Multiplets

by Jon F. Claerbout

On a seismic section a given multiple reflection arrival can split up into a fine structure of several arrivals. Such splitting occurs only when the dip is non-zero. Split up multiple reflections will hereafter be called multiplets. An example of a pegleg multiplet is shown in Figure 1.

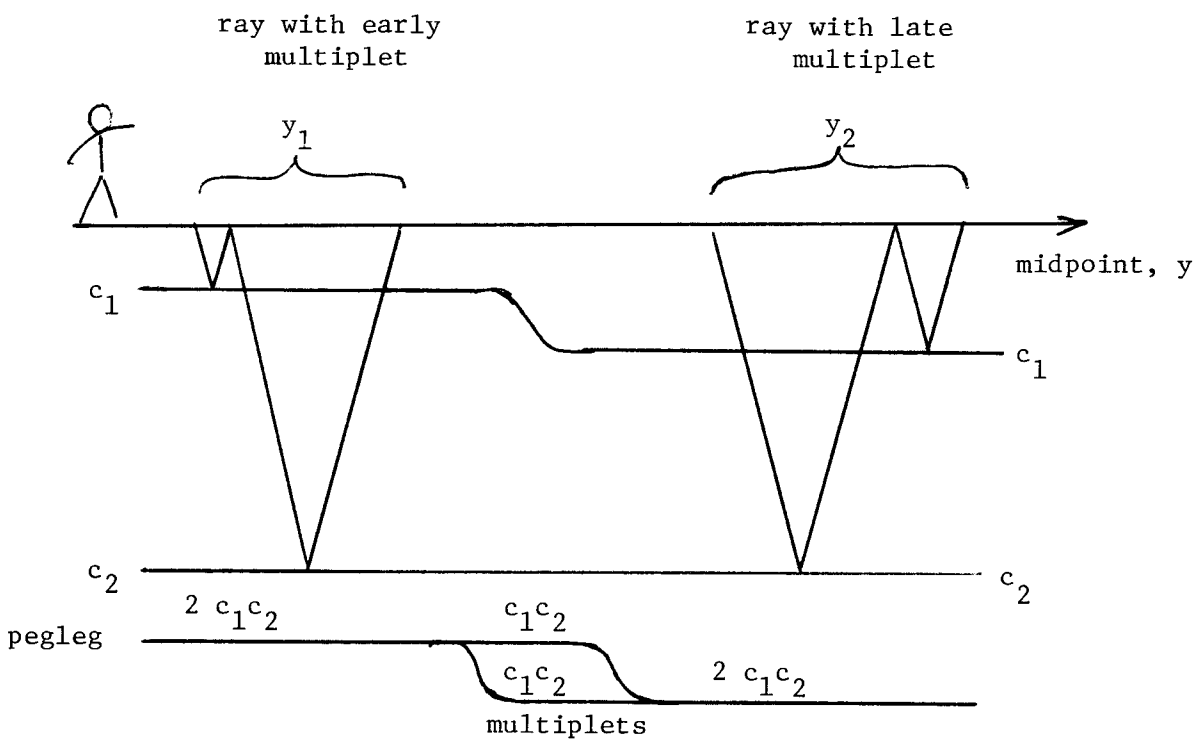

Figure 1. A geometry in which a pegleg multiple reflection splits into a doublet called a multiplet. (Non-flat regions are discussed later.)

A CDP stacked section provided by Western Geophysical Company is shown in Figure 2. It inspired a model for which vertical incidence synthetic data is shown in Figure 3. Figure 4 shows non-zero offset synthetic data. The offset chosen for Figure 4 actually increases linearly with time (to keep the angle of propagation nearly constant) in such a way as to reach half the cable length (half of 10,000 feet) by the bottom of the section (about 3.8 sec). The splitting of the first sea floor multiple on the slope is seen to give well over a half second gap. No splitting is visible on the stacked field data because the stacked data is a sum of many offsets so that the multiplets are too numerous and uniformly distributed to be visible. Consistent with this idea is the comparatively weak multiple on the early dipping part of the stacked section.

The multiplets of Figure 4 were generated with a simple algorithm. Figure 5 and equation (1) together illustrate the multiplet algorithm.

$$U_3^{\,y} = c_3^{\,y} D_0^{\,y} + \frac{c_2^{y+1} D_1^{y+1} + c_2^{y-1} D_1^{y-1} + c_1^{y+2} D_2^{y+2} + c_1^{y-2} D_2^{y-2}}{2} \qquad (1)$$

In equation (1) the variable $U_t^{\,y}$ is the upcoming wave associated with midpoint y at time t. Likewise $D_t^{\,y}$ is the downgoing wave and $c_t^{\,y}$ is the reflection coefficient as a function of travel time depth. By associating a wave with a midpoint we envision that half of the energy moves from left hand shot points to right hand receiver points and half goes the other way. We now introduce the initial condition that the downgoing wave $D_t^{\,y} = \delta_t$ is an impulse at zero time and that the upcoming wave at the surface reflects with sign reversal

