Signal-to-noise through the use of the Semblance Statistic

Andrew J. Kalik

Kansas Geological Survey
The University of Kansas
Lawrence, Kansas 66046

June 27, 1988

Open-file Report No. 88-27

Kansas Geological Survey
Open-file Report

Disclaimer

The signal-to-noise ratio is often used to assess the relative data quality of a seismic record. It may be determined by dividing the integrated signal of a trace by its integrated noise (Sheriff, 1984). Unfortunately, the above procedure is heavily subjective. This is undesirable since we are then attempting to quantify something which has been interpreted. When we look at a seismic section and attempt to distinguish between signal and noise, our first criterion is coherence. This is because coherence is typically indicative of reflection events. Of course, incoherent signal and coherent noise also exist on seismic records.

Incoherent signal is often diagnostic of static problems or structural disturbances, such as fault zones and salt diapirs. It may also have seismic stratigraphic significance (Mitchum, et al, 1977). Incoherent signal is effectively worthless without corresponding coherent signal. Coherent noise is insidious and must be correctly identified to avoid gross misinterpretation. It may result from an incorrect velocity function, insufficient velocity control, static problems, spatial aliasing, or over-processing. Both coherent noise and incoherent signal degrade any attempt to determine the signal-to-noise ratio based on coherency alone. However, if coherent noise, such as multiples and refractions are correctly identified, a signal-to-noise ratio based only on coherence is a conservative estimate of the true signal-to-noise ratio.

Keeping all the above in mind, what we would like as a useful basis for comparison is a ratio of coherent signal to incoherent noise. Such a ratio is available through the use of coherency statistics (Douze, 1979). For signal-to-noise comparisons of seismic data, I developed a coherency program based on the semblance statistic (Sheriff and Geldart, 1982).

$$SEMB(k) = \frac{\sum_{j=k-n/2}^{k+n/2} \left[\sum_{i=1}^{M} HAMP(f_{ij}) \right]^2}{M \sum_{j=k-n/2}^{k+n/2} \sum_{i=1}^{M} HAMP(f_{ij})^2},$$

This is a measure of multichannel coherence, where M is the number of traces; n is the number of samples, centered at k; and HAMPij is the amplitude of the jth sample of the ith trace. The program (Appendix) computes the signal-to-noise ratio based on the coherency of seismic traces summed over the n samples and m+1 CDP's. The program may be used successfully on both stacked and unstacked data. However, to capitalize on the redundancy inherent in CDP data, unstacked data are preferable. As stated previously, accounting for the possibility of both coherent noise and incoherent signal, a coherency plot of the semblance statistic is a suitable measure of signal-to-noise on a seismic section.

Random unstacked data is shown in Figure 1. This data was used in order to test program SN.F77 (Appendix). On Figures 2 through 10 only three parameters were varied: the number of samples (n), the number of CDPs (m+1), and the sampling frequency (st). All the random values are between -100 and 100. There are 1,000 values per trace; four values each millisecond. Barely noticeable on Figure 1, are embedded spikes at 62 ms and 187 ms. The spike at 62 ms was given a value of 50 on each trace. The spike at 187 ms was given a value of 25 from traces 1 to 32. The value at 187 ms on traces 33 through 64 is random. Note how obvious the spikes are on the signal-to-noise plots of Figures 2 through 10. The negligible effect on random data of varying the number of samples (in time) for which the semblance statistic is summed over (n), is shown in Figures 4, 5, 7 and 8. The result is a minor smearing effect (in time) for large values of n (compare Fig's. 4 and 8). The effect of varying the number of traces for which the semblance statistic is summed over (m+1), is shown in Figures 2, 3, and 4. Figure 4 is noticeably noisier (more blank area) than Figure 2. This indicates less random correlation upon averaging. Figures 9 and 10 show the effect of summing over too many traces. This smearing effect indicates instability, and causes random noise to appear coherent. The effect of lower frequency random data can be seen on Figure 6. For this figure, every other value of Figure 1 was linearly interpolated. This process yielded a lower effective frequency. Comparison of Figure 6 with Figure 5 indicates that higher frequency data (Fig. 5) is more prone to random correlation.

Two presentation formats (Fig's. 12 21) of the signal-to-noise content are displayed for actual seismic data acquired near Lyndon, Kansas (Fig. 11). Besides presentation format, Figures 12 and 13 differ only in sampling frequency. After the coherency matrix is computed, all contour plots are made up of data sampled one fifth as often as that of the corresponding wiggle trace plots. This is due to a limitation inherent in the contouring software. The actual data exhibits similar effects to that of the random data upon varying the number of samples (n), and the number of CDPs (m+1). Please note that unlike the data used in Figures 12 and 13, the random data is single fold.

REFERENCES

Douze, E.J., and Laster, S.J., Statistics of Semblance, Geophysics, Vol. 44, No. 12, p. 1999-2003, 1979.

Mitchum, R.M., Jr., Vail, P.R., and Sangree, J.B., Seismic stratigraphy and global changes of sea level, Part six: Stratigraphic interpretation of seismic reflection patterns in depositional sequences, AAPG Memoir 26, p. 128, 1977b.

Sheriff, R.E., ed., Encyclopedic Dictionary of Exploration Geophysics, Society of Exploration Geophysicists, p. 39, 1984.

Sheriff, R.E., and Geldart, L.P., Exploration Seismology, Vol. 1 - 2, Cambridge University Press, 1982.

FIGURE CAPTIONS

Figure 1. Unstacked random seismic data. Barely noticeable are embedded spikes at 62 ms and 187 ms (refer to the text).

Figure 2. Signal-to-noise plot of the random data in Figure 1, summing over m+1=1 traces and n=1 sample. Note the almost startling appearance of the embedded spikes.

Figure 3. Signal-to-noise plot of the random data in Figure 1, summing over m+1=3 traces and n=1 samples. Note the decrease in coherency relative to Figure 2.

Figure 4. Signal-to-noise plot of the random data in Figure 1, summing over m+1=7 traces and n=1 samples. Note the decrease in coherency relative to Figure 3.

Figure 5. Signal-to-noise plot of the random data in Figure 1, summing over m+1=7 traces and n=3 samples.

Figure 6. Signal-to-noise plot of the random data in Figure 1, again summing over m+1=7 traces and n=3 samples, but at an effective frequency of one-half that used for Figure 5. Note the substantial decrease in random coherency relative to Figure 5.

Figure 7. Signal-to-noise plot of the random data in Figure 1, summing over m+1=7 traces and n=7 samples.

Figure 8. Signal-to-noise plot of the random data in Figure 1, summing over m+1=7 traces and n=9 samples.

Figure 9. Signal-to-noise plot of the random data in Figure 1, summing over m+1=13 traces and n=3 samples. Note the incipient "smearing" of apparent coherency.

Figure 10. Signal-to-noise plot of the random data in Figure 1, summing over m+1=21 traces and n=3 samples. Note the smearing effect resulting from summing over too many traces relative to the number of traces in the original data set.

Figure 11. Final stacked seismic section of CDP Line 6 from near Lyndon, Kansas. Note the continuity of the reflector across the chaotic zone (100 ms at CDP 272). Acquire with 220 Hz low cut filters.

Figure 12. Signal-to-noise contour map of the data in Figure 11, summing over m+1=1 CDPs and n=3 traces. Note the proliferation of apparent signal.

Figure 13. Signal-to-noise plot of the data in Figure 11, summing over m+1=1 CDPs and n=3 traces. Note the proliferation of apparent signal.

Figure 14. Signal-to-noise contour map of the data in Figure 11, summing over m+1=3 CDPs and n=3 traces. Note the decrease in apparent signal relative to that of Figure 12.

Figure 15. Signal-to-noise plot of the data in Figure 11, summing over m+1=3 CDPs and n=3 traces. Note the decrease in apparent signal relative to that of Figure 13.

Figure 16. Signal-to-noise contour map of the data in Figure 11, summing over m+1=7 CDPs and n=1 traces. Note how well this figure approximates the reflection signal on Figure 11.

Figure 17. Signal-to-noise plot of the data in Figure 11, summing over m+1=7 CDPs and n=1 traces. Note how well this figure approximates the reflection signal on Figure 11.

Figure 18. Signal-to-noise contour map of the data in Figure 11, summing over m+1=7 CDPs and n=3 traces. Note the negligible change resulting from increasing the number of samples (n) for which the semblance statistic is summed over (compare with Fig. 16).

Figure 19. Signal-to-noise plot of the data in Figure 11, summing over m+1=7 CDPs and n=3 traces.

Figure 20. Signal-to-noise contour map of the data in Figure 11, summing over m+1=13 CDPs and n=3 traces. Note the continuous signal across the area of chaotic energy in Figure 11.

Figure 21. Signal-to-noise plot of the data in Figure 11, summing over m+1=13 CDPs and n=3 traces. Note the continuous signal across the area of chaotic energy in Figure 11.

