SPEDIT:

Spike Edit of Seismic Data

Ralph W. Knapp

AUGUST 1, 1991

KGS Open-File Report 91-29

SPEDIT: Spike Edit for Seismic Data

INTRODUCTION

Seismic data often contain noise spike due to electrical noise from within the system or from other static discharges such as lightning. Air-coupled waves are another source of such noise. The effects of these spikes propagates throughout the processing and overprints good results, makes complete analysis by automatic means impossible, or, in some cases, creates events that might be misinterpreted. The typical means of dealing with spike noise is tedious editing of the data which is both time-consuming and detrimental to the data due to loss of information in the edit process. Editing is like surgery; usually good tissue (data) is taken with the bad to ensure total removal. Careful removal of only the noise portions is typically too tedious to do precisely, and, as a result, less precise values are commonly used, which takes out some good data. Frequently whole traces or records are removed when noise spikes are bad, even when the record may contains some good information.

SPEDIT was designed with the specific intent of attenuating the air-coupled wave. It works as well with noise spikes, although parameter design may be different for the two intents. Figures 1 and 2 compare field file data before and after application of SPEDIT.

FUNCTION

SPEDIT works by using the envelope of the seismic energy, commonly referred to as instantaneous amplitude or reflection strength (Figure 2). It searches for peak values on the envelope. When peaks are found, it averages the energy over a prescribed window centered on the peak (Figure 3b). The width of this window is 2*WIDTH, where WIDTH is one of two parameters of the process. When the value of the peak is greater than the average window energy by a multiplier called FACTOR (the other parameter), it then searches backward and forward from the peak to find the edges of the peak (Figure 3b). Then between the bounds of the peak it re-scales the data so that the instantaneous energy of a point is the linear interpolation of the values at the boundaries (Figure 3c). In fact, the display of Figure 3c is never actually generated, as the scaling is done directly on the data itself (Figure 3d). Thus, the noise spike is reduced to have an amplitude value that does not exceed the background value of the energy of the seismic trace about the region of the spike. Any signal that is within the region of the spike may be reduced in amplitude, but it is not removed. Further removal of the residual spike is accomplished with additional CMP processing of the data, specifically, the stacking or averaging of redundant measurements.

OPERATION

SPEDIT depends on only two parameters, WIDTH and FACTOR. The design of the two is inter-related. For instance, if WIDTH is large, FACTOR can be larger because the influence of the spike on average energy within the window is reduced as more points are averaged in. It is advantageous to have FACTOR be large because it provides better discrimination against large amplitude reflection (data) responses. On the other hand, if WIDTH is too large, large amplitude reflections may start to be identified as noise spikes.

Ideally, WIDTH is design to be about one-half the width of the large amplitude reflection response (the shallow reflections). An instantaneous amplitude display may help determine this value. FACTOR is then designed to be large enough to pass good data but small enough to pick noise. This may require experimentation, especially when applied to the air-coupled wave which is not a true spike. The output list of SPEDIT helps determine this as it identifies peak values, average energy, and ratios of the two. That is, one experimental run with FACTOR too small by design will produce this information.

CONCLUSIONS

SPEDIT removes "spikey" noise from seismic data in a very efficient manner. Its use reduces man-hours of tedious editing, and its use minimizes damage to data on the record section.

