MIGRATION EXAMPLES USING FOURIER TRANSFORMS

Giovanni Da Prat and Walt Lynn

Stolt's method of migration using Fourier transforms has been discussed by Claerbout (this report) and the practical implementation of the technique by Lynn (this report). In this paper, we will show a few examples of frequency domain, f-k , migration on both synthetic and real data.

The first example, shown in Fig. 1, is the migration of a synthetic seismic section consisting of only three non-zero arrivals. The grid dimensions are 64 traces in x with sample interval equal to 1 m and 128 points in time with sample interval equal to 1 sec. The wave field is given by

$$P(x,t,z=0) \quad = \quad 100 \cdot \delta(x-32)[\delta(t-32)+\delta(t-64)+\delta(t-96)].$$

The correct migrated section should consist of three semicircles. The partial semicircles which enter from the side of the grid are due to the periodic boundary conditions inherent in the f-k migration method due to the use of discrete Fourier transforms.

The inverted semicircle at the bottom, associated with the spike at $t=96$, is another matter. We believe it is caused by a violation of causality. A causal time function is represented in a discrete Fourier transform by making the latter half of the time function zero. The causality problem can be rectified by manipulating the phase of the DFT; however, it is easier just to append zeros to the time traces.

A second example is shown in Fig. 2(a). In this case, the grid dimensions are 64 traces in x with sample interval equal to 25 m and 512 points in time with sample interval of 4 msec. The synthetic seismogram is a hyperbola with a velocity of 2000 m/sec. The output for three different velocities, 1000, 2000, and 3000 m/sec, are shown in Fig. 2(b) and 3(a) and (b). Notice in all three cases the aliasing along the x-direction. This effect comes from a bigger sampling interval associated with x . This was not evident in the previous example where the sampling in x is 1 m and the sampling in t is 1 sec. As it is expected, the hyperbola collapses to a point for $v = 2000$ m/sec. Undermigration is observed for $v = 1000$ m/sec, and overmigration is noted for $v = 3000$ m/sec.

The migration of the same hyperbola with an explicit 15° wave equation algorithm is shown in Fig. 4. The result differs from the first method in the absence of periodicity effects.

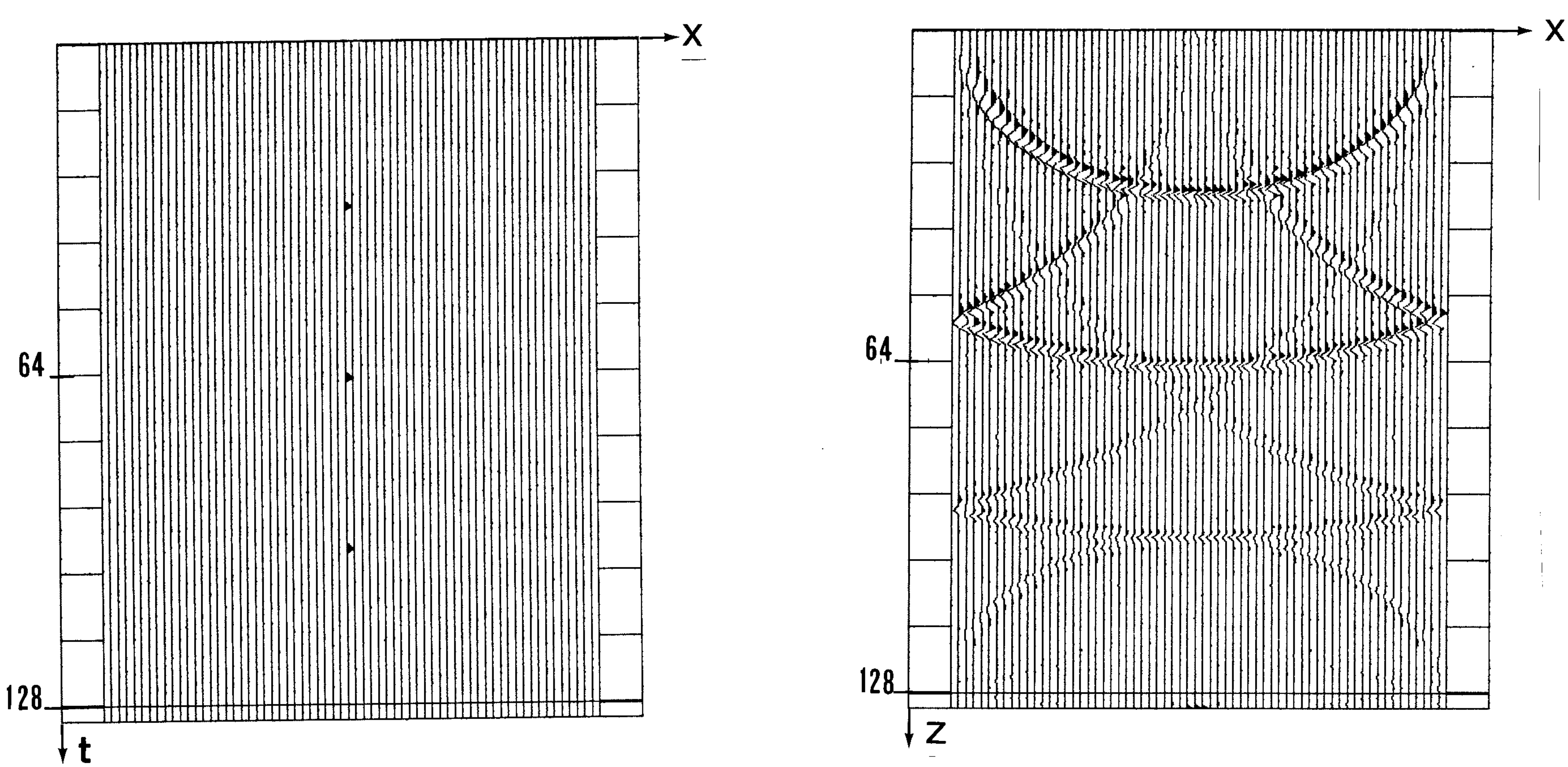
The final example is the application of the method to real data. The input is shown in Fig. 5. It consists of a gather from the Grand Banks off Newfoundland. The field characteristics are: 50 m for group interval, 4-msec sampling rate, and a near trace offset of 311 m. To get a power of two, as required for the FFT's algorithm, the number of traces in the original gather was increased from 48 to 64 each of 1024 time points by appending zero traces to both sides of the grid. In addition, a few traces were set to zero since they contained wrong seismic information.