Figure 1

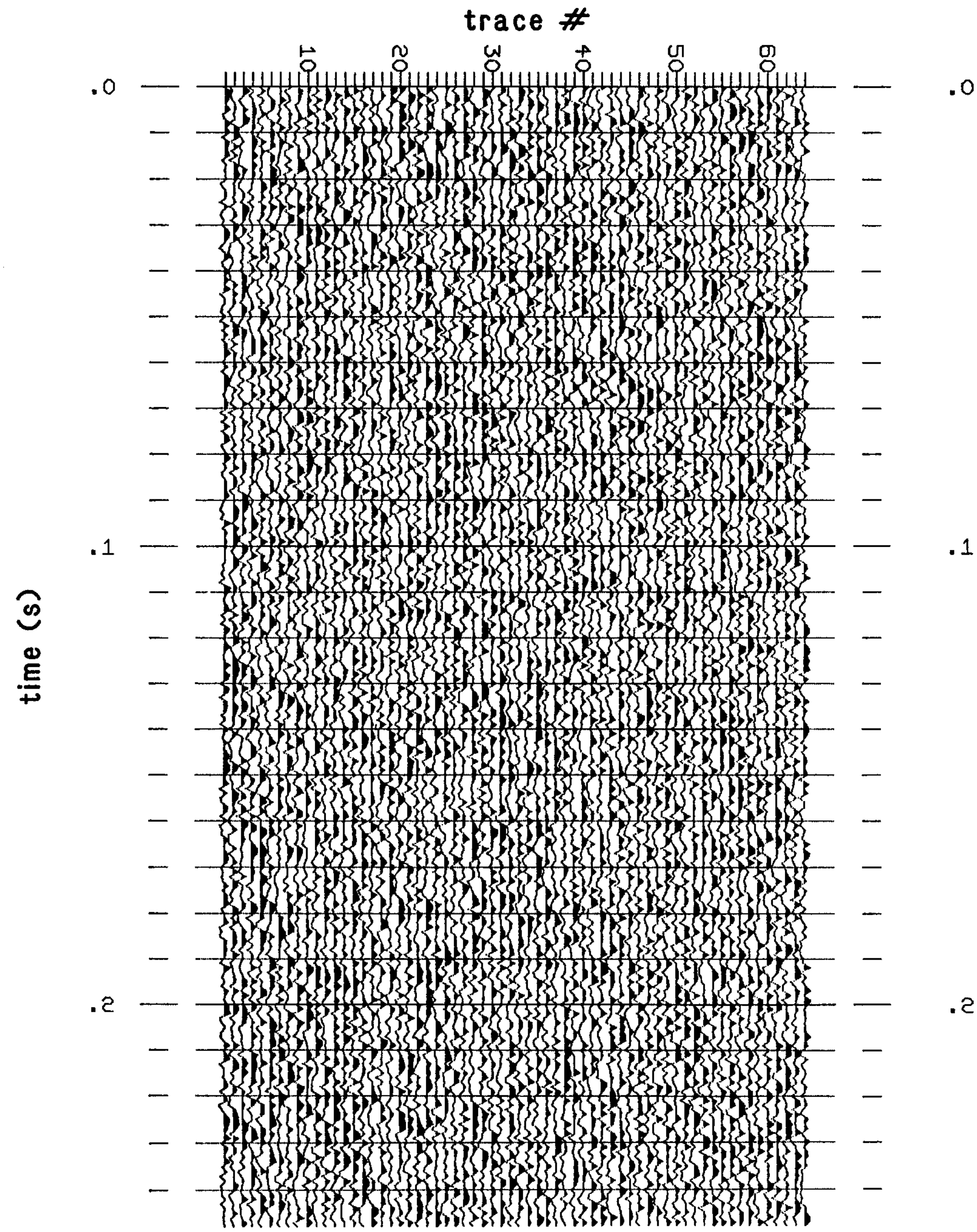


Figure 2

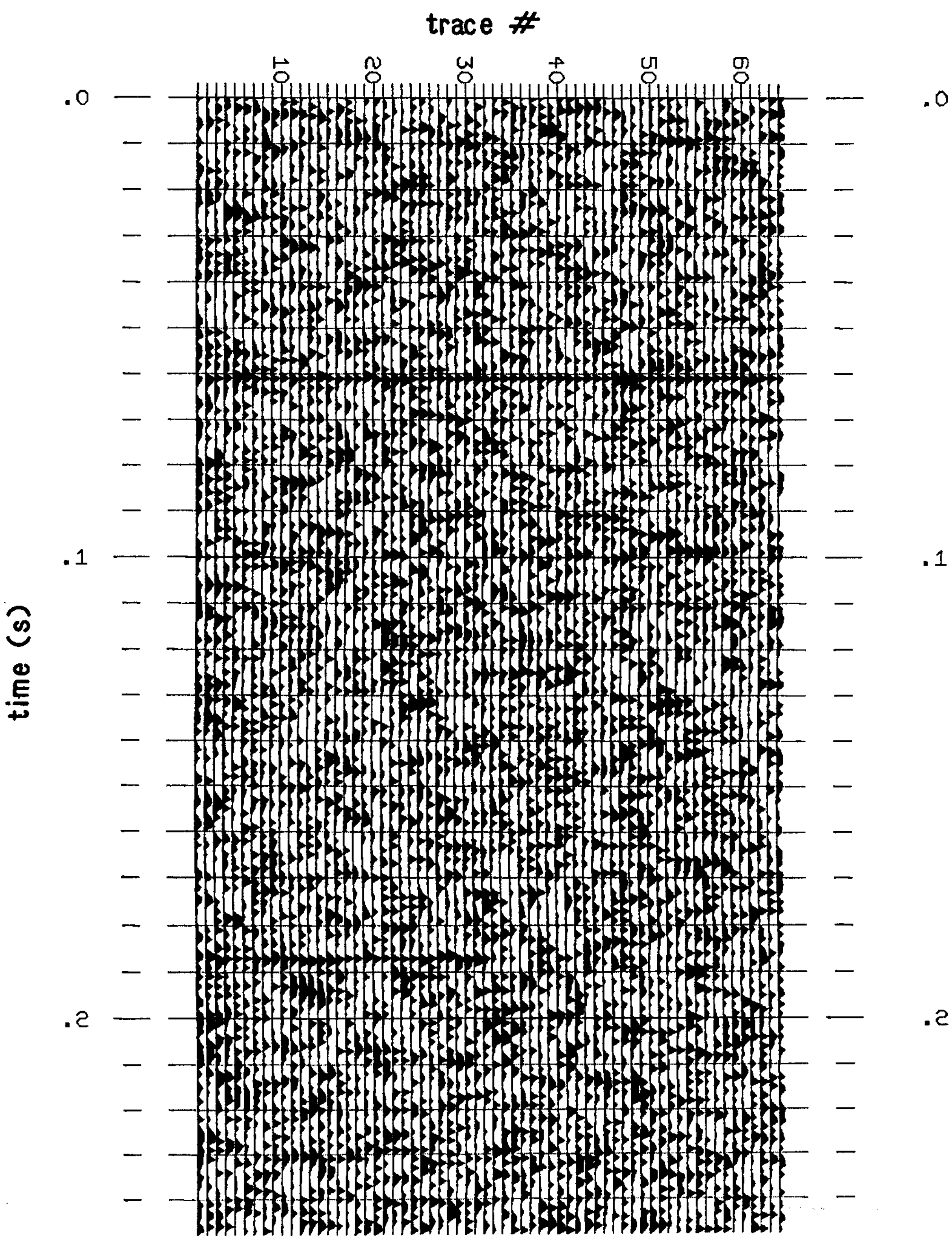


Figure 3

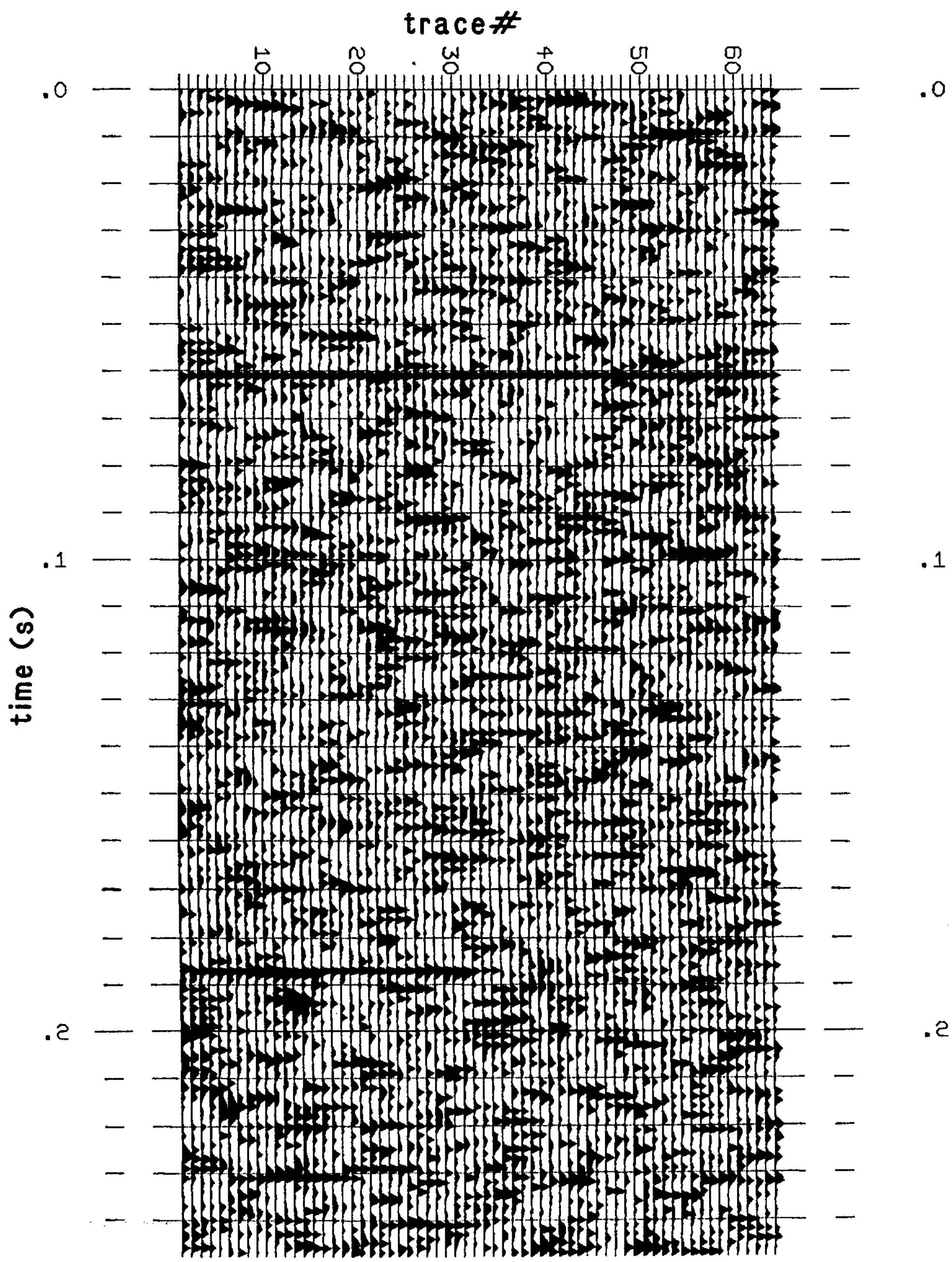


Figure 4