APPENDIX 1: PROGRAM LISTING

```
      PROGRAM spedit
*
* SPIKE EDIT
* JULY 16, 1991
*
      integer*2 j,reel1(1600),k,reel2(200),trh(120)
      integer*2 nsamp,itype,incre,units
      real*4 hamp  (2000)
      COMPLEX A(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,*) ' SPEDIT  910716'
      write(6,*)' datain  = ', datain
      write(6,*)' dataout = ', dataout
      write(6,*)' * * * * *  '
      open(2,file=dataout,iointent='output',mode='binary',recfm='dynamic',form='unfo
      open(1,file=datain,iointent='input',mode='binary',recfm='dynamic',form='unform
*
      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)
      units=reel2(28)
      si=float(incre)/1000000.
      read (5,*) width,factor
      write(6,*)' nsamp = ',nsamp
      write(6,*)' itype = ',itype
      write(6,*)' incre = ',incre
      write(6,*)' units = ',units
      write(6,*)' si    = ',si
      write(6,*)' width = ',width,' (ms)'
      write(6,*)' factor= ',factor
      iw=IFIX(width/si/1000.+.5)
      PRINT *,'  IW  = ', IW,' (POINTS)'
      write(2)j,(reel1(i),i=1,j),k,(reel2(i),i=1,k)
      n=256
9     n=2*n
      if(n.lt.nsamp) go to 9
      PRINT *,' N = ',N
1     call tread (trh,hamp,nsamp,itype,1,ierr)
      if(ierr.ne.0) stop
      print *,trh(6),trh(8),trh(12),trh(14),trh(88) ,trh(15)
      if(trh(15).ne.1) go to 99       ! if dead trace
      do 90 i=1,n
90    a(i)=cmplx(0.,0.)
      do 10 i=1,nsamp
10    a(i)=cmplx(hamp(i),0.)
      call fouri(a,n,-1)
      do 11 i=n/2+1,n
11    a(i)=cmplx(0.,0.)
      do 12 i=2,n/2
12    a(i)=2.*a(i)
      a(1)=cmplx(0.,0.)
      call fouri(a,n,+1)
*     DO 999 I=2,NSAMP
*999  HAMP(I)=CABS(A(I))-CABS(A(I-1))
*     HAMP(1)=0.
```

```
*       GO TO 99
*
        i=iw
20      continue
        i=i+1
        if(i.gt.nsamp-iw) go to 22
        if(cabs(a(i)).lt.1.) go to 20
        if(cabs(a(i)).lt.cabs(a(i+1))) go to 20 ! find peak values
        if(cabs(a(i)).lt.cabs(a(i-1))) go to 20 ! find peak values
        WT=0.
        DO 220 II=I-IW,I+IW
220     WT=CABS(A(II))/FLOAT(2*IW+1) + WT
        if(factor*wt.gt.CABS(A(I))) go to 20          ! check size
        PRINT *,' I = ',I,' WT = ',WT,' A(I) = ',CABS(A(I)),' RATIO = ',CABS(A(I))/WT
        DO 201 II=I-1,I-IW,-1                     ! FIND INFLECTION POINTS
        I1=II
201     IF(CABS(A(II)).LT.WT.AND.CABS(A(II)).LE.CABS(A(II-1)))GO TO 211
211     CONTINUE
        DO 202 II=I+1,I+IW
        I2=II
202     IF(CABS(A(II)).LT.WT.AND.CABS(A(II)).LE.CABS(A(II+1)))GO TO 212
212     CONTINUE
        PRINT *,' I1 = ',I1,' I2 = ',I2!*******************************
        do 21 ii=I1,I2                            ! scale window
        X=CABS(A(I1))+FLOAT(II-I1)/FLOAT(I2-I1)*(CABS(A(I2))-CABS(A(I1)))
        wt=X/cabs(a(ii))
        hamp(Ii)=hamp(iI)*wt
21      continue
        i=i2
        go to 20
*
22      continue
99      CONTINUE
        j=2*nsamp+120
        write(2) j,trh,(hamp(i),i=1,nsamp)
        GO TO 1
        end
```

TABLE 1. Sample run of SPEDIT and sample input deck

execute:

X :UDD:GEOPHYSICS:SPEDIT, *input, list*

input:

RWK.FIELD	*data in*
RWK.FIELD.SPED	*data out*
20. 2.2	*WIDTH, FACTOR, respectively*