The results for three different velocities, 1000, 2000, and 3000 m/sec, are shown in Figs. 6, 7, and 8. Notice the similarity of the results for each migrated hyperbola as compared with the synthetic case.

The computer time needed on our PDP-11/34 to migrate the gather for a given velocity was 2 hr .8 min. Out of this, 1 hr was required for the FFTs alone. The reason is that at the time the program was executed there was no floating-point hardware. The program that does the interpolation is presented here. The time of execution for this program was 33 min.

```
c
      INTERPOLATION  PROGRAM
c
      complex cm(1025),co(1025),clog,cexp,cabs
      data nx/64/,nt/1024/,st/0.004/,sx/50./
      call setfil(10,'/mnt/daprat/inter',512)
      call setfil(11,'/mnt/daprat/intermult',512)
      pi=3.141592
      v=500.
      delx=(2.*pi)/(nx*sx)
      delz=(2.*pi)/(nt*st*v)
      delw=delz*v
      nyx=nx/2+1
      nyt=nt/2+1
      nyx1=nyx+1
      wny=pi/st
c
c     do loop from kx=0. to kx=knyquist.
c
      do 30 j=1,nyx
      x=delx*(j-1)
      read(10),(cm(i),i=1,nt)
      cm(nt+1)=cm(1)
      do 20 i=1,nyt
      z=delz*(i-1)
      it=nt-i+2
      if(z.eq.0.)goto 5
      w=v*z*sqrt(1.+(x/z)**2)
      go to 6
 5    w=v*x
 6    if(w.gt.wny)goto 1000
      al1=w/delw+1.
      l1=int(al1)
      q=al1-float(l1)
      l2=nt-l1+2
      if(j.eq.1.and.i.eq.1)goto 7
      s=(-v*z)/sqrt(x**2+z**2)
      go to 8
 7    s=-v
c8    co(it)=(1.-q)*cm(l1)+q*cm(l1+1)
c     write(6,987),cm(l1),cm(l1+1)
c987  format(e20.6,e20.6)
 8    if(cm(l1).eq.(0.,0.).or.cm(l1+1).eq.(0.,0.))goto 5000
      co(it)=cm(l1)*cexp(q*clog(cm(l1+1)/cm(l1)))
      go to 6000
 5000 co(it)=cmplx(0.,0.)
 6000 co(it)=co(it)*s
c     co(i)=(1.-q)*cm(l2)+q*cm(l2-1)
      if(cm(l2).eq.(0.,0.).or.cm(l2-1).eq.(0.,0.))goto 6001
      co(i)=cm(l2)*cexp(q*clog(cm(l2-1)/cm(l2)))
      go to 6002
 6001 co(i)=cmplx(0.,0.)
 6002 co(i)=co(i)*s
      go to 20
 1000 co(i)=0.
      co(it)=0.
```