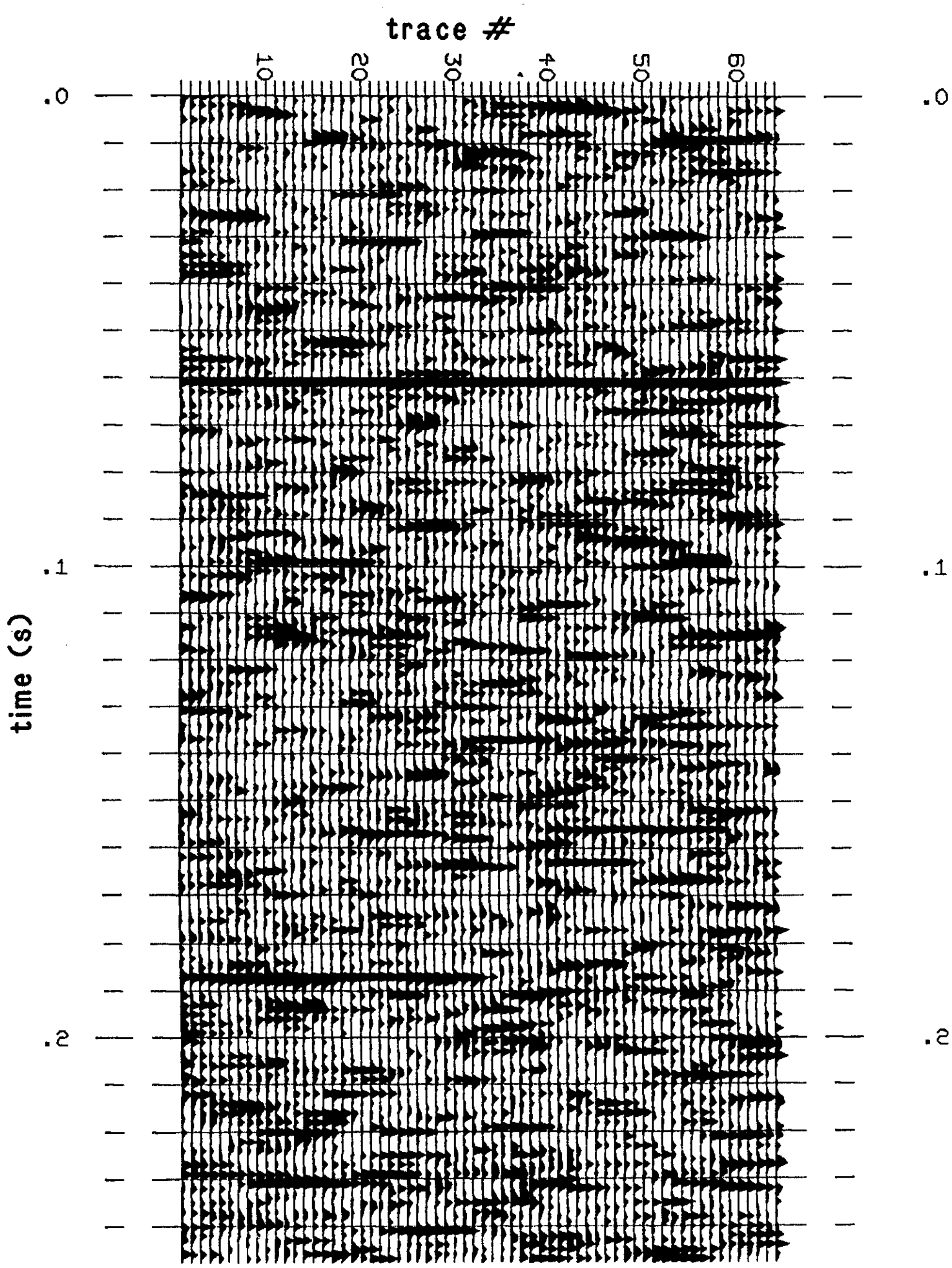


Figure 5

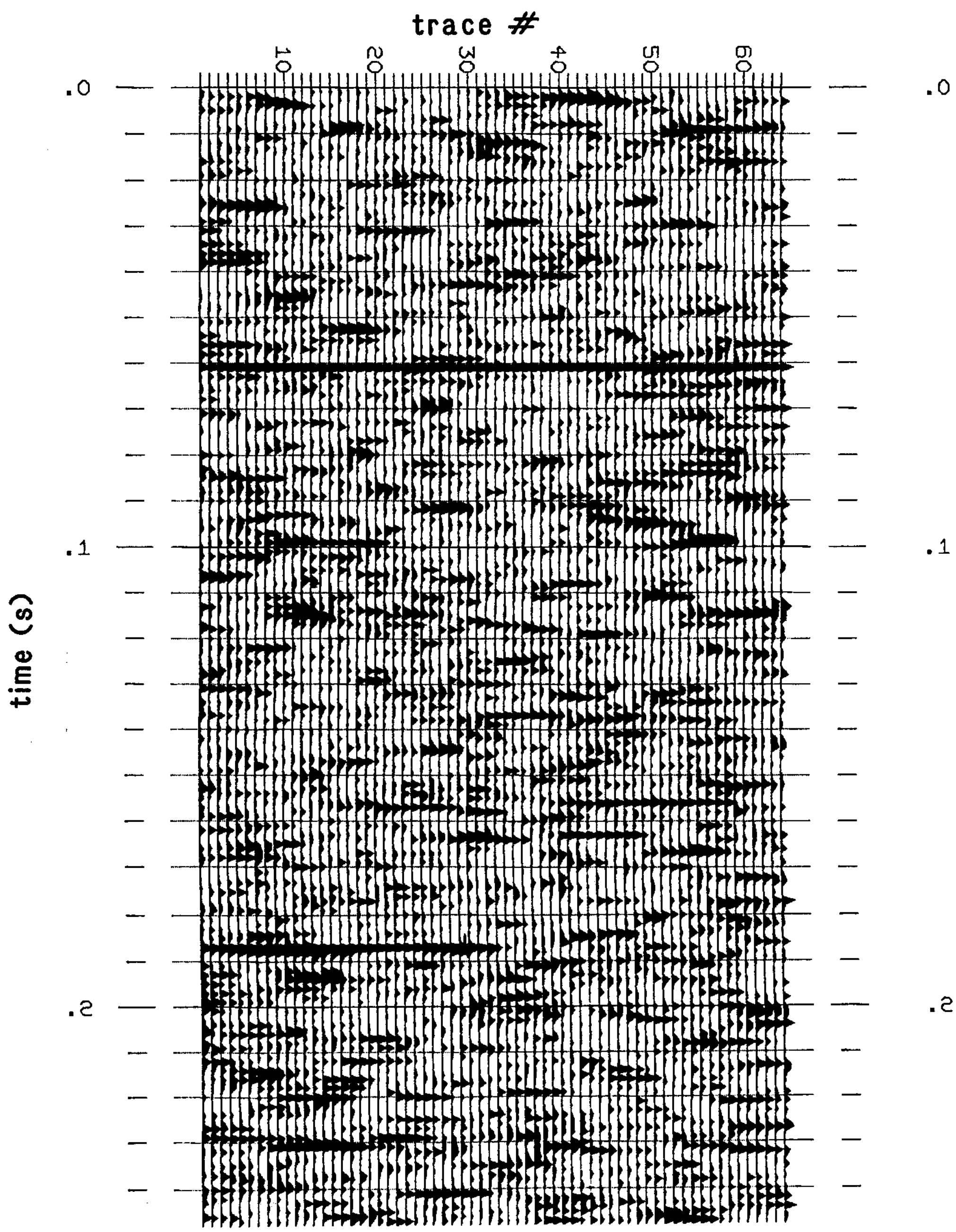


Figure 6

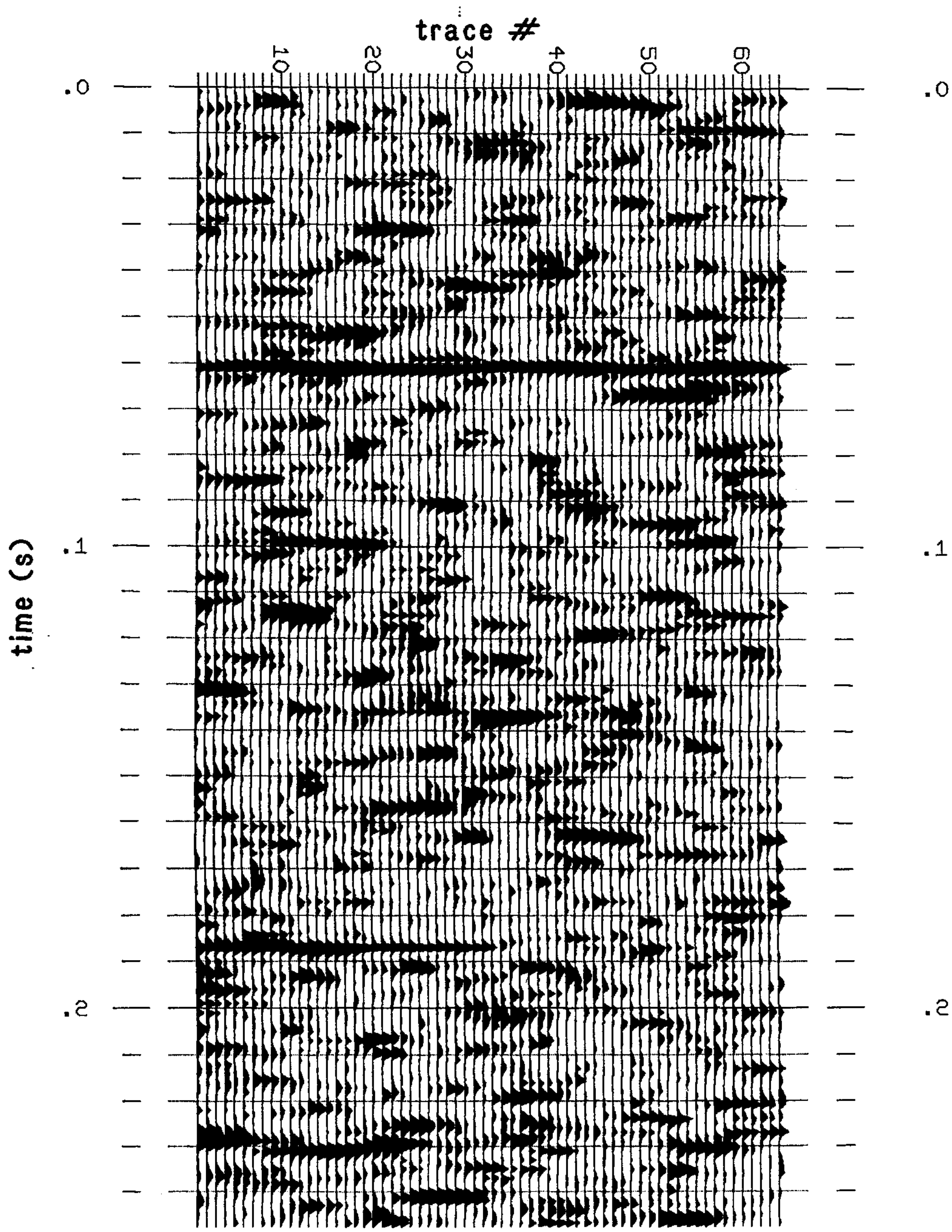


Figure 7