TABLE 2. Sample output list for SPEDIT

```
SPEDIT   910716
datain  = kv.ind.f.EDIT.FILT
dataout = kv.ind.f.EDIT.FILT.SPED
* * * * *
nsamp = 600
itype = 1
incre = 500
units = 0
si    = 5.E-04
width = 20. (ms)
factor= 2.2
 IW  = 40 (POINTS)
N = 1024
12  1  194  1  0  1
12  2  195  2  0  1
 I = 274 WT = 4027.09 A(I) = 10232.5 RATIO = 2.54092
 I1 = 269 I2 = 280
 I = 402 WT = 4776.05 A(I) = 10529.6 RATIO = 2.20467
 I1 = 386 I2 = 411
12  3  196  3  0  1
 I = 166 WT = 5881.82 A(I) = 13309.1 RATIO = 2.26275
 I1 = 160 I2 = 171
12  4  197  4  0  1
 I = 180 WT = 4754.87 A(I) = 18645.6 RATIO = 3.92138
 I1 = 170 I2 = 186
 I = 330 WT = 4221.11 A(I) = 11353.8 RATIO = 2.68977
 I1 = 311 I2 = 338
12  5  198  5  0  1
 I = 197 WT = 4476.5 A(I) = 10892. RATIO = 2.43316
 I1 = 190 I2 = 220
12  6  199  6  0  1
 I = 219 WT = 6923.37 A(I) = 19162.2 RATIO = 2.76776
 I1 = 206 I2 = 233
 I = 341 WT = 3407.58 A(I) = 9441.85 RATIO = 2.77083
 I1 = 336 I2 = 349
12  7  200  7  0  1
 I = 56 WT = 5114.5 A(I) = 11682.5 RATIO = 2.2842
 I1 = 47 I2 = 89
 I = 230 WT = 6455.52 A(I) = 19232.2 RATIO = 2.97919
 I1 = 207 I2 = 245
 I = 332 WT = 2835.48 A(I) = 9353.87 RATIO = 3.29886
 I1 = 325 I2 = 339
 I = 393 WT = 4363.48 A(I) = 10627.4 RATIO = 2.43554
 I1 = 381 I2 = 399
12  8  201  8  0  1
 I = 58 WT = 4604.59 A(I) = 10934.2 RATIO = 2.37462
 I1 = 48 I2 = 75
 I = 242 WT = 5736.69 A(I) = 22113.9 RATIO = 3.[illegible]
 I1 = 232 I2 = 249
```

PEAK LOCATION (po[illegible]