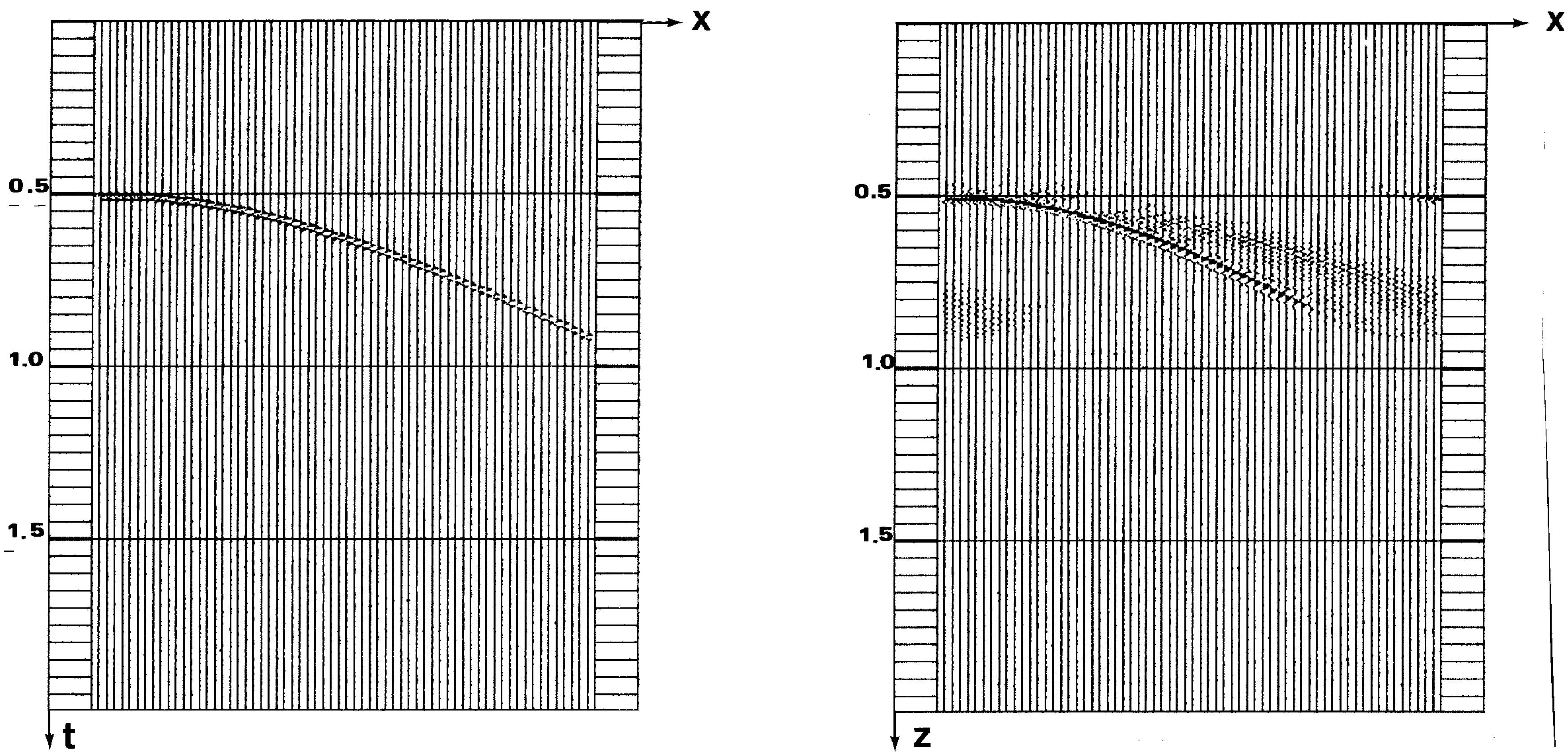
```
 20     continue
        write(11),(co(i),i=1,nt)
 30     continue
c       do 31 j=1,nyx
c       write(6,401),(co(i),i=1,nt)
c 401   format(8f6.2,8f6.2)
c 31    continue
c
c       do loop from kx=-knyquist to kx=0.
c
        do 40 j=nyx1,nx
        kx=nx-j+1
        read(10),(cm(i),i=1,nt)
        cm(nt+1)=cm(1)
        do 51 i=1,nyt
        x=-kx*delx
        z=delz*(i-1)
        it=nt-i+2
        if(z.eq.0.)goto 50
        w=v*z*sqrt(1.+(x/z)**2)
        go to 60
 50     w=v*sqrt(x**2)
 60     if(w.gt.wny)goto 999
        al1=w/delw+1.
        l1=int(al1)
        q=al1-float(l1)
        l2=nt-l1+2
        if(j.eq.1.and.i.eq.1)goto 70
        s=(-v*z)/sqrt(x**2+z**2)
        go to 80
 70     s=-v
c80     co(it)=(1.-q)*cm(l1)+q*cm(l1+1)
 80     if(cm(l1).eq.(0.,0.).or.cm(l1+1).eq.(0.,0.))goto 8000
        co(it)=cm(l1)*cexp(q*clog(cm(l1+1)/cm(l1)))
        go to 8002
 8000   co(it)=cmplx(0.,0.)
 8002   co(it)=co(it)*s
c       co(i)=(1.-q)*cm(l2)+q*cm(l2-1)
        if(cm(l2).eq.(0.,0.).or.cm(l2-1).eq.(0.,0.))goto 9000
        co(i)=cm(l2)*cexp(q*clog(cm(l2-1)/cm(l2)))
        go to 9001
 9000   co(i)=cmplx(0.,0.)
 9001   co(i)=co(i)*s
        go to 51
 999    co(i)=0.
        co(it)=0.
 51     continue
        write(11),(co(i),i=1,nt)
 40     continue
c       do 52  j=nyx1,nx
c       write(6,402),(co(i),i=1,nt)
c 402   format(8f6.2,8f6.2)
c 52    continue
        stop
        end
```



(a) (b)

FIGURE 1.—(a) Input seismic section. The triangular shape of the input spikes is due to interpolation in the plotting. The arrivals are at t = (1/4, 1/2, and 3/4) t_{max}, where $t_{max} = 128$. The sample in X is 1, and the sample in t is 1.