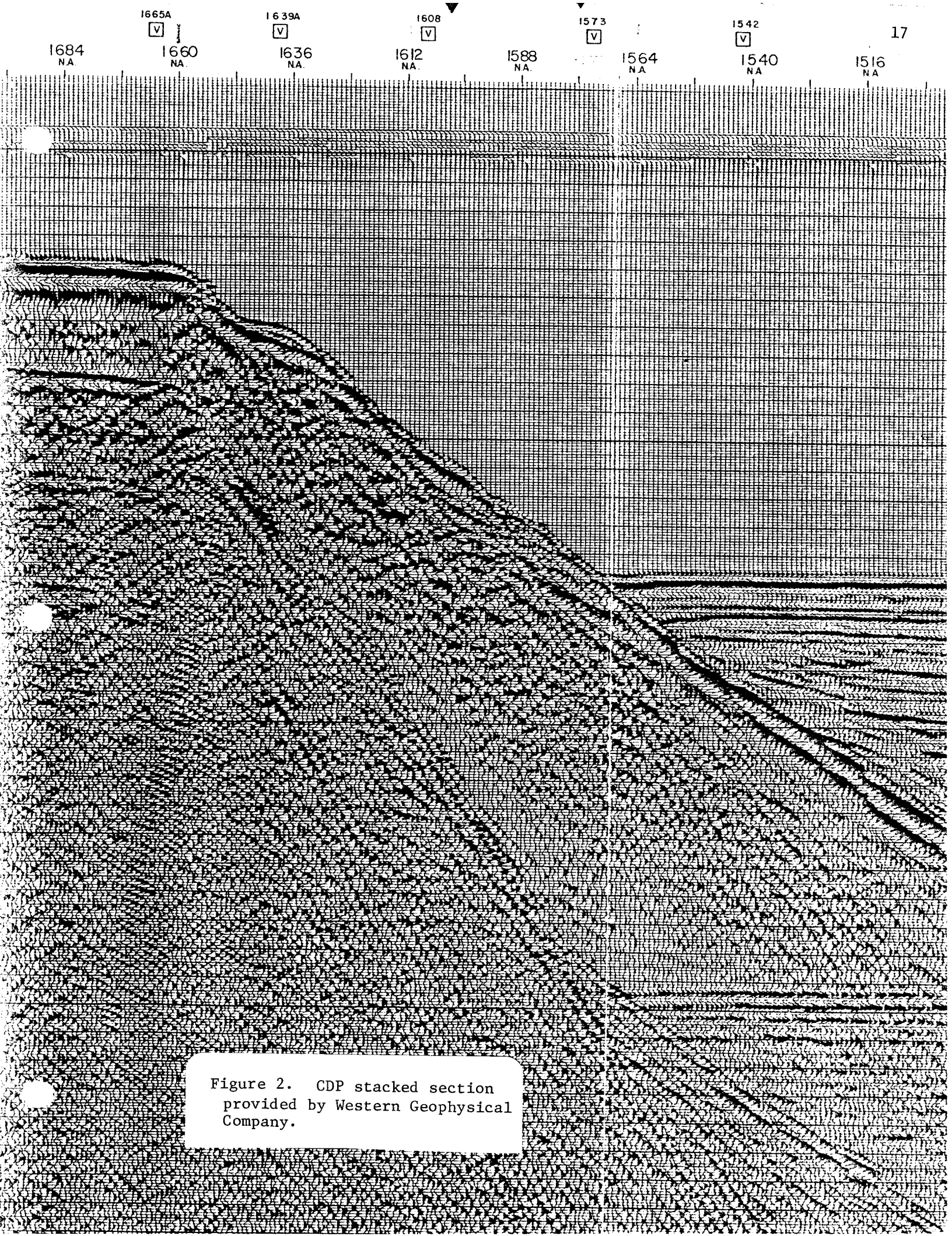

Figure 2. CDP stacked section provided by Western Geophysical Company.