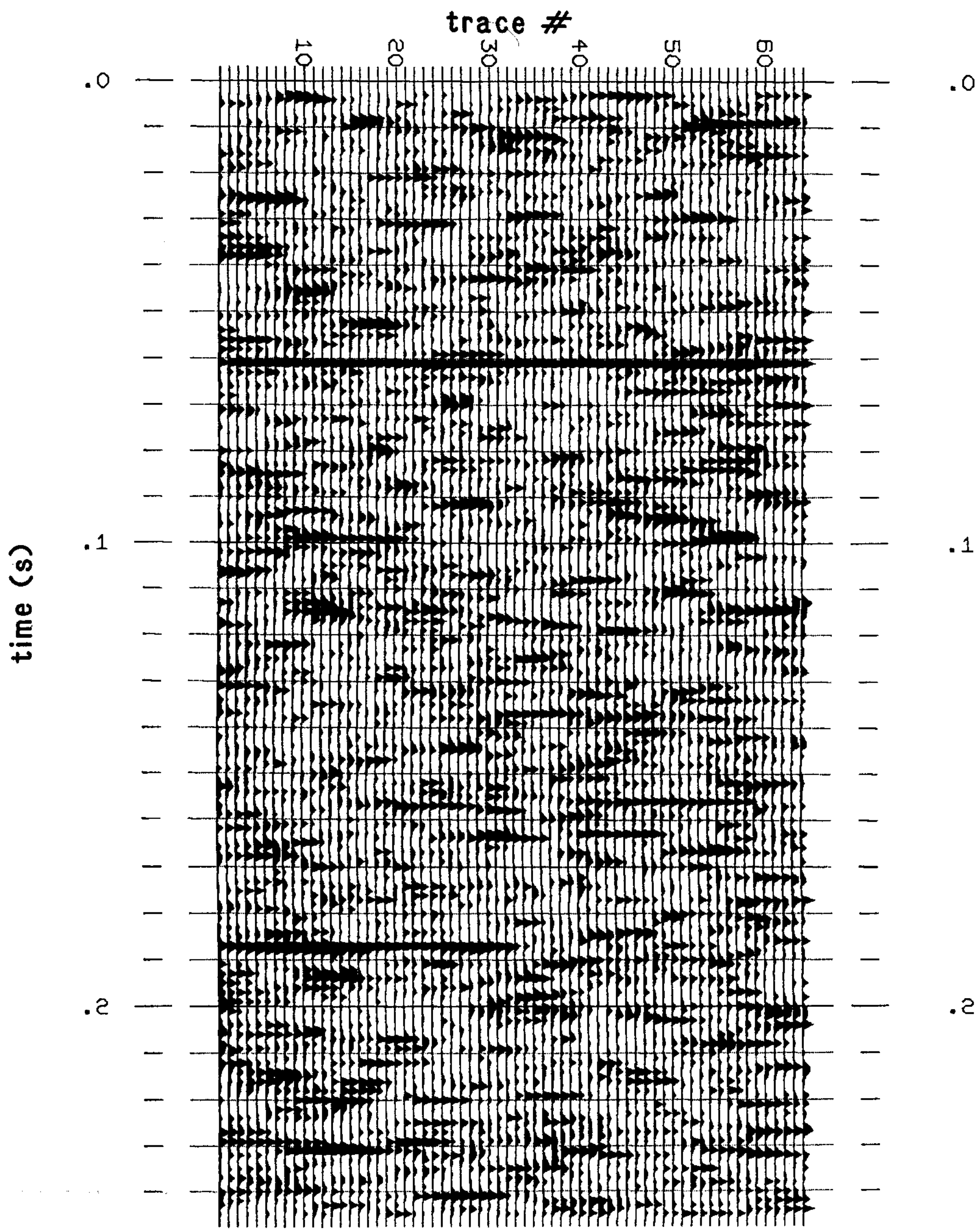


Figure 8

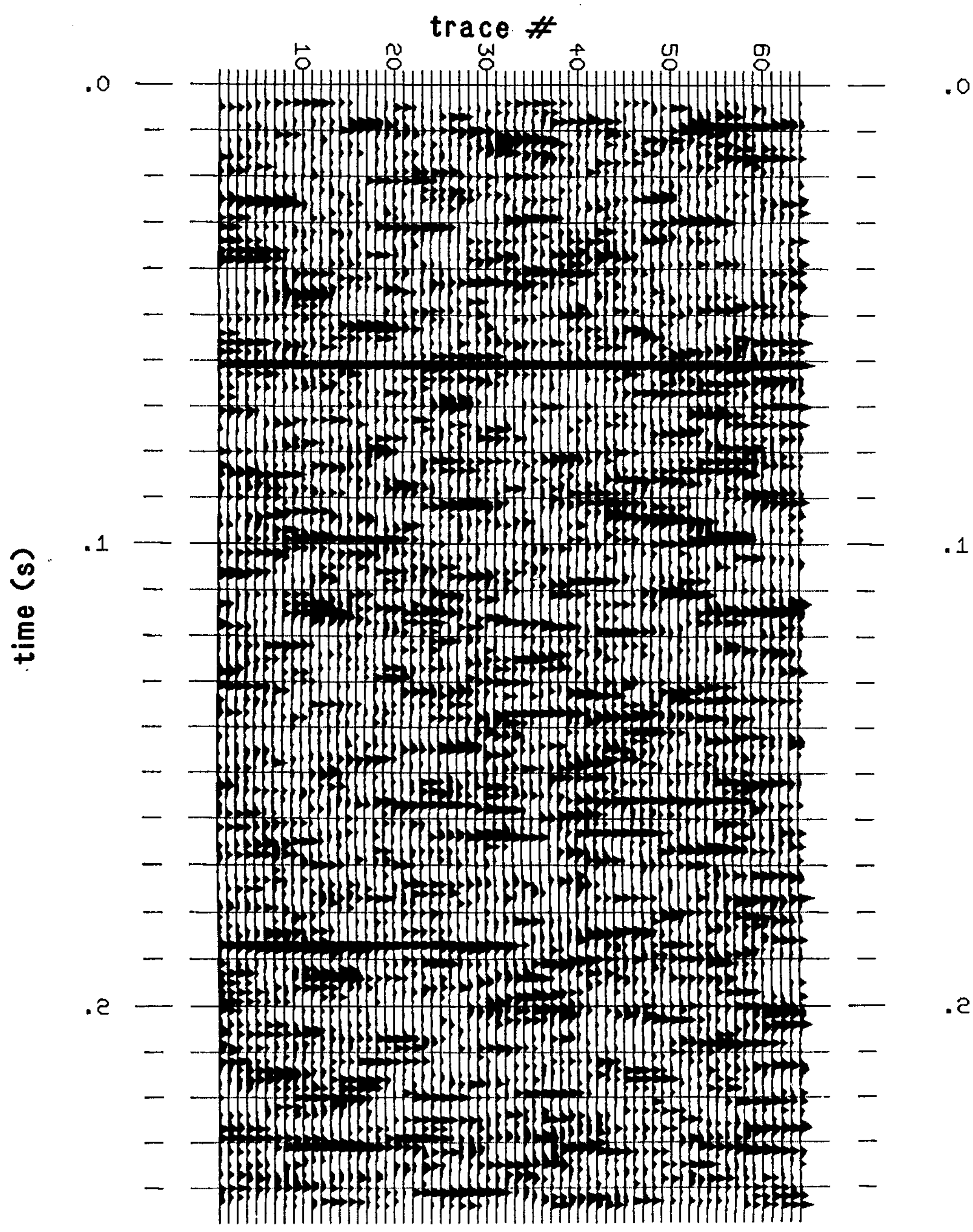


Figure 9

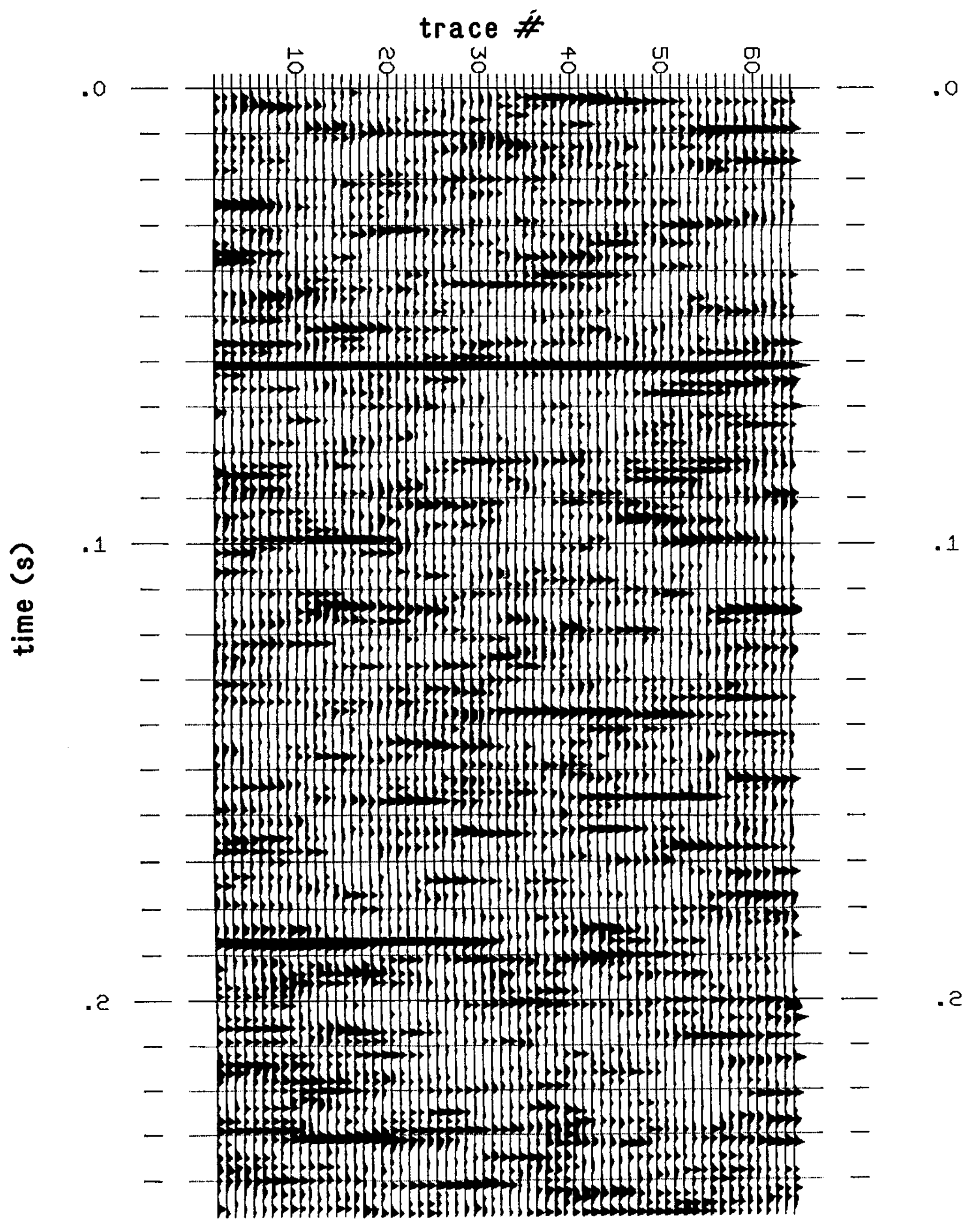
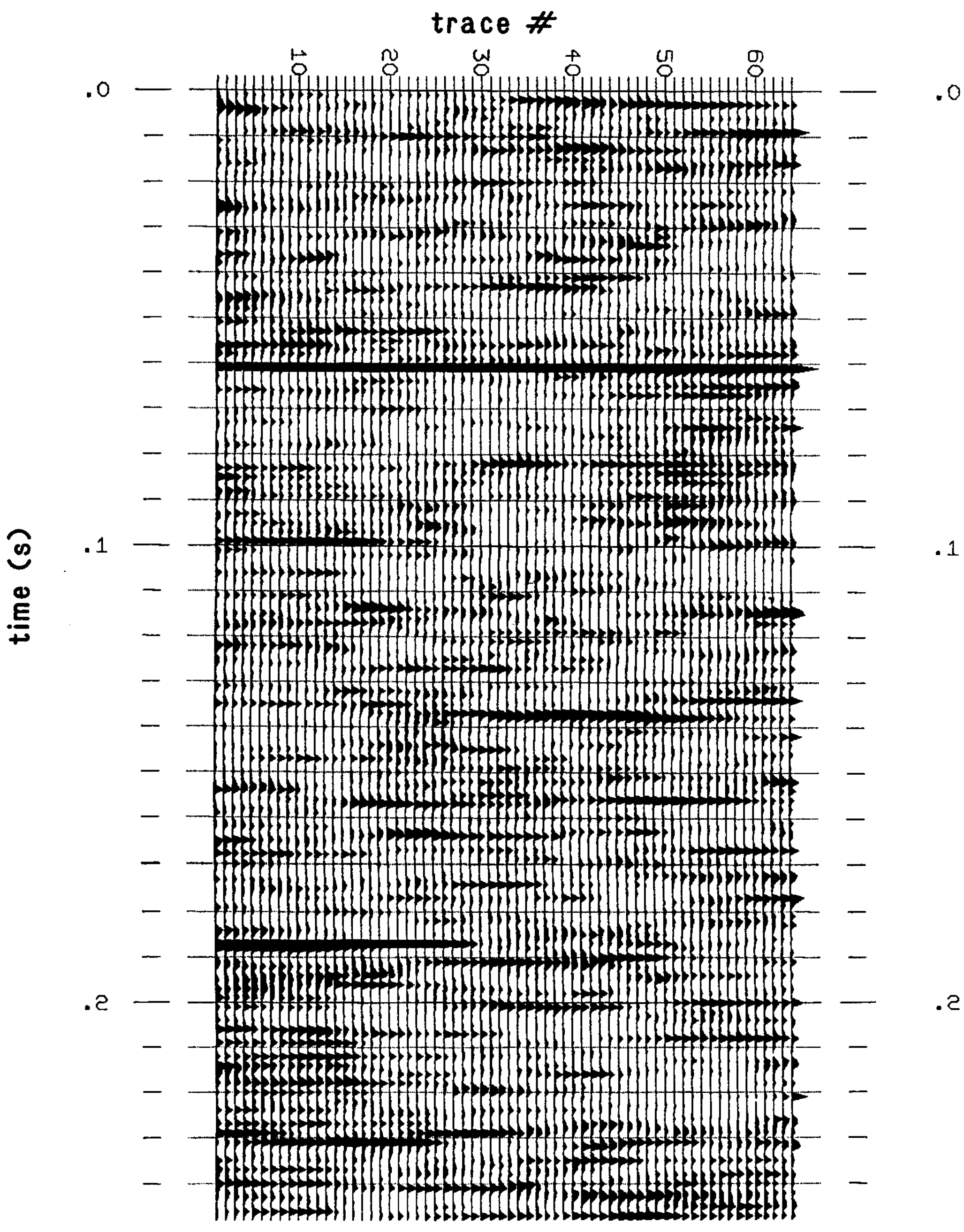