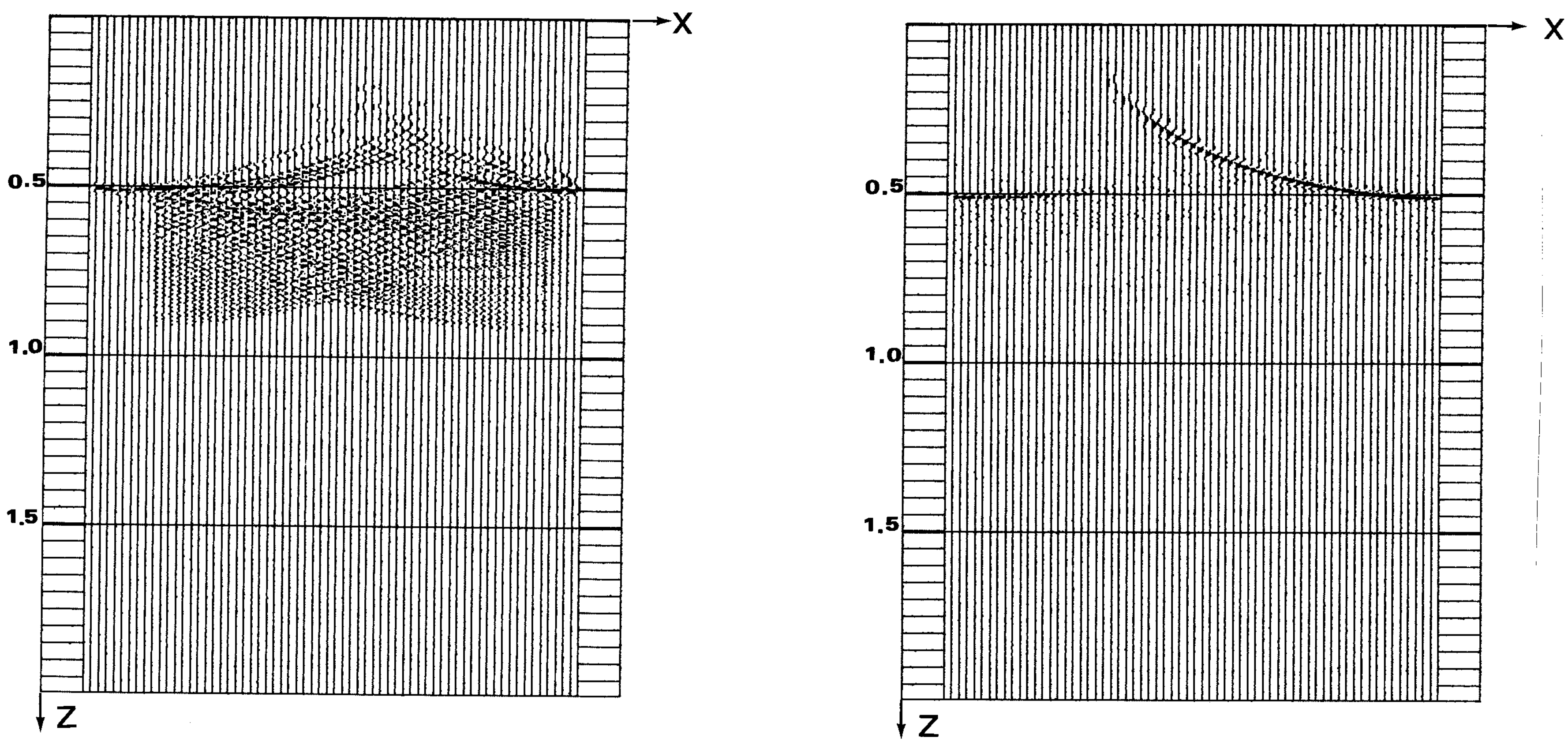
(b) Migrated section. The velocity used was 1. The inverted semicircle at the bottom associated with the spike at t = 3/4 t_{max} results from a causality problem. The other spurious semicircles are caused by the periodic boundary conditions. All of these unwanted semicircles can be made arbitrarily zero by padding zeros around the grid.



(a) (b)

FIGURE 2.—(a) Input seismic section, a synthetic hyperbola. Sample rate in X is 25 meters and in t is 4 msec, with velocity of 2000 m/sec.

(b) Migration with velocity of 1000 m/sec. The energy is moving to the left, but is obviously undermigrated. Aliasing effect along the X direction is noted.