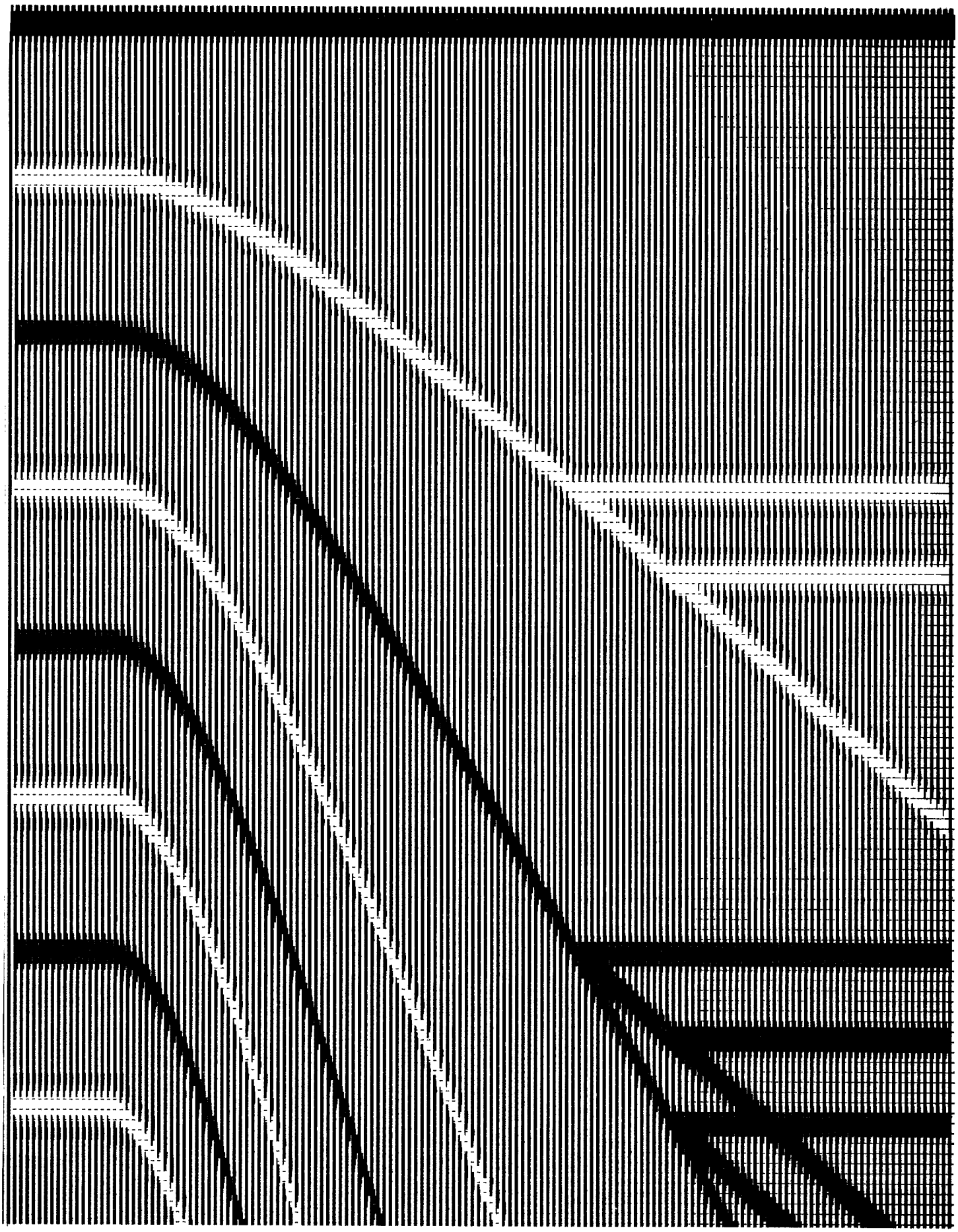

Figure 3. Synthetic seismograms for vertical incidence multiple reflections.

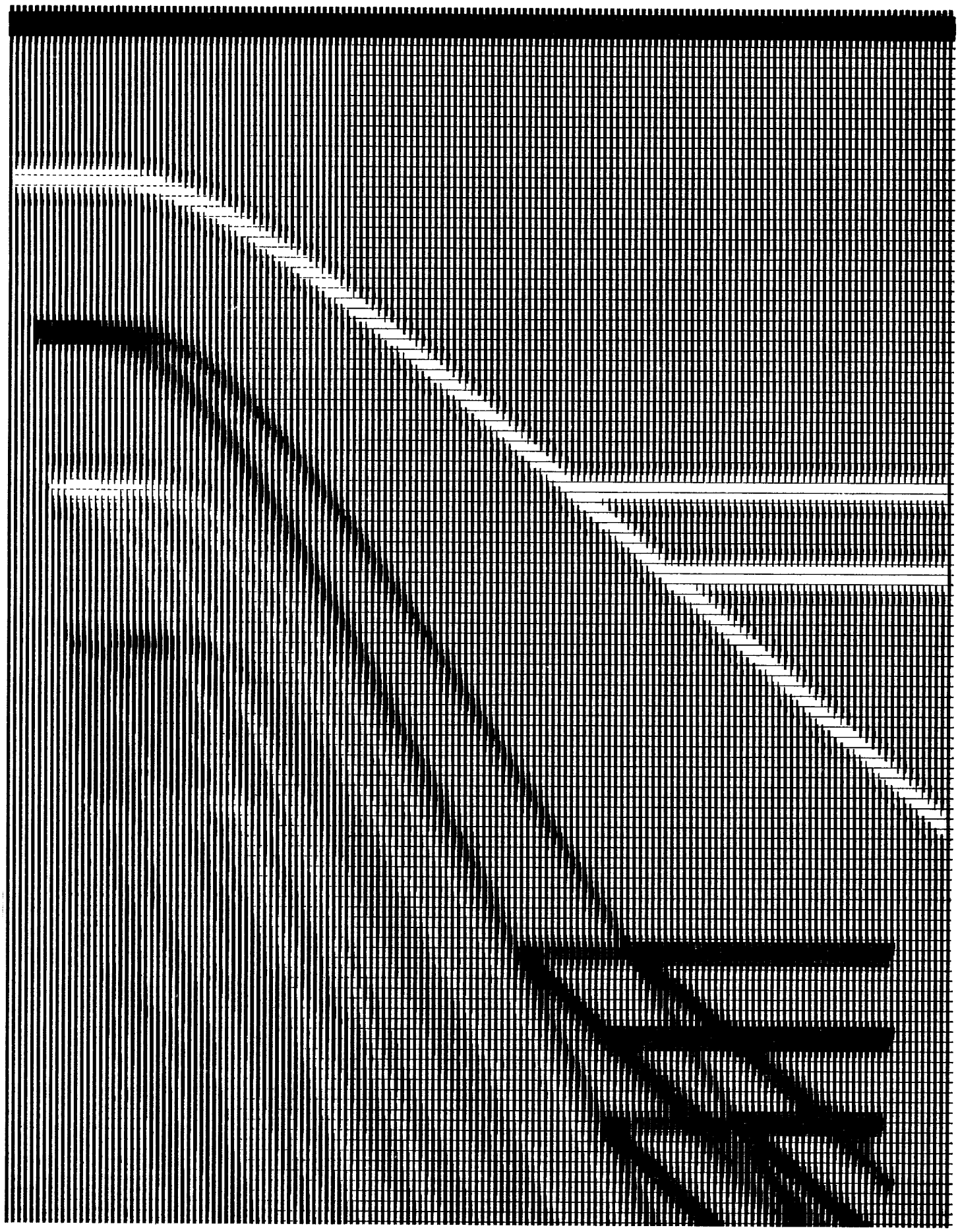

Figure 4. Multiples on dipping section split into multiplets. (See also cover for higher gain.)