boundaries of peak

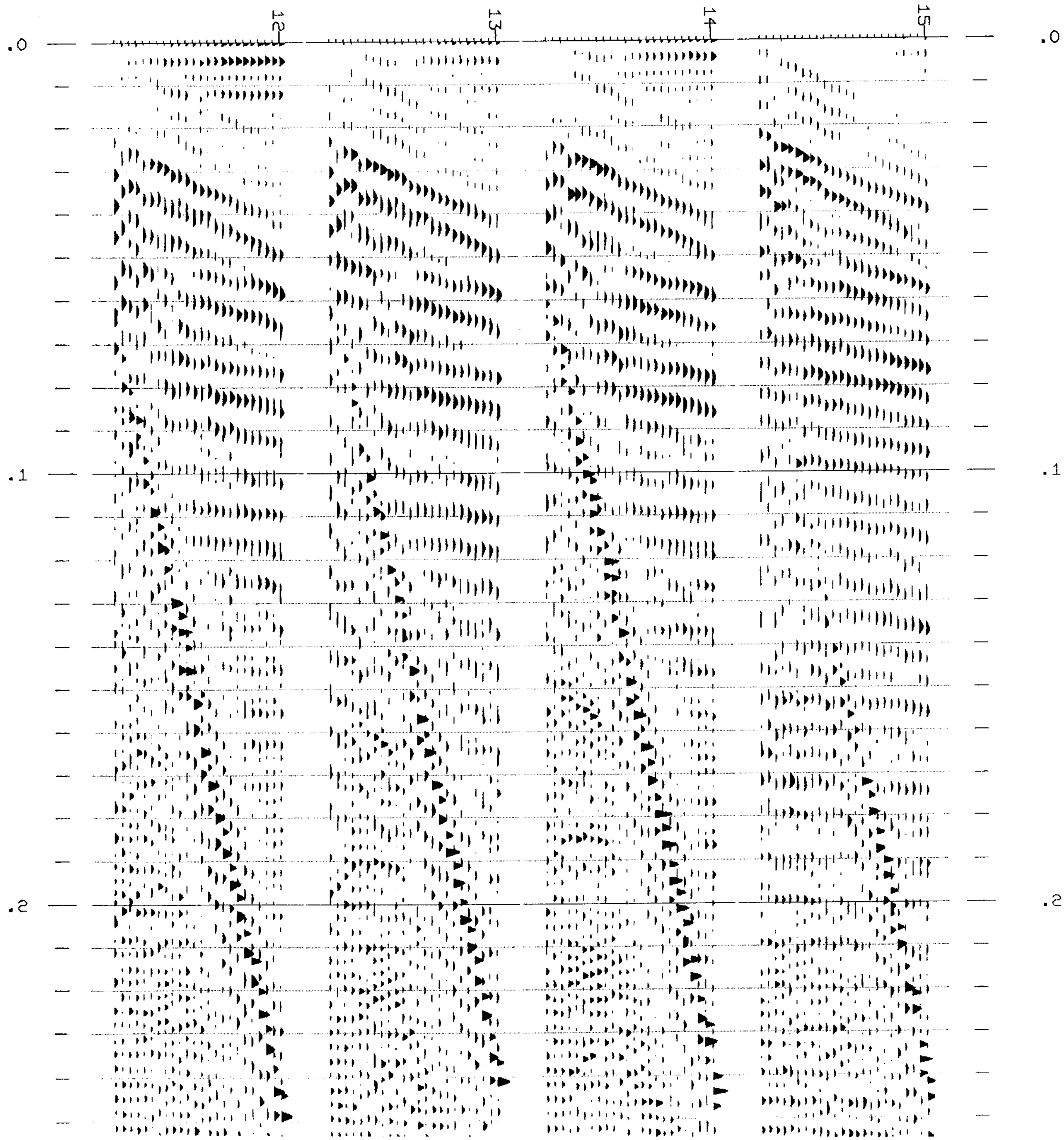

FIGURE 1. a. Input data for example run.