(a) (b)

FIGURE 3.—(a) Migration of synthetic hyperbola in Fig. 2. The velocity was 2000 m/sec. All the energy has been collapsed to the left. Periodicity is noted to the right side.

(b) Migration of synthetic hyperbola in Fig. 2. Since the velocity was 3000 m/sec, the data are overmigrated. Periodicity is again observed to the right side.

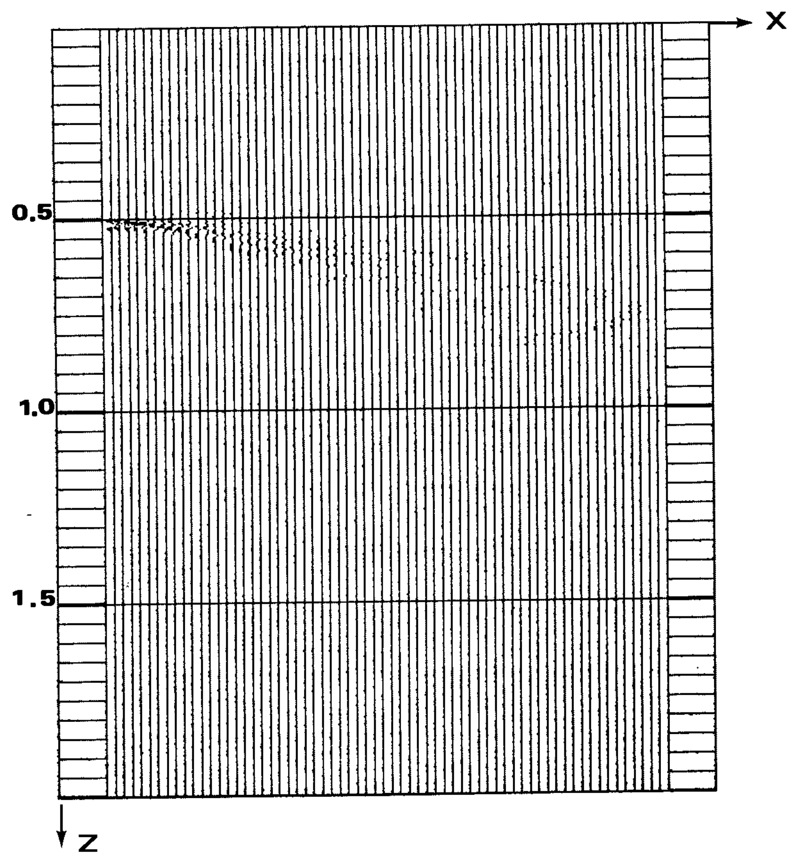


FIGURE 4.—Migration of synthetic hyperbola in Fig. 2. An explicit 15° wave equation algorithm was used. The velocity was 2000 m/sec. No periodicity effects are noted as compared with Fig. 3(a).