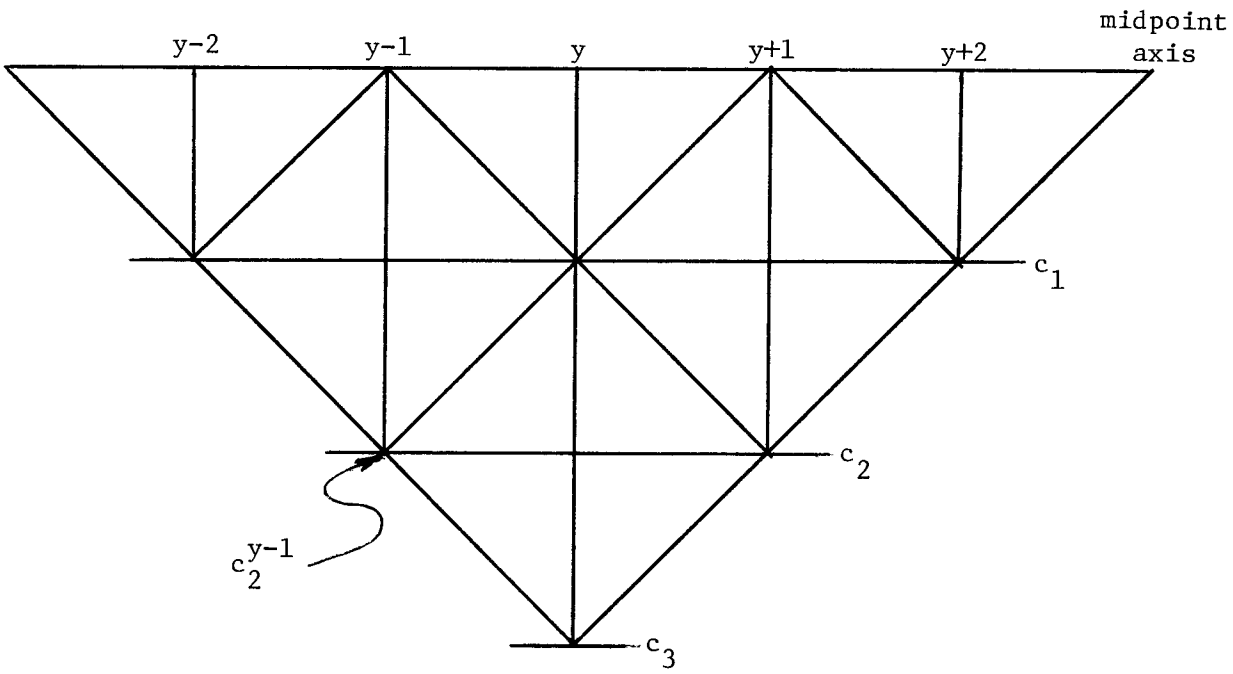

Figure 5. Basic geometry for the multiplet algorithm.

to make further downgoing waves, $D_t^y = -U_t^y$ for $t > 0$. Then (1) takes the form

$$D_t^y = -c_t^y - \frac{1}{2}\sum_{\tau=1}^{t-1} c_{t-\tau}^{y+\tau} D_\tau^{y+\tau} + c_{t-\tau}^{y-\tau} D_\tau^{y-\tau} \qquad (2)$$

Equation (2) may be solved recursively for D_t^y given c_t^y. Surprisingly the inverse problem solving for c_t^y given D_t^y is identical to the forward problem!

Figure 6 contains a checkout program and a subroutine (NOACMP, NOAh Common Mid Point) for the forward or inverse multiplet algorithm. Figure 7 is a printout of the input model (which resembles Figure 1). The input model printout was identical to the reconstructed model printout. The multiplet arising from this model is shown in Figure 8.

```
      C        TEST MULTIPLET ALGORITHM
            DIMENSION C(52,10),D(52,10)
            NY=52
            NT=10
            DO 10 IY=1,NY
            DO 10 IT=1,NT
     10     C(IY,IT)=0.
            DO 20 IY=1,26
     20     C(IY,3)=.05
            DO 25 IY=26,NY
     25     C(IY,5)=.05
            DO 30 IY=1,NY
     30     C(IY,6)=.8
            DO 40 IY=1,NY
     40     D(IY,1)=1.
            CALL OUT(NY,NT,C)
            CALL NOACMP(NY,NT,C,D)
            CALL OUT(NY,NT,D)
            CALL NOACMP(NY,NT,D,C)
            CALL OUT(NY,NT,C)
            STOP
            END

            SUBROUTINE NOACMP(NY,NT,C,D)
            DIMENSION C(NY,NT),D(NY,NT)
            DO 10 IT=2,NT
            DO 10 IY=1,NY
     10     D(IY,IT)=-C(IY,IT)
            DO 20 IT=3,NT
            NZ=IT-1
            DO 20 IZ=2,NZ
            LAT=(IT-IZ)
            IY1=LAT+1
            IY2=NY-LAT
            ITZ=IT-IZ+1
            DO 20 IY=IY1,IY2
            IP=IY+LAT
            IM=IY-LAT
            TERM=.5*(C(IP,IZ)*D(IP,ITZ)+C(IM,IZ)*D(IM,ITZ))
     20     D(IY,IT)=D(IY,IT)-TERM
            RETURN
            END

            SUBROUTINE OUT(NY,NT,C)
            DIMENSION  C(NY,NT),LINE(100)
            PRINT 10
     10     FORMAT('1 NEXT SECTION')
            DO 30 IY=1,NY
            DO 20 IT=1,NT
     20     LINE(IT)=100.5*C(IY,IT)
     30     PRINT 40,IY,(LINE(IT),IT=1,NT)
     40     FORMAT( 8X,12I6)
            RETURN
            END
```

Figure 6. Subroutine and checkout program for multiplet generation and inversion.