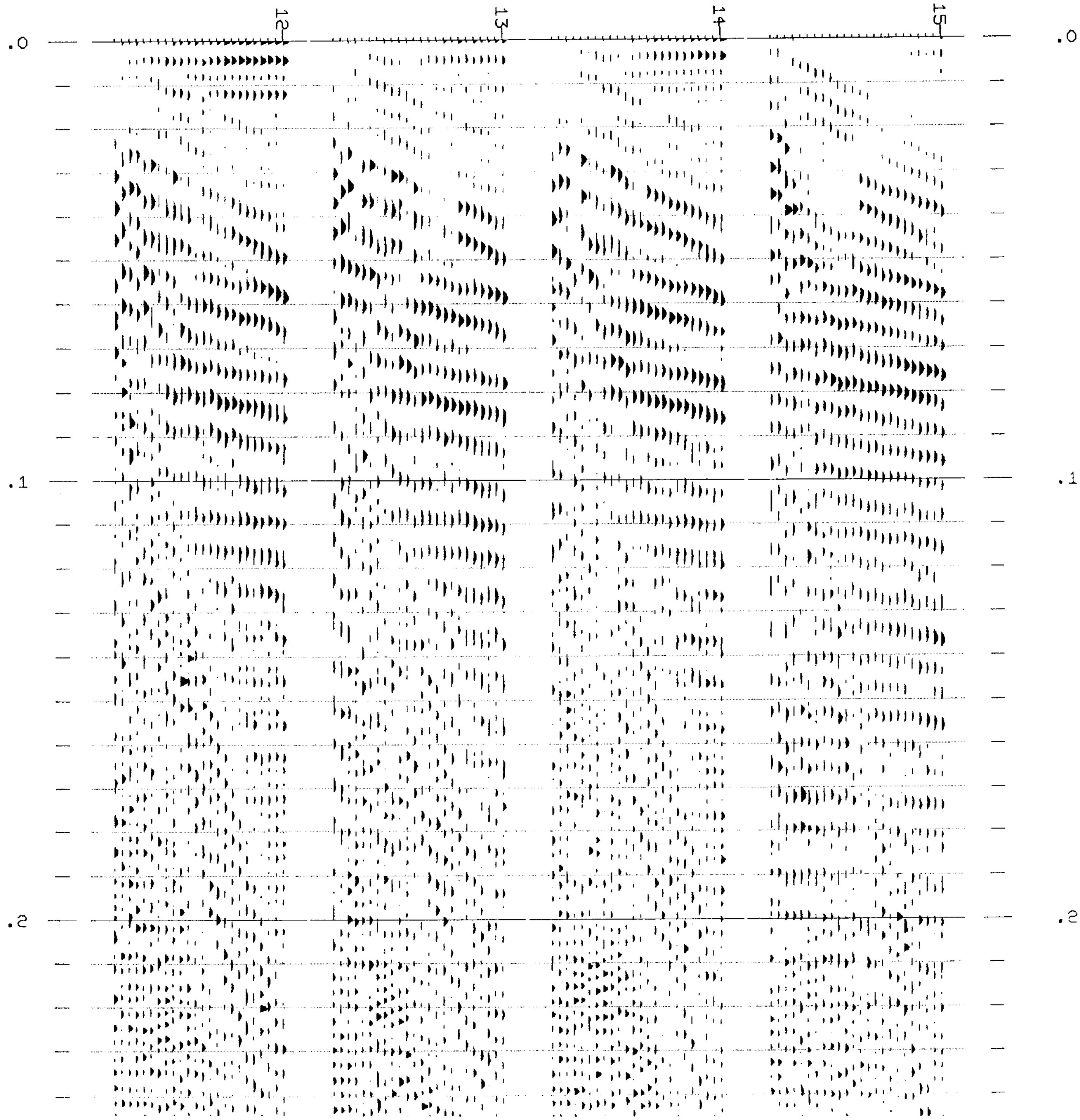

FIGURE 1. b. Output of SPEDIT.

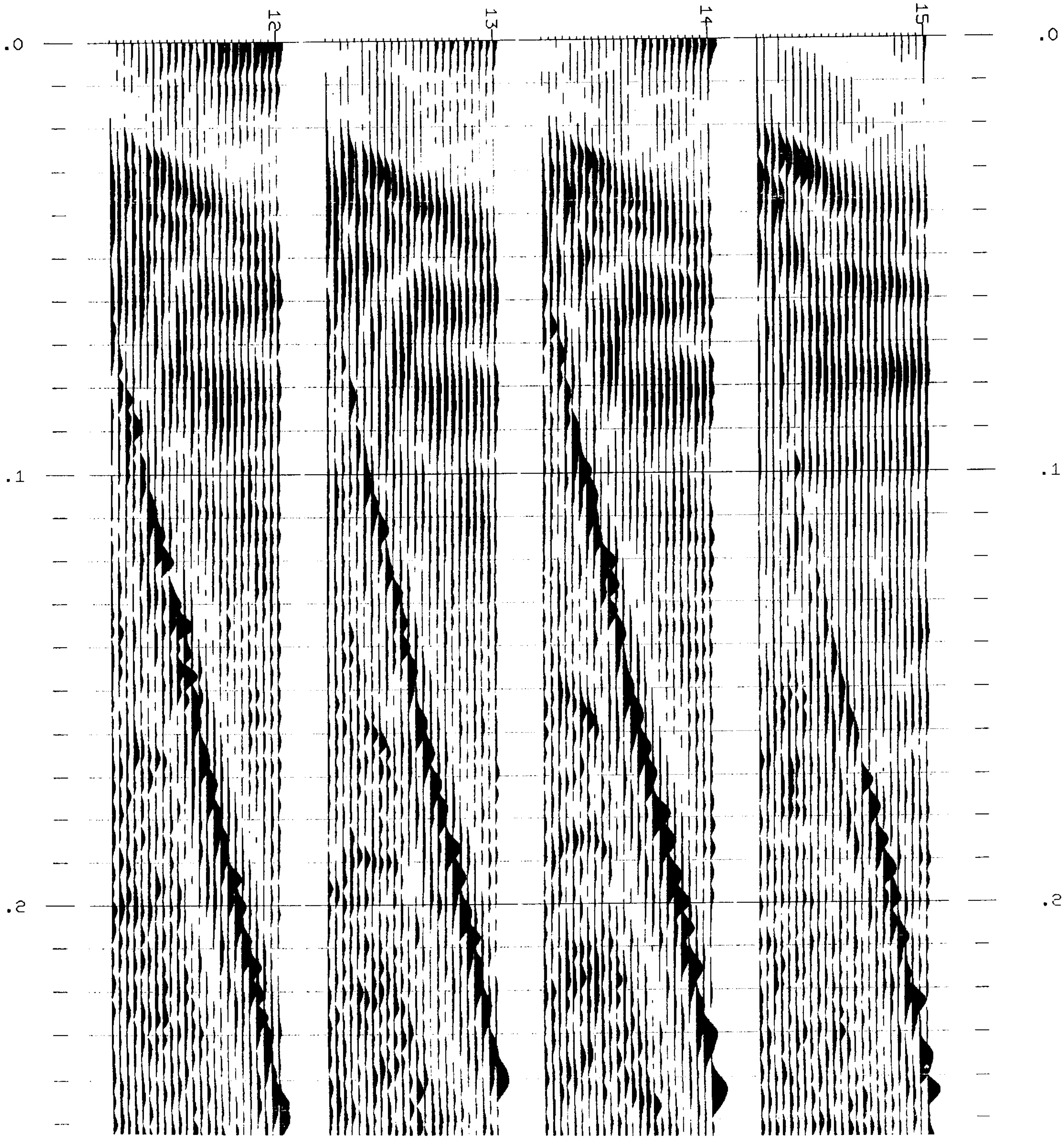

FIGURE 2. a. Input data of Figure 1a. Instantaneous amplitude.