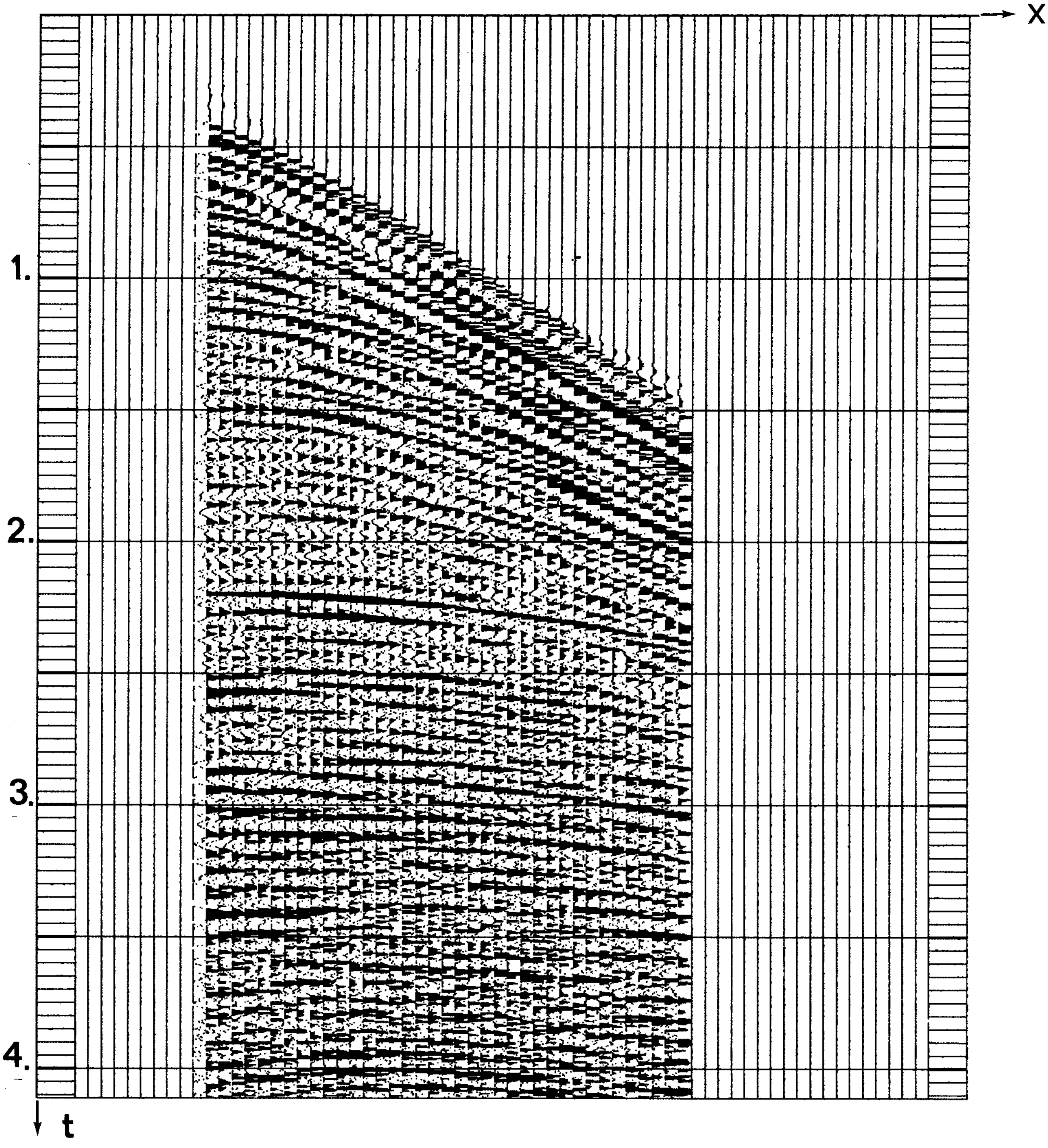


FIGURE 5.—Common shot gather input. Sample in X is 50 meters and in t is 4 msec. The number of traces in the original gather was 48, but was increased to 64 by appending zero traces to both sides of the grid (a power of two is required by the FFT algorithm). An exponential gain, exp(0.75 t), and a spherical divergence correction have been applied to the data for plotting.