Figure 10



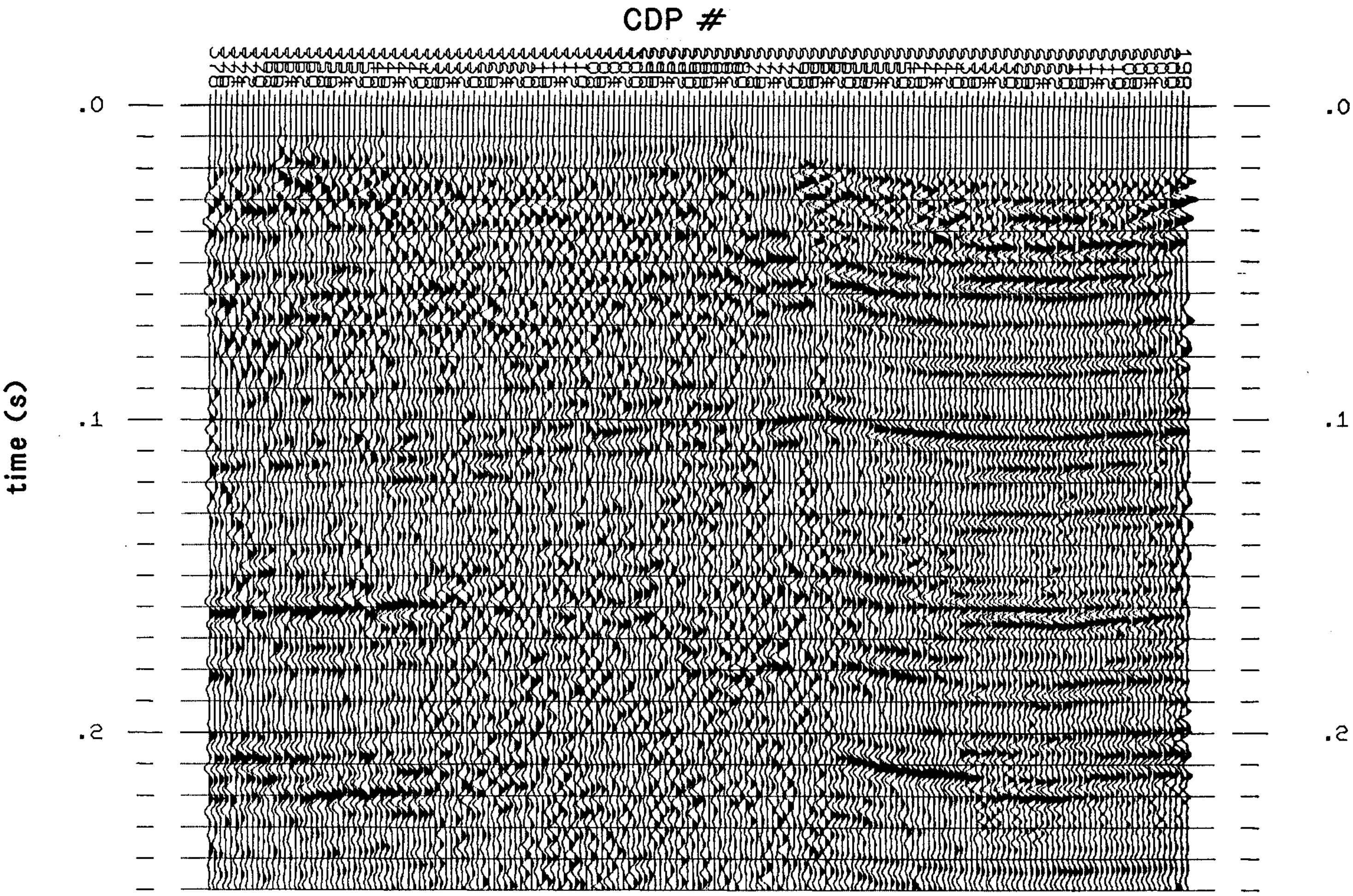
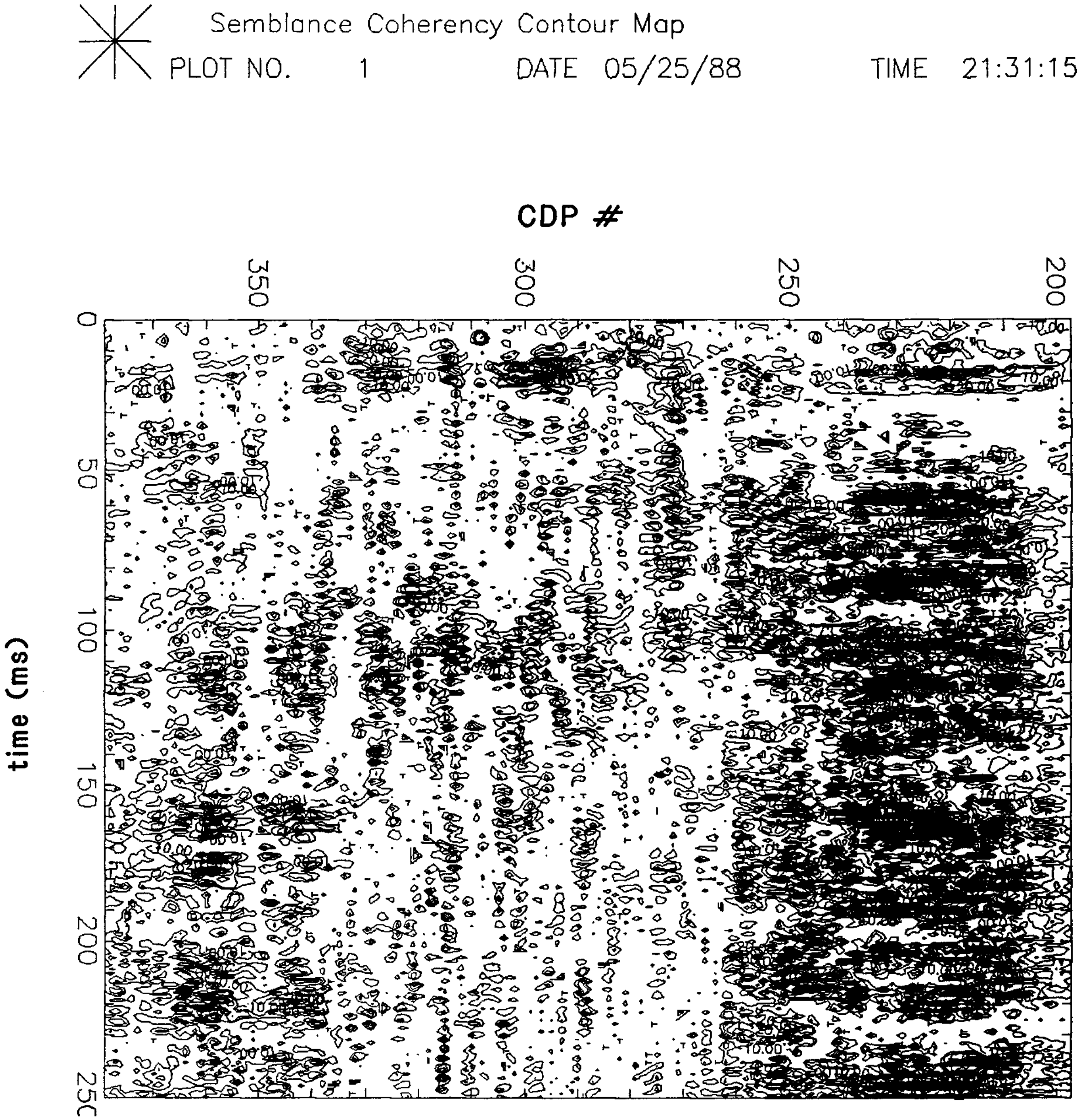


Figure 11



Semblance Coherency Contour Map
PLOT NO. 1
DATE 05/25/88
TIME 21:31:15
CDP #
350
300
250
200
time (ms)
0
50
100
150
200
250