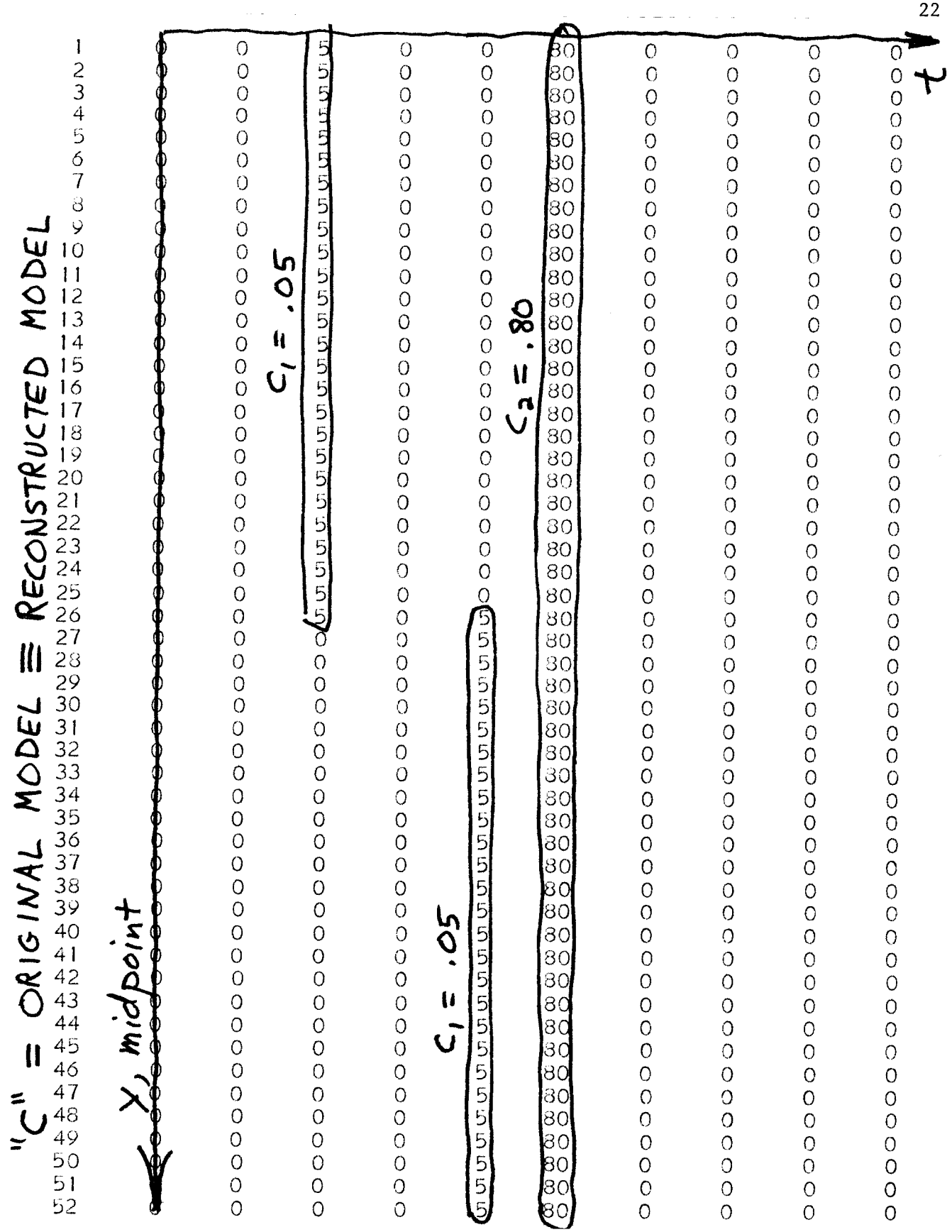

Figure 7. Model and reconstructed model for checkout of NOACMP multiplet algorithm.