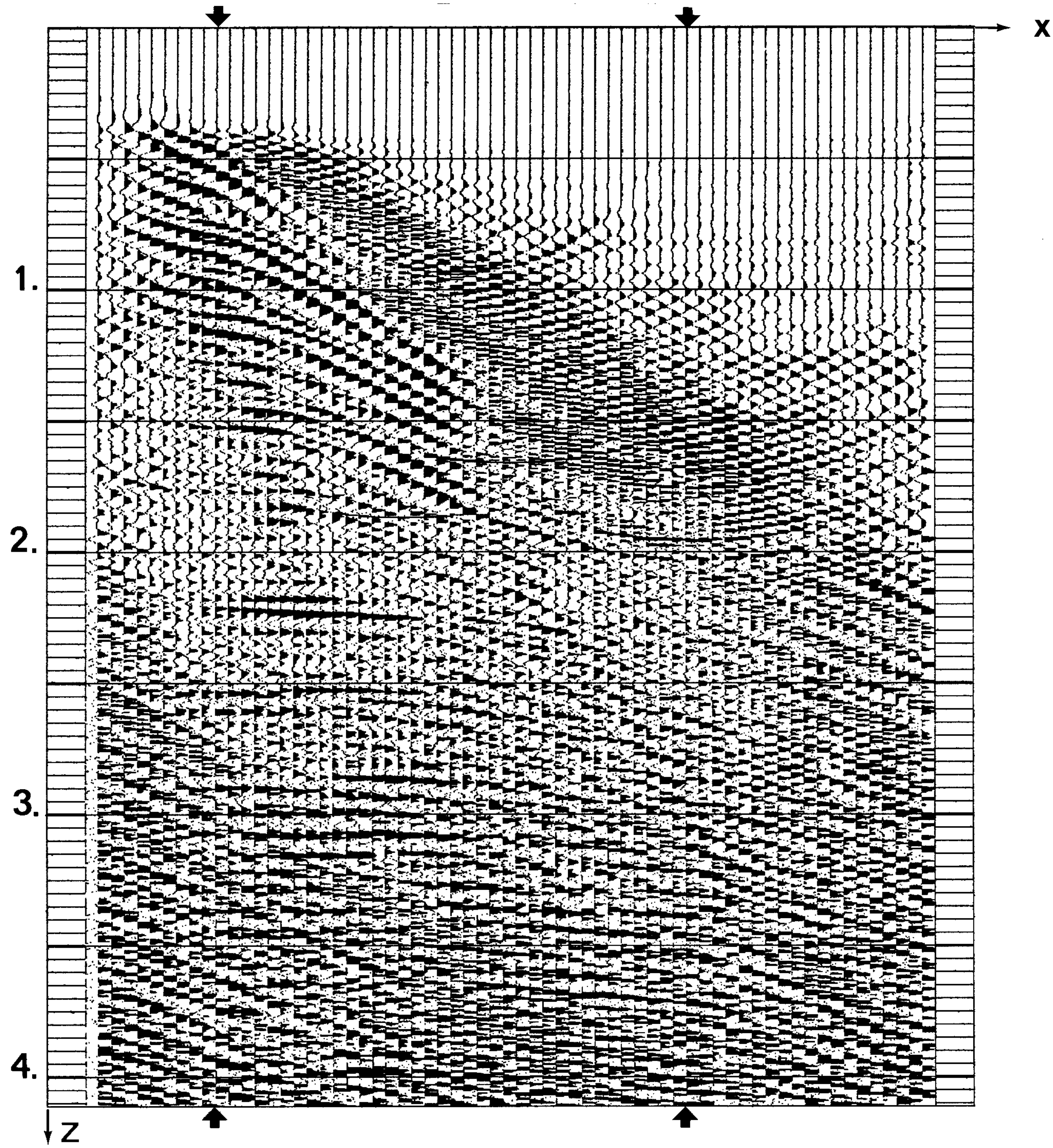


FIGURE 6.—Migrated gather of Fig. 5. The velocity used was 1000 m/sec. No zeros were appended to the bottom of the traces. Overmigration effects for the first arrivals are noted (the effect of appending zeros to the input is to reduce the effect of the periodicity [see Fig. 2(b)]). The arrows at the top mark the boundary of the original gather. The same exponential gain and spherical divergence correction were applied.