Figure 12

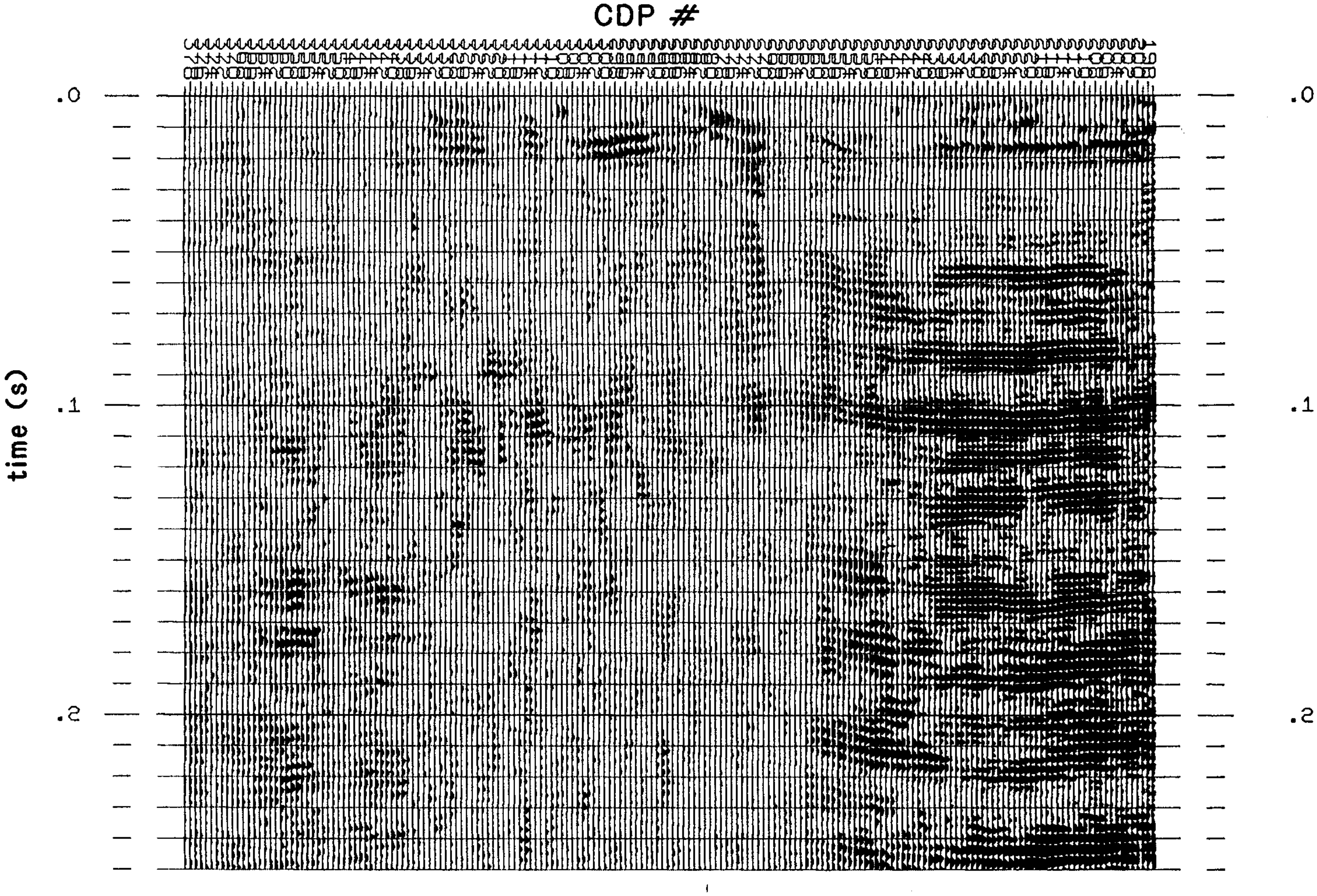


Figure 13

Semblance Coherency Contour Map

PLOT NO. 1 DATE 05/25/88 TIME 21:50:41

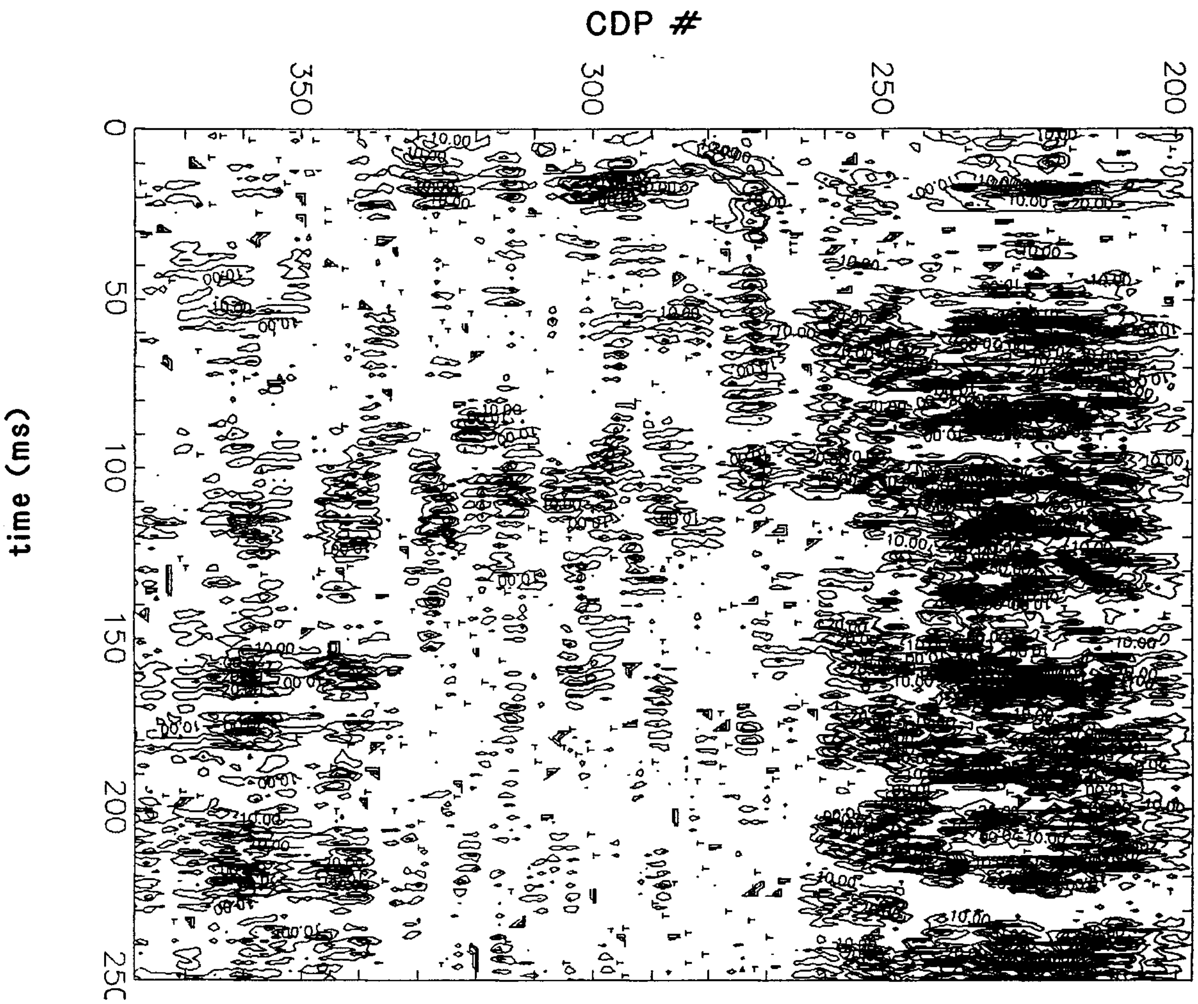


Figure 14

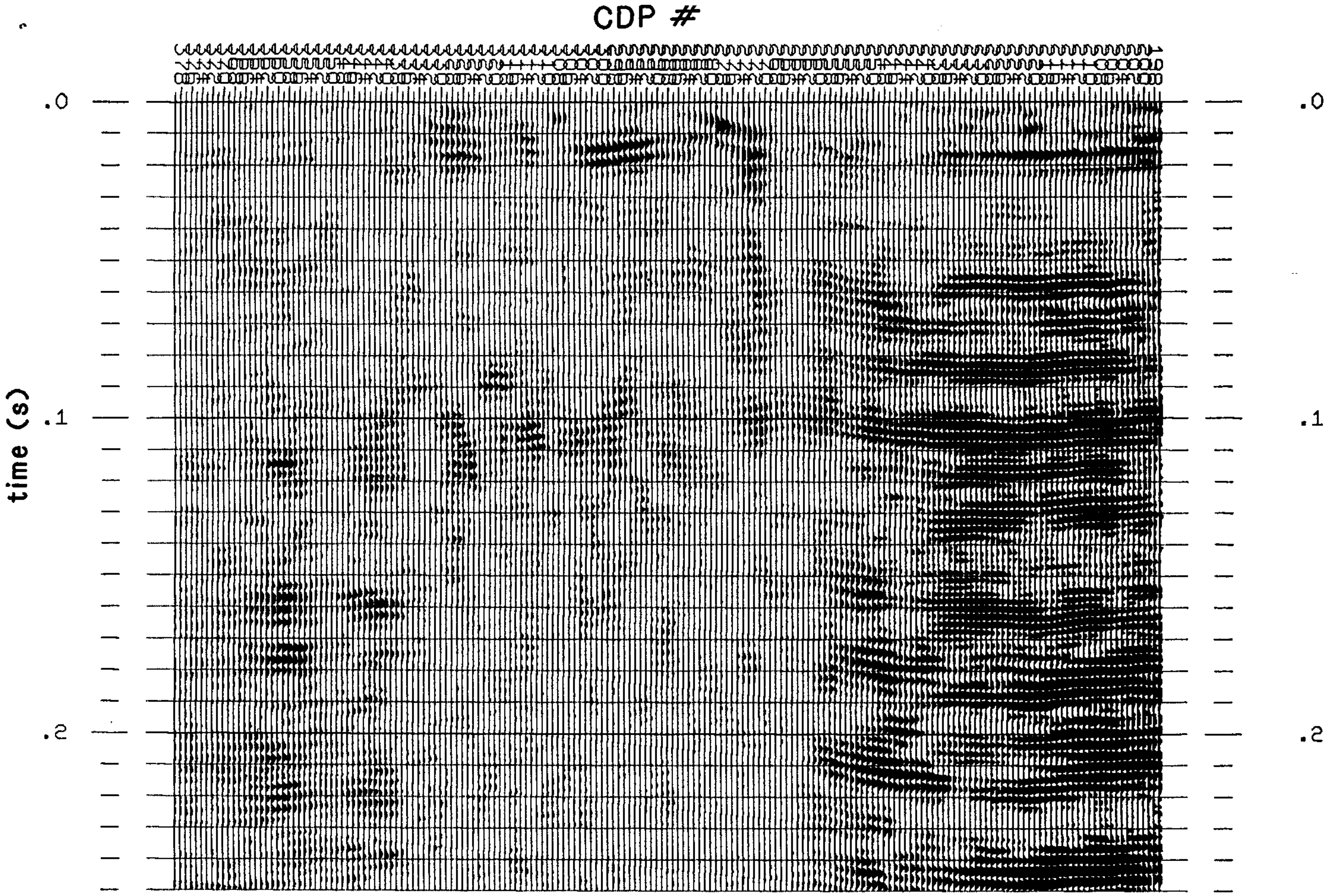


Figure 15

Semblance Coherency Contour Map

PLOT NO. 1 DATE 05/25/88 TIME 23:18:31