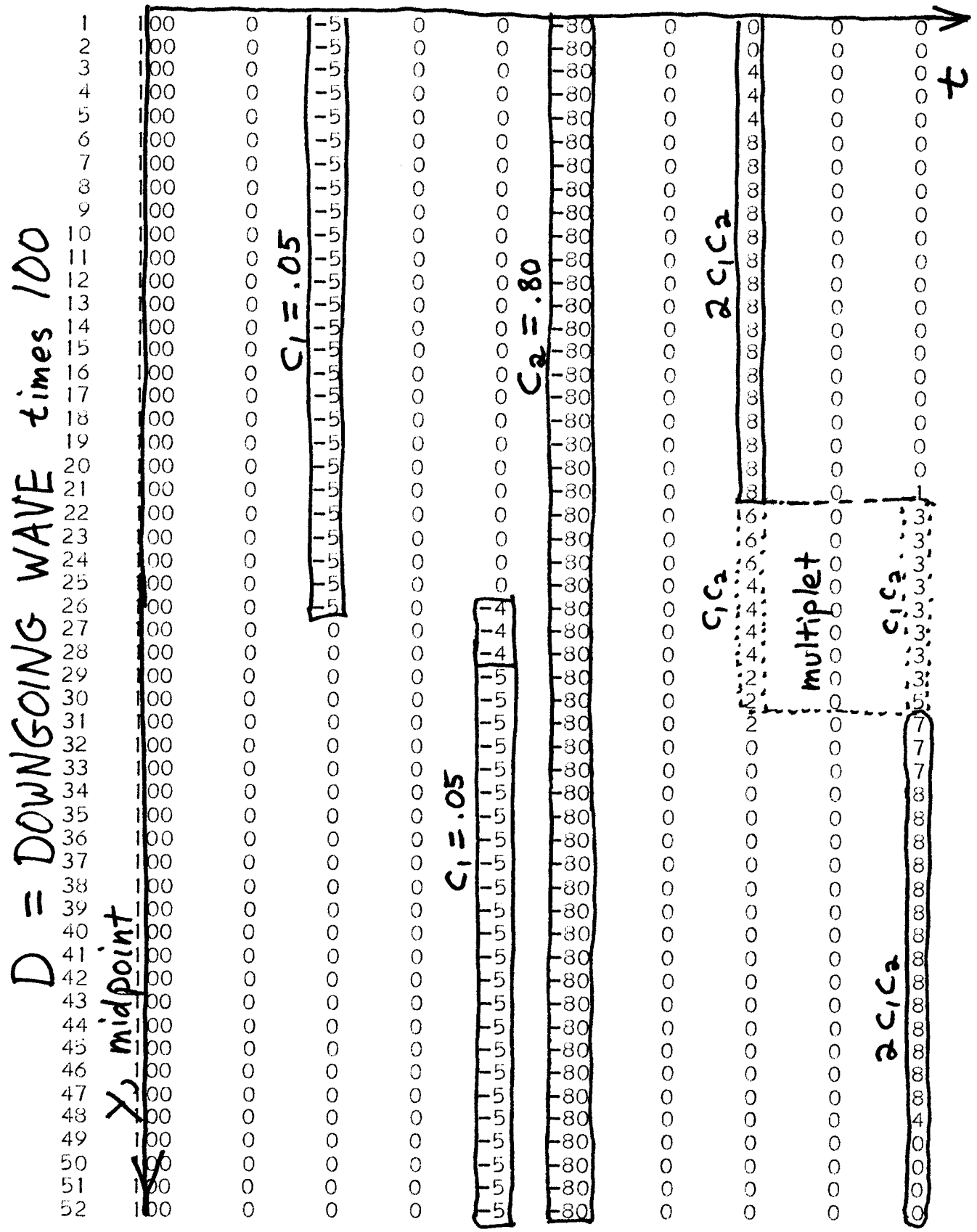

Figure 8. Pegleg multiplet arising in checkout program.

Our most ambitious goal in this research is to accurately predict and remove the multiples and multiplets on the various slant stacks before they are combined in a final stack. This will certainly require a good shot waveform estimation procedure.

A more modest goal is to use the multiplet algorithm (and possibly migration) to generate only the synthetic multiples which arise from strong events. These synthetic multiples and multiplets could then be rectified and smoothed to provide inverse weighting functions to be used before conventional CDP stack.

Notice in Figure 4 the large size of the window between the two multiplets of the sea floor first multiple. Note also on Figure 4 the triangular region of missing data along the left margin. This triangular region is indicative of the offset of this synthetic data. In this case the offset is only about 23 traces or half the cable length at the bottom of the section. Clearly, the gap within multiplets can be even larger at larger offsets, and there should be some big windows to look through at the primaries.

The multiplet algorithm would seem to be strictly valid only for layered earth models where reflection coefficient amplitudes vary slowly along the reflectors compared to a wavelength. But a mathematical limit of such models will produce dipping beds. Bedding dip must also give rise to diffraction and migration phenomena not included in the multiplet algorithm. Presumably these can be handled in the fashion of SEP-3, Don C. Riley's PhD thesis. The multiplet algorithm was developed to broaden the applicability of Riley's

work to wide offset sections. Doherty's work on large offset, SEP-4, covered wide offset on upcoming waves but did not couple the up and downgoing wide offset waves. Earlier work in this report and SEP-5 looks less symmetrical than the multiplet algorithm. However, studies beginning on page 36 seem to indicate that the multiplet algorithm probably applies to common geophone (or shot) slant stacks and not to common midpoint slant stacks as I first believed. I had hoped to derive an equation governing common midpoint slant stacks which within various scale factors might look like

$$\partial_z U(y,t) = \partial_{yy}^t U(y,t) + c(y+z,z)\ D(y+z,t-2z) + c(y-z,z)\ D(y-z,t-2z)$$

Analogous equations are found in SEP-2, p. 307; also in this report "Coupled Slanted Beams, Equations for Multiples Program", on page 35.

We found we had some difficulty in understanding why the sea floor multiple in Figure 4 split into two. The explanation for this shows why we believe these algorithms apply to slant stacks instead of radial traces. In studying the example of Figure 8.5, keep in mind that a "radial trace" is a bit like a trace from a constant offset section from conventional point source data, whereas a slant stack represents a plane wave source. It seems to require less inhomogeneity to produce multiple arrivals for a plane wave source than a point source. I suppose that it is often a reasonable practical expedient to utilize the multiplet algorithm equation (2) without diffraction. Justification might be that offset angle is most commonly larger than the earth dip angle.