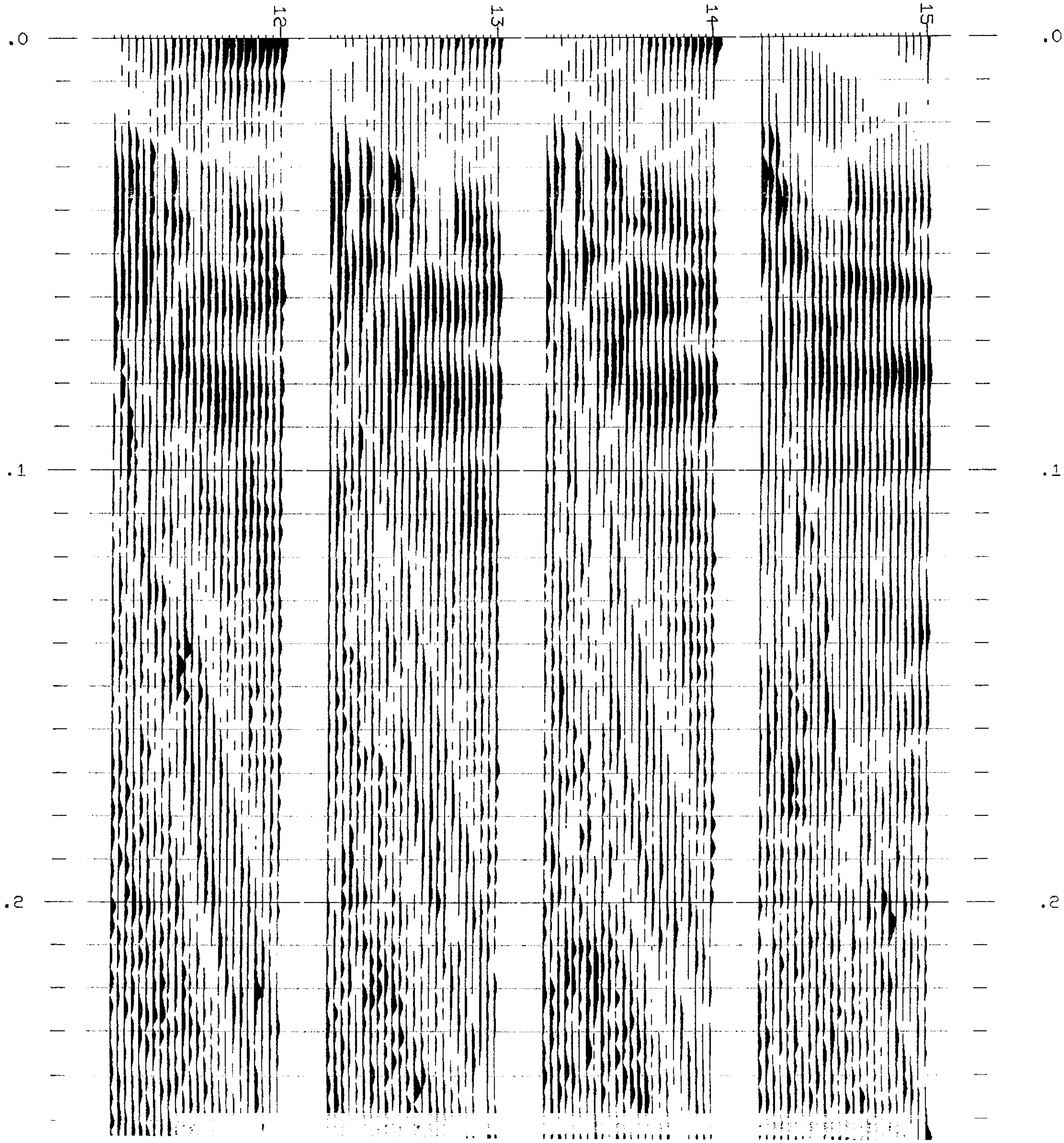

FIGURE 2. b. Output data of Figure 1b. Instantaneous amplitude.