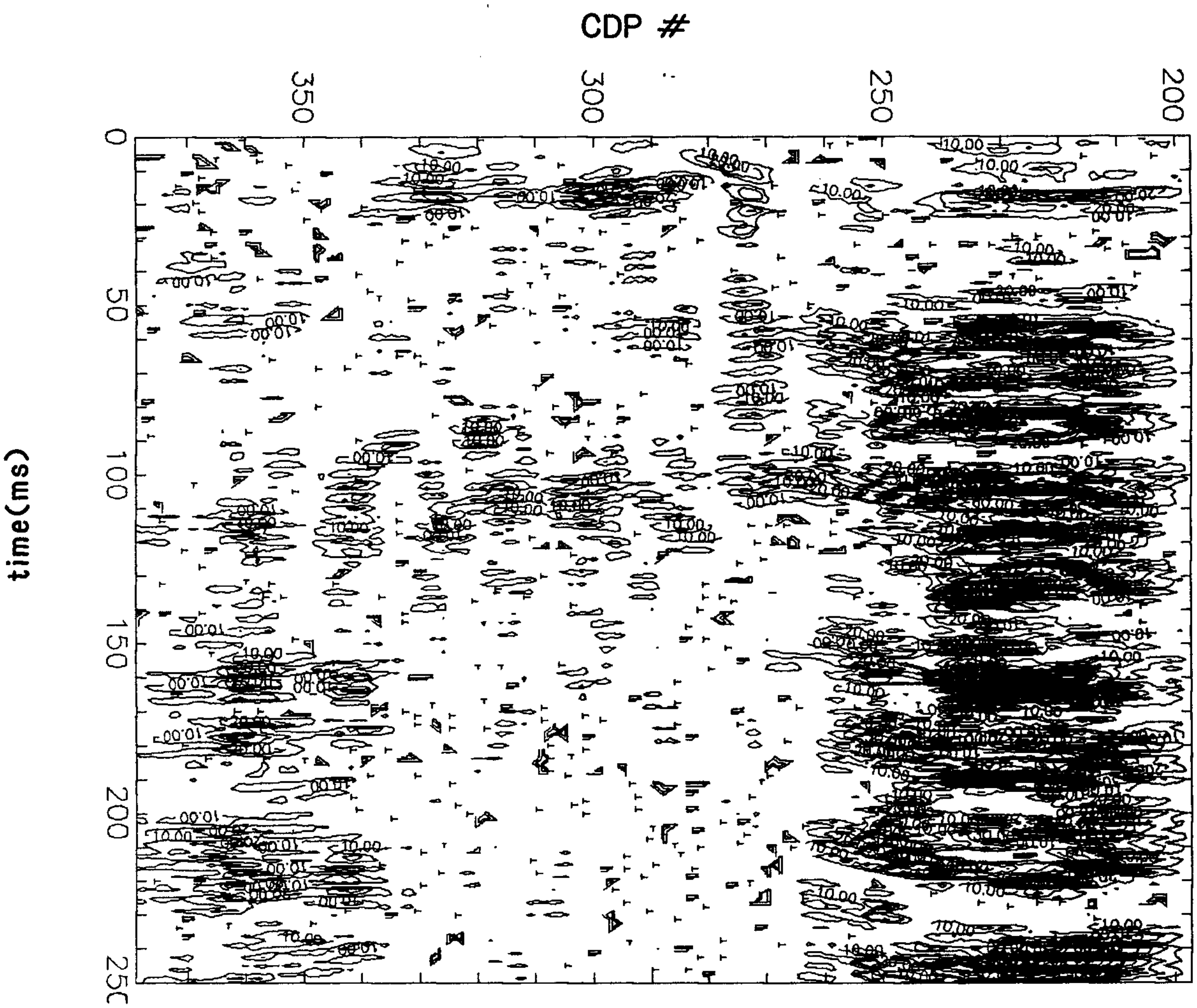


Figure 16

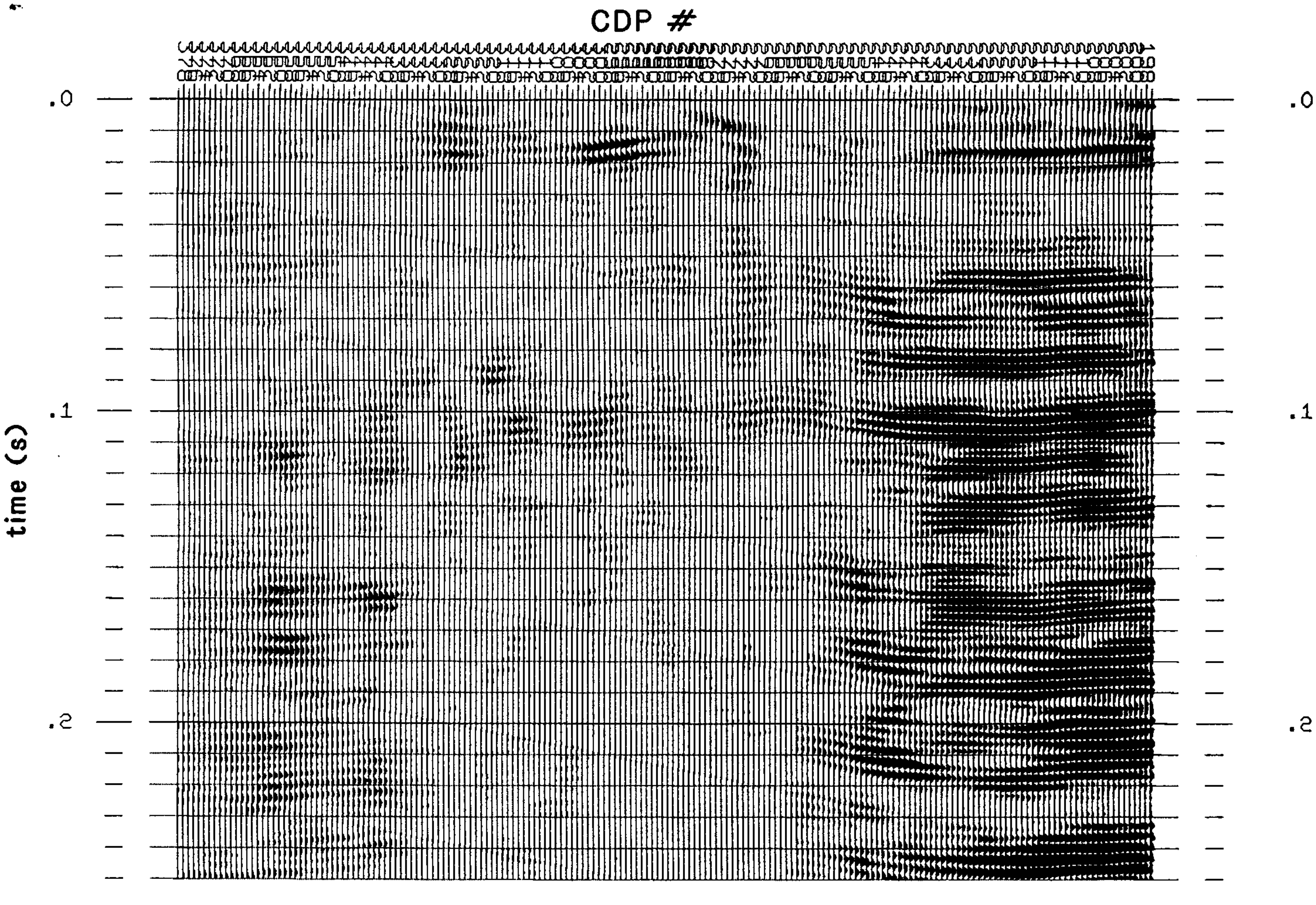


Figure 17

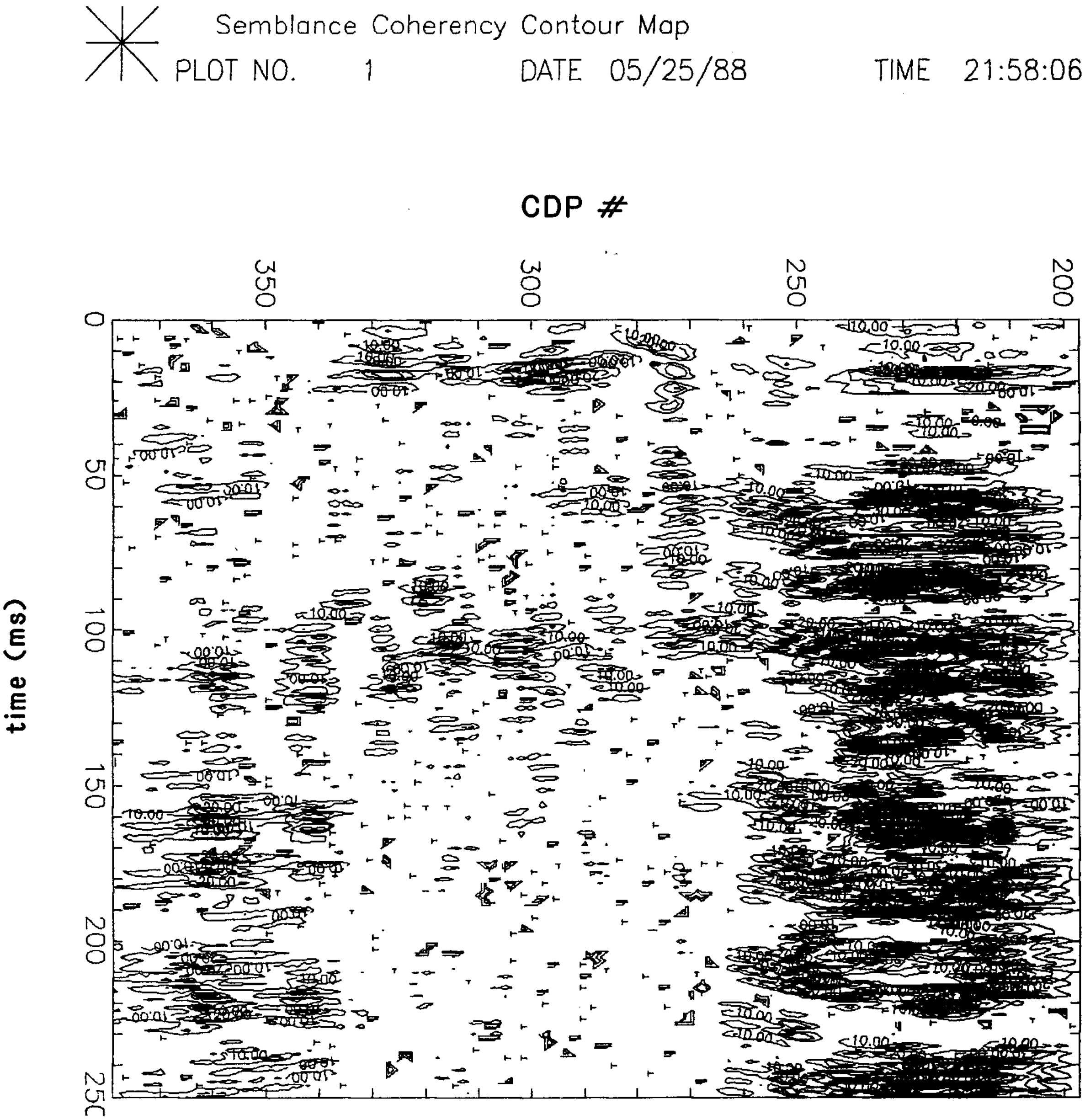

Semblance Coherency Contour Map
PLOT NO. 1
DATE 05/25/88
TIME 21:58:06
CDP #
350
300
250
200
time (ms)
0
50
100
150
200
10.00
20.00

