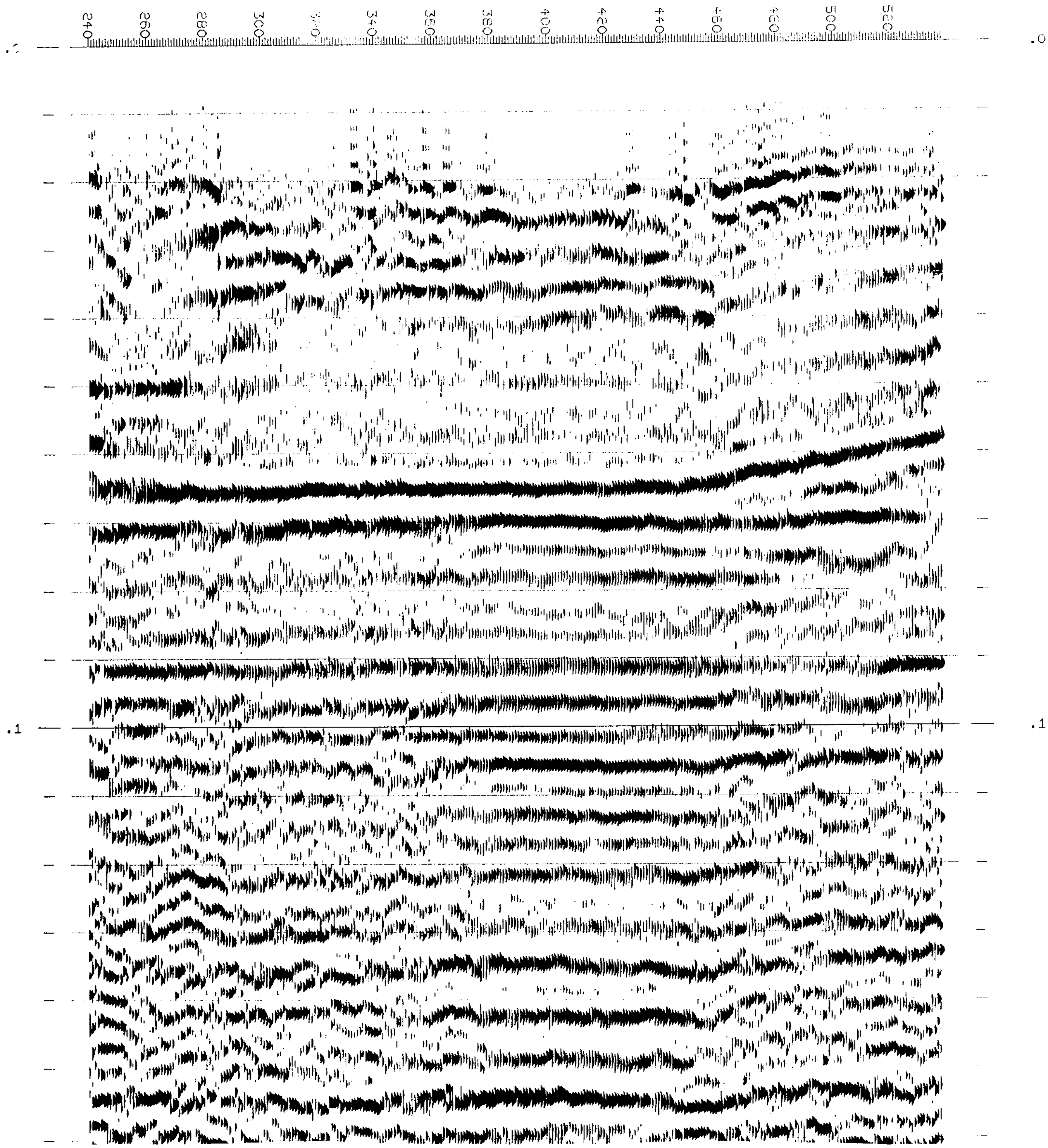


**FIGURE 3.** Seismic section of Figure 1 after spectral whitening.

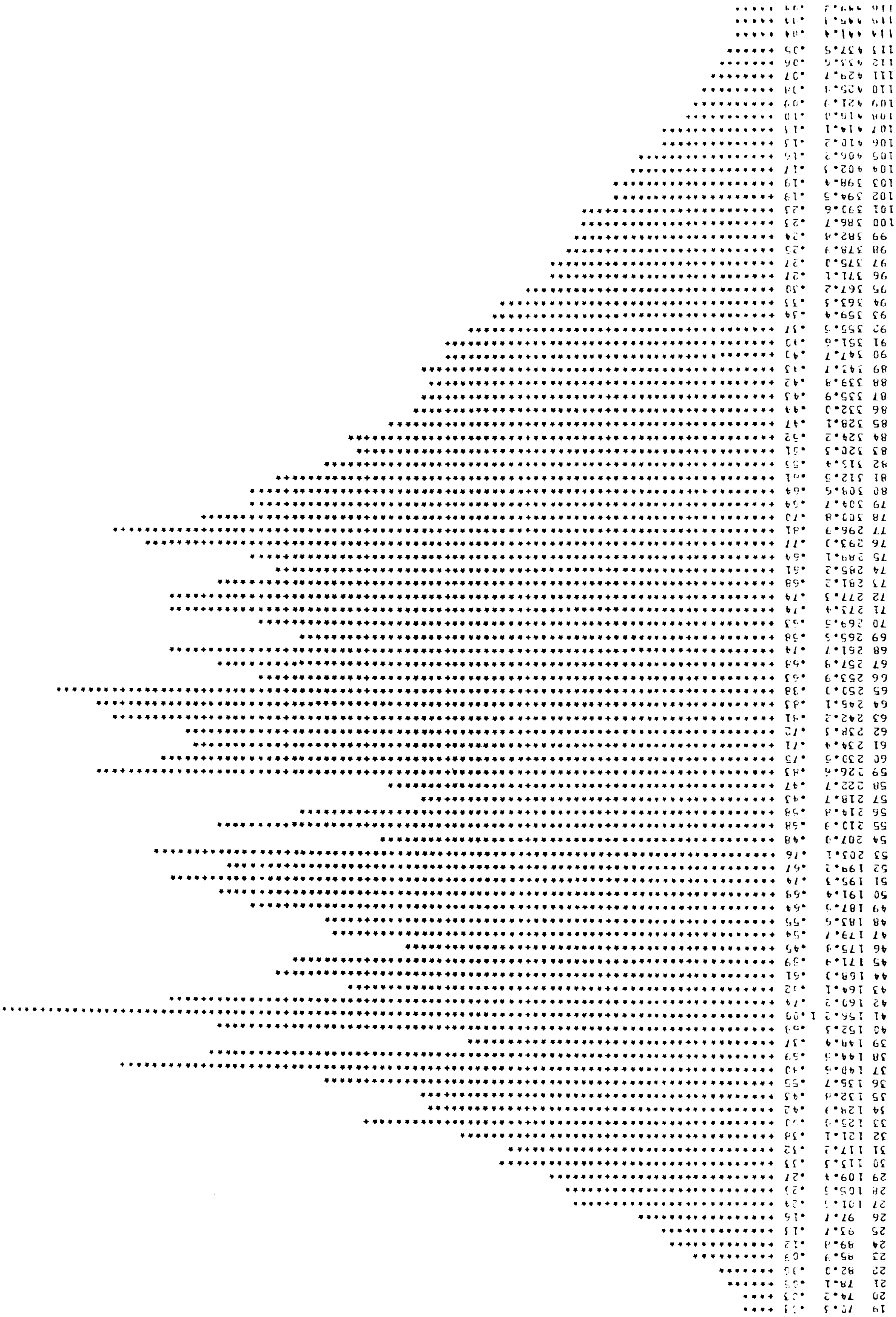

**FIGURE 4.** Amplitude spectrum of Figure 3, average of all traces.

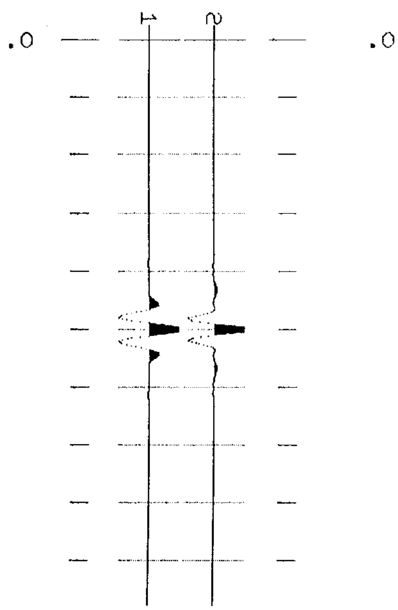


**FIGURE 5.** (a) Zero-phase wavelet derived from smoothed version of Figure 2. (b) Zero-phase wavelet derived from smoothed version of Figure 4.

## APPENDIX 1. PROGRAM SPEC

```
        PROGRAM spec
* amplitude spectrum for seismic data
* August 8, 1991    RWK
*
        integer*2 j,reel1(1600),k,reel2(200),trh(120)
        integer*2 nsamp,itype,incre
        real*4 hamp(1000),out(256)
        complex z(1024)
        integer param
        CHARACTER*1 STARS(101)
        character*30 dataIN,fileOUT
        equivalence(range,trh(19))
        pi=3.14159265