Finally, I would like to append the program which generated Figures 3 and 4 and add a few practical details. It was important that the lateral shift should be less than one midpoint per time point.

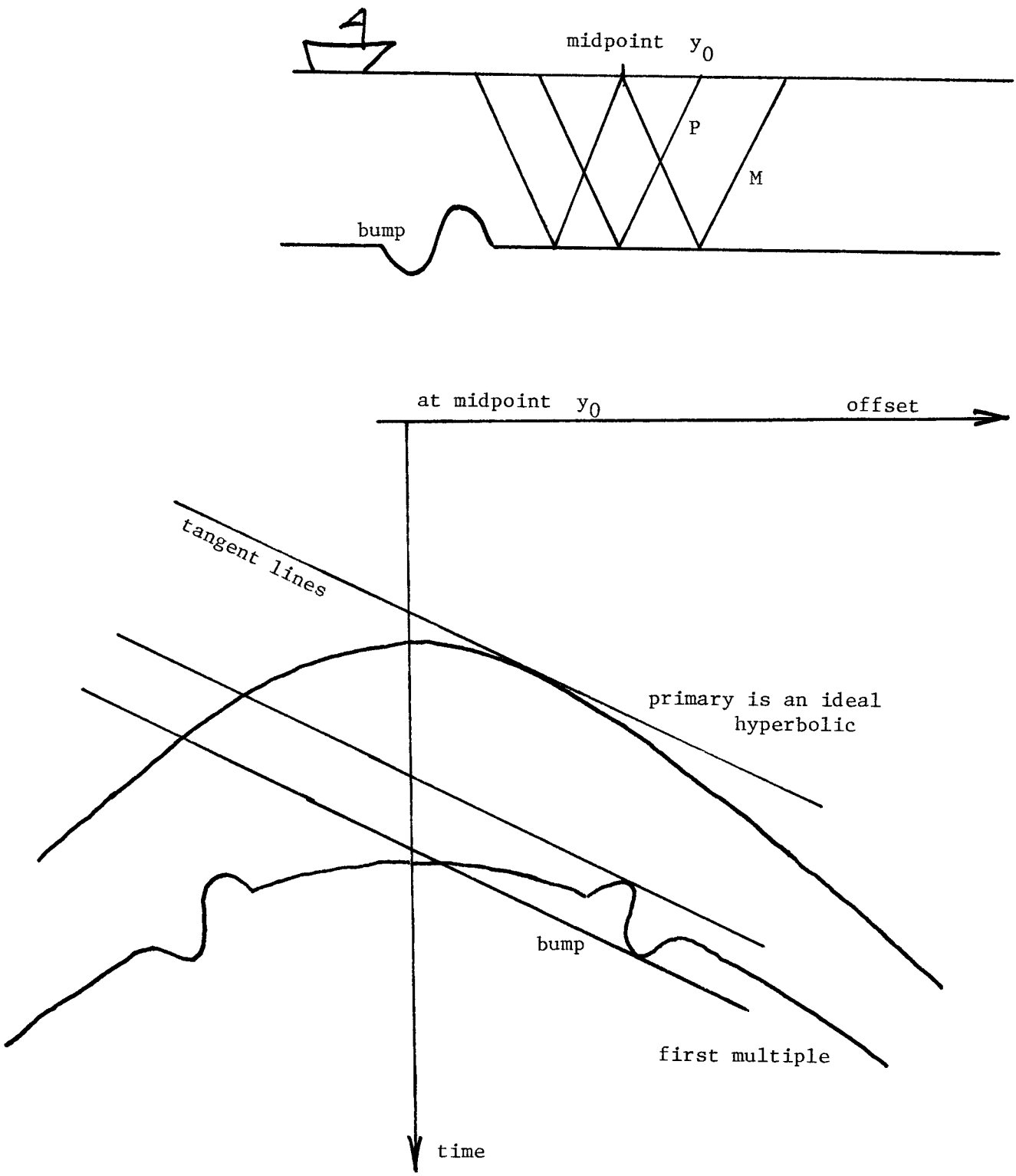

Fig. 8.5. The number of arrivals on a slant stack may easily exceed the number of arrivals from point sources.