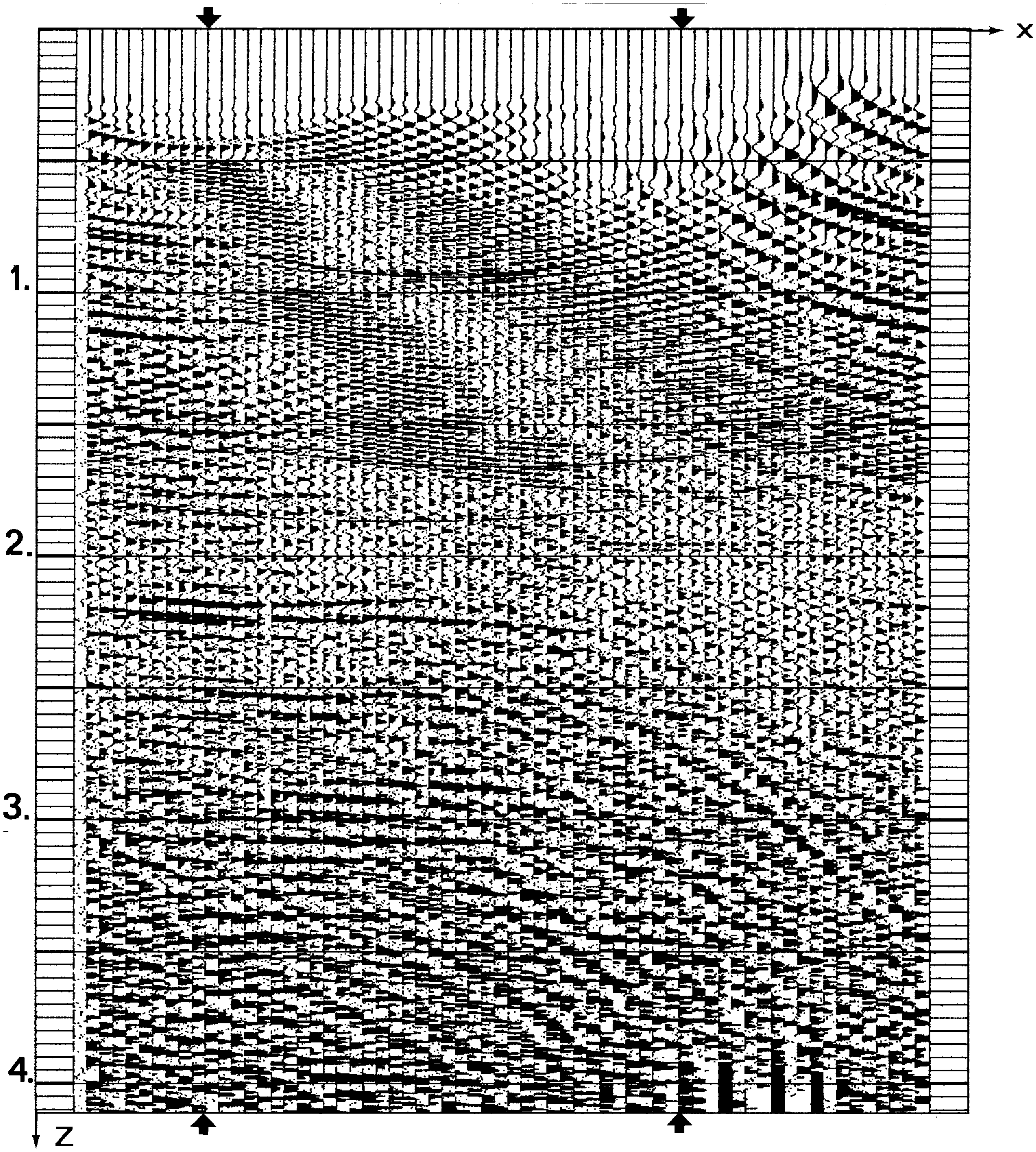


FIGURE 7.—Migration of input gather in Fig. 5. The velocity was 2000 m/sec. Nearly all the energy was moved to the left for the first arrivals. More overmigration than in Fig. 6. Notice aliasing along the X direction clearly noted between (0.5 - 2) sec, characterized by high frequencies. The appended zeros are insufficient to prevent the periodicity effect seen at the right. Event at about 2.3 sec apparently shows under- and overmigration effects. The arrows at the top and bottom mark the boundary of the original gather.

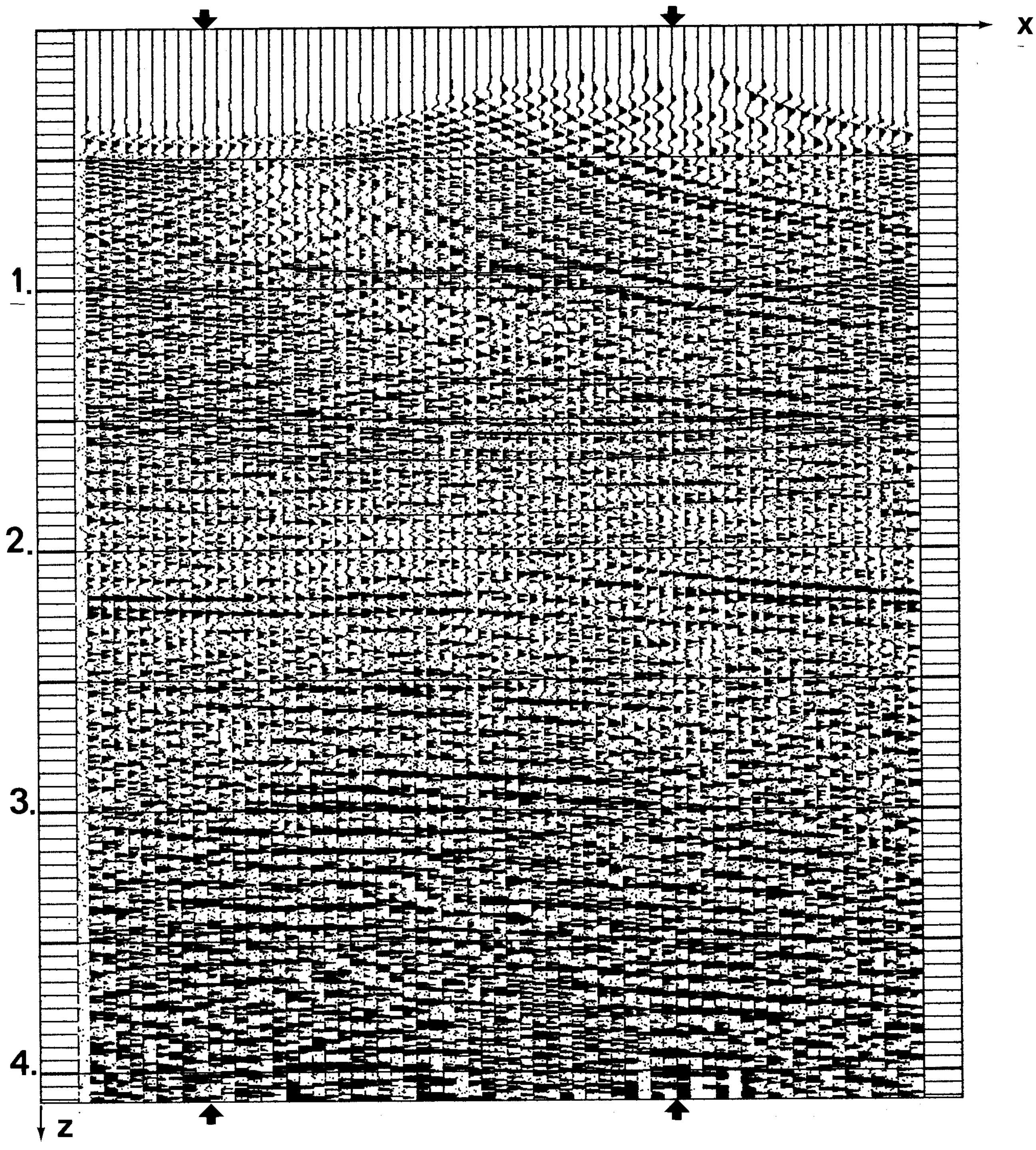


FIGURE 8.—Migration of input gather in Fig. 5 with velocity of 3000 m/sec. Nearly all the section is overmigrated. The event at about 2.3 sec is now creeping in at the far (right) side. Compare the same event in Fig. 7. The arrows at the top and bottom mark the boundary of the original gather.