        l=param('ga',2,0,fileout,lerror)
        l=param('ga',1,0,DATAIN ,lerror)
        open(6,file=fileout,iointent='output')
        open(1,file=dataIN ,iointent= 'INput',mode='binary',recfm='dynamic',form='unfo
        READ (1)J,(REEL1(I),I=1,J),K,(REEL2(I),I=1,K)
        NSAMP=REEL2(11)
        ITYPE=REEL2(13)
        incre=reel2(9)
        si=float(incre)/1000000.
        do 10 i=1,10
        l=2**i
        if(l.gt.nsamp) go to 11
10      continue
11      continue
        fny=.5/si
        df =fny*2./float(l)
*
        do 12 i=1,l/4
12      out(i)=0.
        DO 13 I=1,101
13      STARS(I)='*'
        DO 14 I=1,101,10
14      STARS(I)='+'
*
        write(6,*) ' AVERAGE AMPLITUDE SPECTRUM OF SEISMIC LINE '
        WRITE(6,*) ' '
        WRITE(6,*) ' INPUT FILE = ',DATAIN
        WRITE(6,*) ' '
        WRITE(6,*) '     NSAMP = ',NSAMP
        WRITE(6,*) '     INCRE = ',INCRE
        WRITE(6,*) '     SI    = ',SI
        WRITE(6,*) '     '
        WRITE(6,*) ' * * * * * * * * * * * * * * * * * * * * * * * * '
        WRITE(6,*) ' '
1       CALL TREAD (TRH,HAMP,NSAMP,ITYPE,1,IERR)
        IF (IERR.NE.0) go to 100
        do 2 i=1,l
2       z(i)=cmplx(0.,0.)
        do 21 i=1,nsamp
21      z(i)=cmplx(hamp(i),0.)
        call fouri(z,l,-1)
        do 22 i=1,l/4
22      out(i)=cabs(z(i))/1000.+out(i)
        GO TO 1
100     omax=0.
        do 101 i=1,l/4
101     omax=amax1(out(i),omax)
        do 102 i=1,l/4
102     out(i)=out(i)/omax
        do 103 i=1,l/4
        f=float(i-1)*df
        n=ifix (out(i)*100.+1.5)
        write(6,'(1x,i3,1x,f5.1,1x,f4.2,1x,101a1)')
     + i,f,out(i),(stars(m),m=1,n)
103     continue
        stop
        end
```