Figure 18

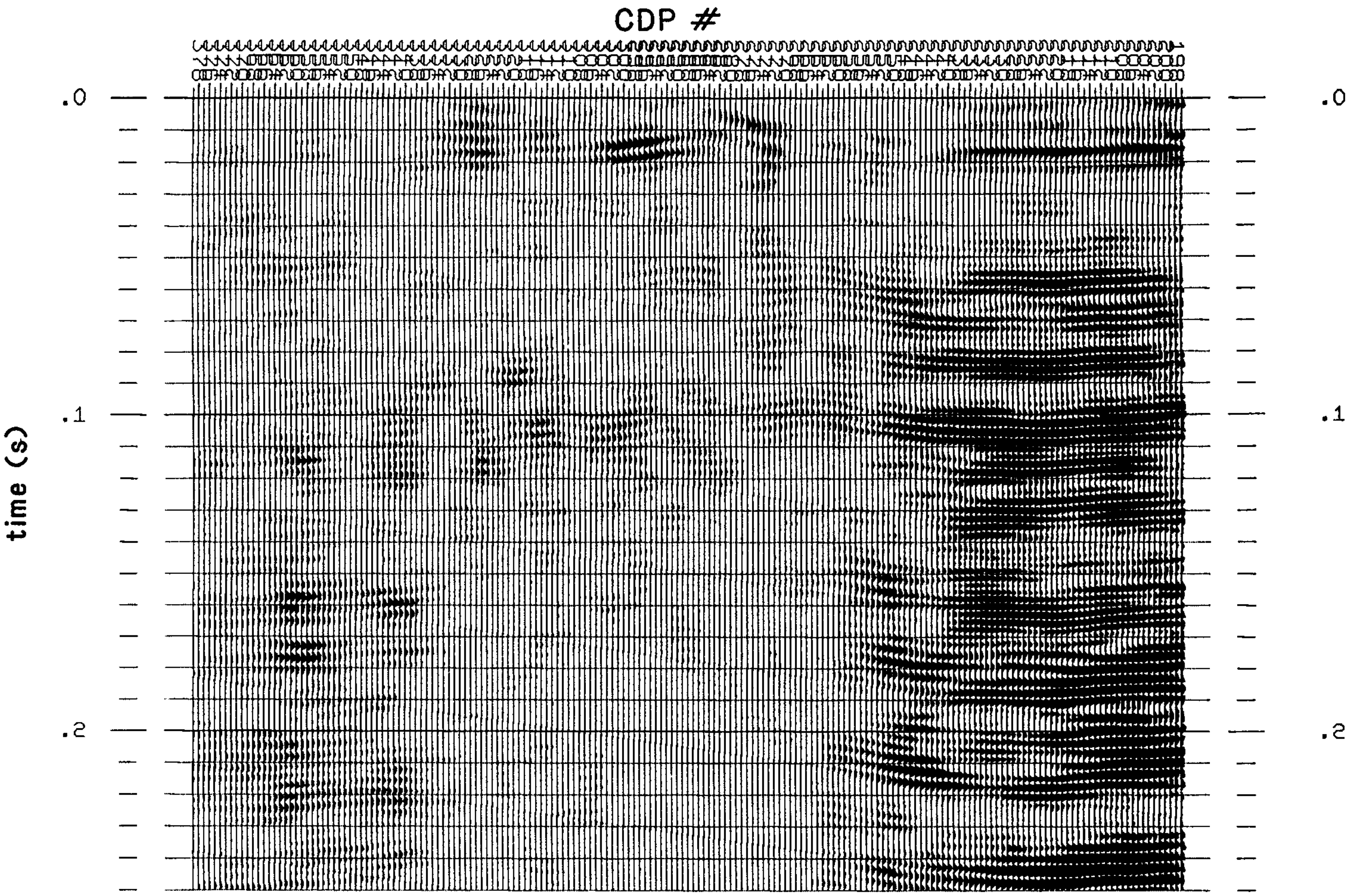


Figure 19

Semblance Coherency Contour Map

PLOT NO. 1 DATE 05/25/88 TIME 22:10:41

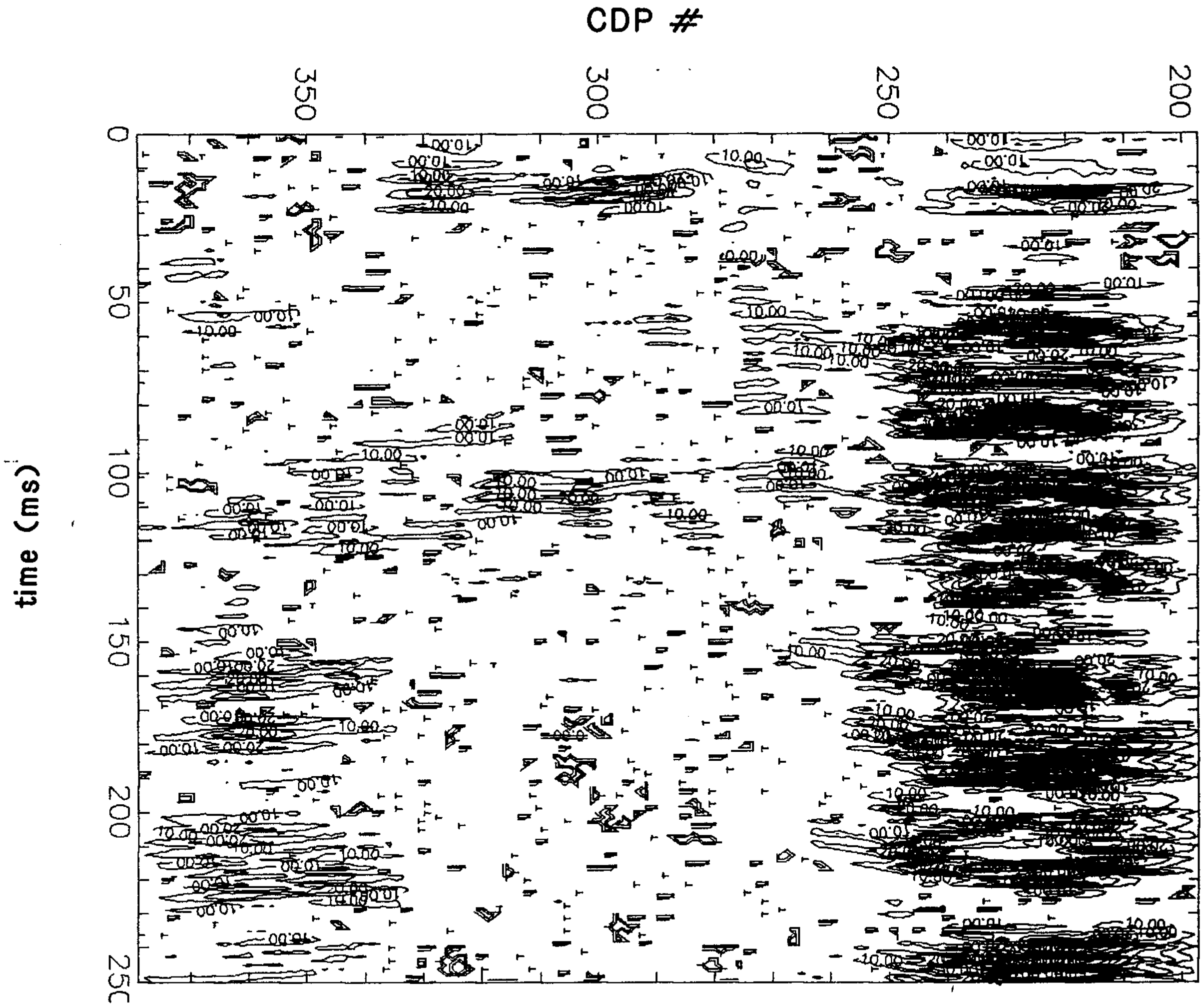


Figure 20

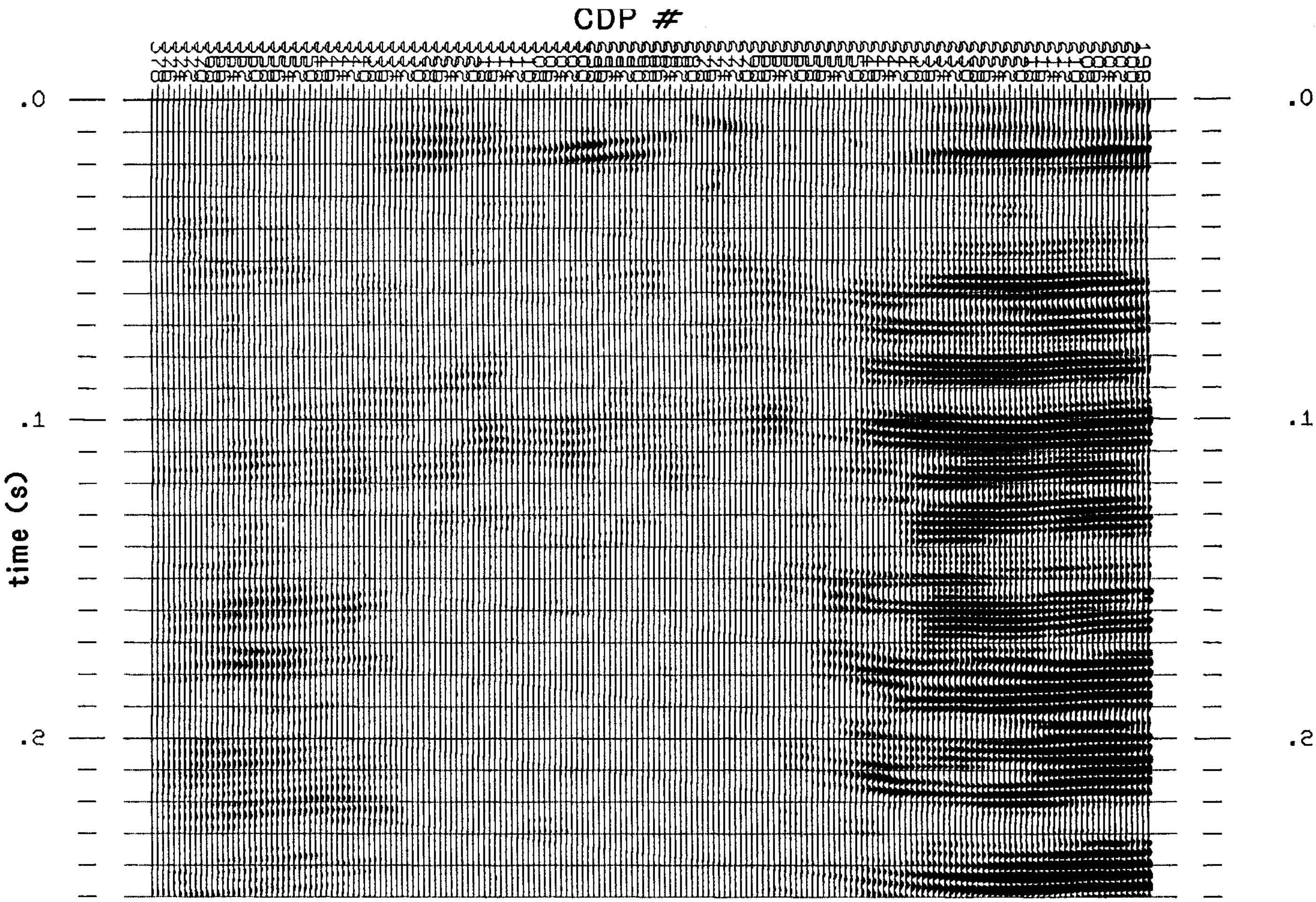


Figure 21

Appendix

```
      program sn
*********************************************************
c     sn.f77  version: 1.0
c     written by Andrew J. Kalik 4/21/88
c
c     This program computes signal-to-noise content
c     based upon the semblance coherency statistic within
c     a CDP window m+1 and sample interval n.  The output
c      file given by the user must be plotted using SPEX
c     software.  The output file beginning with "C" must
c     be run through SURFACE 2 for contour plots.
c
c     link with :udd:geophysics:tread.ob
c     Link with MTOP > 15
c
c     Necessary parameters provided by the user:
c
c     n     = Window Length
c     m     = Number of CDP's to Sum Over
c     time1 = Start Time
c     time2 = End Time
c     dt    = Sampling frequency for contour plots
c     beg   = Beginning CDP
c     end   = Ending CDP
c     st    = Sampling rate of original data
c
c     Computed parameters:
c
c     fileout = Output file for contour maps
c     nsamp   = Number of samples per trace
c     incre   = Sample interval
c     len     = Record length
c     t1      = first sample
c     t2      = last sample
c
c     Unstacked data is preferable!
c
*********************************************************
c
c    Dimension and Initialize Variables
c
      parameter (is=150,if=400)
      real*4 amp(1000,is:if),den(1000,is:if)
      real*4 ss(1000,is:if),s(1000,is:if)
      real*4 hin(1000),hamp(1000,1800)
      real xmax,c
      integer*2 j,reel1(1600),k,reel2(200),trh(120)
      integer*2 nsamp,itype,incre
      integer n,t1,t2,dt,size(is:if),time1,time2,st
      integer cdp,top,bot,a,b,len,mid,m,mm,kount,count
```

```
      integer beg,end
      character*30 datain,dataout,fileout
c
c    Call Subroutine IO
c
      call io(n,m,time1,time2,dt,beg,end,datain,dataout)
      fileout=dataout//"C"
c
c    Open Data Files
c
      open(15,file=fileout,status='fresh',recfm='ds',
     +iointent="output",maxrecl=9900)
      open(1,file=datain,iointent='input',mode='binary',
     +recfm='dynamic',form='unformatted')
      open(2,file=dataout,iointent='output',
     +mode='binary',recfm='dynamic',form='unformatted')
c
c    Read and Write Reel Header
c
      read(1) j,(reel1(i),i=1,j),k,(reel2(i),i=1,k)
      write(2) j,(reel1(i),i=1,j),k,(reel2(i),i=1,k)
c
c    Determine the Number of Samples, Sample Inteval,
c    Record Length, and Sampling Extremes
c
      nsamp=reel2(11)
      incre=reel2(9)
      itype=reel2(13)
      a=nsamp
      b=incre
      c=float(b)/1000.
      len=int(a*c)
      if (time2 .eq. 0) time2=len
      t1=n/2+1+(time1/c)
      t2=(time2/c)-n/2
      st=1
      cdp=1+end-beg
      j=120+2*nsamp
c
c    Set Matrices and Array Values to Zero
c
      do 10 ii=beg-m,end+m
        size(ii)=0
        do 20 kk=t1,t2
          den(kk,ii)=0.
          amp(kk,ii)=0.
          s(kk,ii)=0.
          ss(kk,ii)=0.
20      continue
10    continue
      do 22 jj=1,1800
```

```
        do 25 kk=t1,t2
          hamp(kk,jj)=0.
25      continue
22    continue
c
c    Read Traces and Accumulate Matrices
c
      count=0
30    call tread(trh,hin,nsamp,itype,1,ierr)
        if (ierr .ne. 0)  goto 80
        if (trh(12) .lt. beg .or. trh(12) .gt. end)
     +goto 70
        cdp=trh(12)
c
c    Keep Track of the Fold (size) of each CDP
c
        size(cdp)=size(cdp)+1
        count=count+1
        do 35 kk=t1,t2,st
          hamp(kk,count)=hin(kk)
          if (st .ne. 1) hamp(kk-1,count)=(hin(kk)+
     +hin(kk-2))/2.
35      continue
70    goto 30
c
c    Last Trace has been Read, HAMP Matrix is Complete
c
80    continue
c
c    Compute Moving Pointers for First and Last Trace
c    within the Window (m) for each CDP
c
      do 40 ii=beg,end
        mid=0
        do 41 mm=beg,ii
          mid=mid+size(mm)
41      continue
        top=mid
        do 42 mm=ii-m/2,ii
          top=top-size(mm)
42      continue
        bot=mid
        do 43 mm=ii,ii+m/2
          bot=bot+size(mm)
43      continue
C
c      Compute Effective Numerator and Denominator of
c      Semblance Statistic within the Sample and CDP
c      Trace Windows
c
        do 50 kk=t1,t2
```

```
          do 60 jj=kk-n/2,kk+n/2
            do 65 ll=top,bot
              amp(kk,ii)=amp(kk,ii)+hamp(kk,ll)
              den(kk,ii)=den(kk,ii)+hamp(kk,ll)**2
65          continue
60        continue
50      continue
40    continue
c
c    Compute Semblance Matrix and Largest Value for
c    Normalization
c
      xmax=0.
      do 90 ii=beg,end
        do 100 kk=t1,t2
          if (den(kk,ii) .eq. 0.) then
           s(kk,ii)=0.
           goto 100
          endif
          s(kk,ii)=amp(kk,ii)**2/(den(kk,ii)*count)
          xmax=amax1(xmax,s(kk,ii))
100     continue
90    continue
c
c    Normalize Semblance Matrix for Wiggle Plots
c
      do 115 ii=beg,end
        do 120 kk=t1,t2
          ss(kk,ii)=100.*s(kk,ii)/xmax
120     continue
        trh(6)=ii
        trh(88)=1
        write(2) j,trh,(ss(kk,ii),kk=1,nsamp)
115   continue
c
c    Compute Normalized Matrix for Contour Plots
c
      do 105 ii=end,beg,-1
        kount=0
        do 95 ik=t1,t2,dt
          kount=kount+1
          ss(kount,ii)=100.*s(ik,ii)/xmax
95      continue
        write(15,110) (ss(nn,ii),nn=1,kount)
110     format(1x,200f5.1)
105   continue
c
      stop
      end
**********************************************************
      subroutine io(n,m,time1,time2,dt,beg,end,datain,
```

```
     +dataout)
      integer n,beg,end,m,mm,time1,time2,dt
      real res
      character*30 datain,dataout
c
c    Retrieve Required Parameters Interactively
c
      write(*,1)
1     format(/,10x,"Enter input pathname: ",/)
      read(*,2) datain
2     format(a30)
      write(*,5)
5     format(/,10x,"Enter output pathname: ",/)
      read(*,2) dataout
9     write(*,10)
10    format(/,10x,"Enter window length (n): ")
      write(*,15)
15    format(10x,"(Default = 1)",/)
      read(*,20) n
20    format(i10)
      if (n .gt. 9) then
       write(*,25)
25     format(/,10x,"WINDOW IS TOO LARGE",/)
       goto 9
      endif
      if (n .eq. 0) n=1
      write(*,30)
30    format(/,10x,"Enter number of CDP's (m): ",/)
      read(*,20) mm
      res=float(mm)/2.
      if (frac(res) .ne. 0.) then
       m=mm-1
      else
       m=mm
      endif
      write(*,40)
40    format(/,10x,"Enter start time (time1): ",/)
      write(*,45)
45    format(10x,"(Default = 0)",/)
      read(*,20) time1
      if (time1 .eq. 0) time1=0
      write(*,50)
50    format(/,10x,"Enter end time (time2): ",/)
      write(*,55)
55    format(10x,"(Default = End of Record)",/)
      read(*,20) time2
      write(*,65)
65    format(/,10x,"Enter sample interval (dt): ",/)
      write(*,75)
75    format(10x,"(Default = 1)",/)
      read(*,20) dt
```

```
      if (dt .eq. 0) dt=1
      write(*,60)
60    format(/,10x,"Enter beginning CDP (beg): ",/)
      read(*,20) beg
      write(*,70)
70    format(/,10x,"Enter ending CDP (end): ",/)
      read(*,20) end
      return
      end
```