In the program of Figure 9 such reduced lateral shifting was made possible by the TANDD parameter which was zero for Figure 3 and about .1 for Figure 4. A few preliminary runs showed that linear lateral interpolation was a worthwhile improvement over closest neighbor interpolation. In reorganizing the program to take arbitrary initial downgoing waveforms the ability to do the inverse algorithm with the same program as the forward algorithm was lost. Luckily the compiler used considers the product of two half word integers to be a full word integer; otherwise fixed point overflow would have occurred in the product of the downgoing wave D times the reflection coefficient C . It would be safer to do the program in floating point. Half word integer calculations were done to save memory. The use of integer calculations is justified by some exponential scaling transformations. The model generation program MODELW in Figure 10 has the task of laying down on a mesh a delta function at an arbitrary time point. Subroutine ADDELT does this by means of the Schultz function, SEP-5, p. 79.

```
10.             INTEGER*2 D(151,201),C(151,201)
11.             VER=1.
12.             WRT=1.
13.             NY=151
14.             NT=201
15.             NSOUR=3+3*VER
16.             TANDD=.5*46.*(NY/150.)/NT
17.             CALL MODELW(NY,NT,C)
19.             CALL DOWN(NY,NT,D,NSOUR)
20.             CALL NOACMP(NY,NT,D,C,   0.)
21.             CALL OUTPUT(NY,NT,D,VER,WRT)
22.             CALL DOWN(NY,NT,D,NSOUR)
23.             CALL NOACMP(NY,NT,D,C,TANDD)
24.             CALL OUTPUT(NY,NT,D,VER,WRT)
27.             STOP
28.             END

52.             SUBROUTINE NOACMP(NY,NT,D,C,TANDD)
53.             INTEGER*2 D(NY,NT),C(NY,NT),ITERM
54.             DO 10 IT=2,NT
55.             DO 10 KT=2,IT
56.             IZ=IT-KT+2
57.             SHIFT=(IT-IZ)*TANDD
58.             LAT=SHIFT
59.             LATP=LAT+1
60.             IY1=LATP+1
61.             IY2=NY-LATP
62.             ITZ=IT-IZ+1
63.             A=(1.-SHIFT+LAT)/2048.
64.             B=(   SHIFT-LAT)/2048.
65.             DO 10 IY=IY1,IY2
66.             IP=IY+LAT
67.             IM=IY-LAT
68.             ITERM=A*(C(IP  ,IZ)*D(IP  ,ITZ)+C(IM  ,IZ)*D(IM  ,ITZ))
69.            1     +B*(C(IP+1,IZ)*D(IP+1,ITZ)+C(IM-1,IZ)*D(IM-1,ITZ))
70.             D(IY,IT)=D(IY,IT)-ITERM
71.       10    CONTINUE
72.             RETURN
73.             END

74.             SUBROUTINE DOWN(NY,NT,D,NSOUR)
75.             INTEGER*2 D(NY,NT)
76.             DIMENSION PASCAL(11)
77.             PASCAL(1)=1.
78.             PASCAL(2)=PASCAL(1)
79.             DO 10 K=3,NSOUR
80.             PASCAL(K)=PASCAL(K-1)
81.             KM=K-1
82.             DO 10 I=2,KM
83.       10    PASCAL(K-I+1)=PASCAL(K-I+1)+PASCAL(K-I)
84.             SCALE=1.5*1024./PASCAL(NSOUR/2)
85.             DO 30 IY=1,NY
86.             DO 20 IT=1,NT
87.       20    D(IY,IT)=0
88.             DO 30 IT=1,NSOUR
89.       30    D(IY,IT)=PASCAL(IT)*SCALE
90.             RETURN
91.             END
```

Figure 9. Test program, laterally interpolating multiplet program, and downgoing wave initialization program (binomial coefficient wavelet).

```
      SUBROUTINE MODELW(NY,NT,C)
      INTEGER*2 C(NY,NT)
      DATA C1,C2,C3,C4,C5/1.,1.,1.,1.,1./
      DATA D1,D2,D3/.095,.38,.45/
      DATA Y1,Y2,Y4/.1,.6,1./
      DO 05 IY=1,NY
      DO 05 IT=1,NT
05    C(IY,IT)=0.
      A=(D2*D2-D1*D1)/(Y2-Y1)**2
      IY1=1+(NY-.5)*Y1
      IY2=1+(NY-.5)*Y2
      IY3=IY1+(NY-.5)*SQRT((D3*D3-D1*D1)/A)
      D=D1*(NT-1)+1.5
      IY1M=IY1-1
      DO 15 IY=1,IY1M
15    CALL ADDELT(IY,NY,NT,C,D,C1*1024)
      DO 10 IY=IY1,NY
      D=SQRT(D1*D1+A*((IY-IY1)/(NY-1.))**2)*(NT-1)+1.5
      CC=C1
      IF(IY.GE.IY2) CC=C4
      IF(IY.GE.IY3) CC=C5
10    CALL ADDELT(IY,NY,NT,C,D,CC*1024)
      D=D2*(NT-1)+1.5
      IY2P=IY2+1
      DO 20 IY=IY2P,NY
20    CALL ADDELT(IY,NY,NT,C,D,C2 *1024)
      D=D3*(NT-1)+1.5
      IY3P=IY3+1
      DO 30 IY=IY3P,NY
30    CALL ADDELT(IY,NY,NT,C,D,C3*1024)
      RETURN
      END
      SUBROUTINE ADDELT(IY,NY,NT,Q,TO,SCALE)
      INTEGER*2 Q(NY,NT)
      C568=.568*SCALE
      C0688=.0688*SCALE
      ITO=TO
      IF(ITO.LT.2.OR.ITO.GT.NT-2)RETURN
      DELT=TO-ITO
      DT=.5-DELT
      SL1=(SCALE-C568)/.5
      SL2=C568/.5
      SL3=C0688/.5
      IF(DT) 20,20,40
  20  DELT=1.-DELT
      Q(IY,ITO-1)=Q(IY,ITO-1)-DELT*SL3
      Q(IY,ITO)=Q(IY,ITO)+DELT*SL2
      Q(IY,ITO+2)=Q(IY,ITO+2)-DELT*SL3
      Q(IY,ITO+1)=Q(IY,ITO+1)-DT*SL1+C568
      GO TO 100
  40  Q(IY,ITO-1)=Q(IY,ITO-1)-DELT*SL3
      Q(IY,ITO)=Q(IY,ITO)+DT*SL1+C568
      Q(IY,ITO+2)=Q(IY,ITO+2)-DELT*SL3
      Q(IY,ITO+1)=Q(IY,ITO+1)+DELT*SL2
 100  RETURN
      END
```

Figure 10. Subroutines for initializing the model of Figure 3 and 4.