# APPENDIX 2. PROGRAM WHITE

```
      PROGRAM white
*
* spectral whiten seismic data
* Aug 8, 1991     RWK
*
        integer*2 j,reel1(1600),k,reel2(200),trh(120)
        integer*2 nsamp,itype,incre,units
        real*4 hamp  (2000) ,wt(1025),winput(100),freq(100)
        COMPLEX z(2048)
        integer param
        character*30 datain,dataout,filein,fileout
        equivalence (range,trh(19))
        pi=3.14159265
        l=param('ga',2,0,fileout,lerror)
        l=param('ga',1,0,filein ,lerror)
        open(5,file=filein,pad='yes')
        open(6,file=fileout)
        read (5,'(a30)') datain,dataout
        write(6,*) ' whiten  910808'
        write(6,*)' datain  = ', datain
        write(6,*)' dataout = ', dataout
        write(6,*)' * * * * *  '
        open(2,file=dataout,iointent='output',mode='binary',recfm='dynamic',form='unfc
        open(1,file=datain,iointent='input',mode='binary',recfm='dynamic',form='unforn
*
        read(1)j,(reel1(i),i=1,j),k,(reel2(i),i=1,k)
        NSAMP=REEL2(11)
        ITYPE=REEL2(13)
        reel2(13)=1
        incre=reel2(9)
        write(2)j,(reel1(i),i=1,j),k,(reel2(i),i=1,k)
*
        si=float(incre)/1000000.
        fny = .5/si
        do 10 i=5,11
        n=2**i
        if(n.ge.nsamp) go to 11
10      continue
11      continue
        PRINT *,' N = ',n
        df=2.*fny/float(n)
*
        write(6,*)' nsamp = ',nsamp
        write(6,*)' itype = ',itype
        write(6,*)' incre = ',incre
        write(6,*)' si    = ',si
        do 12 i=1,100
        read(5,*) freq(i),winput(i)
        if(freq(i).ge.9999.) go to 13
12      continue
13      continue
*
        l=1
        do 20 i=1,n/2+1
        f=float(i-1)*df
21      continue
        wt(i)=winput(l)
        if(f.le.freq(l).and.l.eq.1) GO TO 20     ! FIRST POINTS
        if(freq(l+1).ge.9999.) go to 20          ! LAST POINT
        if(f.ge.freq(l+1)) then
                l=l+1
                go to 21
        endif
        wt(i)=wt(i)+(f-freq(l))/(freq(l+1)-freq(l))*(winput(l+1)-wt(i))
20      continue
```

```
      PRINT *,(WT(I),I=1,N/2+1)
*
1     call tread (trh,hamp,nsamp,itype,1,ierr)
      if(ierr.ne.0) stop
      if(trh(15).ne.1) go to 999      ! if dead trace
      do 90 i=1,n
90    z(i)=cmplx(0.,0.)
      do 91 i=1,nsamp
91    z(i)=cmplx(hamp(i),0.)
      call fouri(z,n,-1)
      do 92 i=2,n/2+2
92    z(i)=z(i)*wt(i)
      do 93 i=n/2+2,n
93    z(i)=conjg(z(n+2-i))
      Z(n/2+1)=cmplx(0.,0.)
      Z(1)=cmplx(0.,0.)
      call fouri(z,n,+1)
      DO 99  I=1,NSAMP
 99   HAMP(I)=real(z(i))
*
999   continue
      j=2*nsamp+120
      write(2) j,trh,(hamp(i),i=1,nsamp)
      GO TO 1
      end
```

## APPENDIX 3. INPUT TO PROGRAM WHITE

```
KV.IND.CSTAK.FLAT3
KV.IND.CSTAK.FLAT.WHITE3
74 1.00
82 1.41
84 1.73
90 1.95
98 1.95
106 1.84
113 1.70
148 1.45
152 1.41
203 1.05
226 1.10
242 1.18
158 1.26
265 1.34
277 1.44
300 1.70
305 1.77
316 1.90
324 2.05
332 2.24
344 2.72
356 3.24
367 3.50
375 3.54
387 3.72
398 3.37
410 2.89
445 1.00
99999 1.00
```