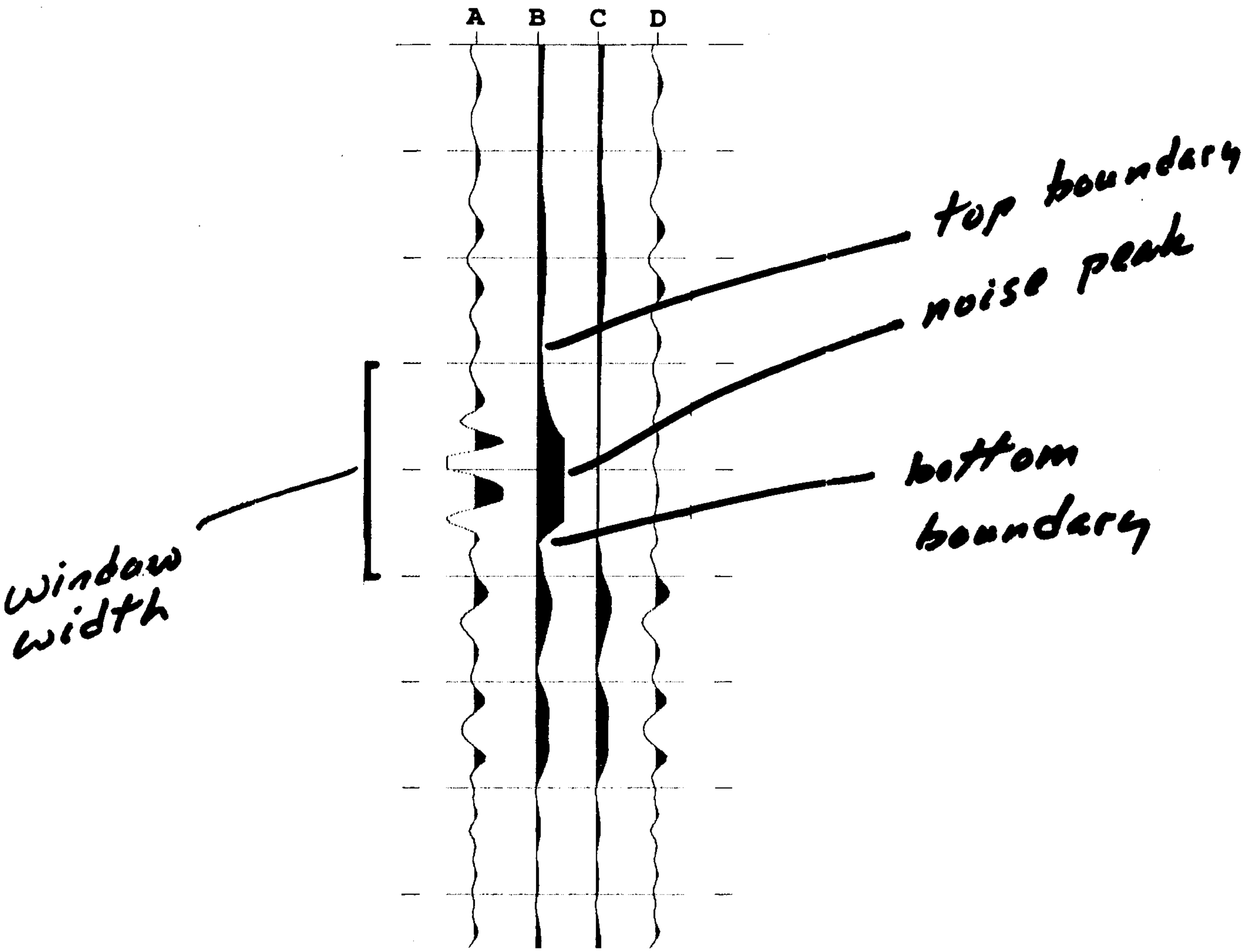

FIGURE **3. a.** Single trace of data.
b. Instantaneous amplitude, with noise peak window width and peak boundaries shown.
c. Instantaneous amplitude after application of SPEDIT.
d. Trace display after SPEDIT application.