A STABLE LATERAL VELOCITY ESTIMATION SCHEME

Walt Lynn

In the last SEP report, an rms velocity estimation scheme was presented based on keeping the second derivative of the velocity with respect to midpoint in the normal moveout equations (see Lynn, SEP-14, pp. 95-118). The motivation for this technique was to estimate velocity variations of less than a cable length, something that conventional coherency-type velocity estimation schemes cannot do. To do so we considered the fact that the traveltime for a given event to a given offset is dependent both on the rms velocity and its lateral derivatives beneath each midpoint. Expanding the rms velocity beneath each midpoint in a second order Taylor series and assuming straight ray paths, we obtained a tridiagonal system of equations that could be inverted to obtain the velocity.

In the tridiagonal method, we discovered that there is an inherent instability problem. Figure 7d on page 106 of SEP-14 illustrates this. This is bad news, especially when we have to iteratively solve for both the velocity and the depth to the reflector. In this paper we will show how to avoid the instability problems by going to a fourth order scheme. The motivation for the fourth order scheme is only to stabilize the method, not to gain higher resolution.

To see the cause of the problem we start with Equation (1) of the SEP-14 article:

$$t_j = (f^2 + 4z_j^2)^{1/2} \left(w_j + \frac{f^2}{24} w_j''\right) \tag{1}$$

where t_j is the traveltime for a given ray to the offset f reflecting off an interface of depth z, and w_j is the rms slowness (1/velocity) at the j-th midpoint. Writing (1) in finite difference form,

$$\frac{t_j}{a_j} = rw_{j-1} + (1-2r)w_j + rw_{j+1}, \qquad j = 1,2,.....n \tag{2}$$

where $a_j = (f^2 + 4z_j^2)^{1/2}$, $r = f^2/(24\Delta y^2)$, and Δy is the midpoint spacing. Equation (2) represents a tridiagonal system of equations with $(1 - 2r)$ on the diagonal and r on the off-diagonal of the matrix. The heart of the instability lies in the fact that the diagonal term is $(1 - 2r)$ and not the familiar $(1 + 2r)$ we get when applying the Crank-Nicolson scheme to the wave equation. With zero-slope boundary conditions, the eigenvalues λ of the tridiagonal matrix can be shown to be

$$\lambda_k = 1 - 4\, r \cos^2\left(\frac{k\pi}{2n}\right) \qquad k = 1,2,.....n \tag{3}$$

Hence, if r is not less than one-quarter, the potential for zero eigenvalues arises. Since r is inversely related to Δy^2 we circumvented this problem by estimating the second derivative using midpoints separated by several Δy's. Such a scheme has several undesirable aspects; the main ones are having to decouple the velocity estimates in between adjacent midpoints and imposing boundary conditions for each individual problem. In other words, we would solve for, say, v_1 v_6 v_{11}, etc., with one set of equations. Next we would solve for v_2 v_7 ..., and so on.

A better means of avoiding the zero eigenvalues can be seen by considering the continuous case, *i.e.* infinitesimal midpoint spacing. Taking the Fourier transform of Equation (1) with respect to midpoint y, we obtain

$$\text{F.T.}\left\{\frac{t(y)}{a(y)}\right\} = \left(1 - \frac{f^2 k_y^2}{24}\right) W(k_y) \tag{4}$$

A plot of the "dispersion relation" is shown in Figure 1.

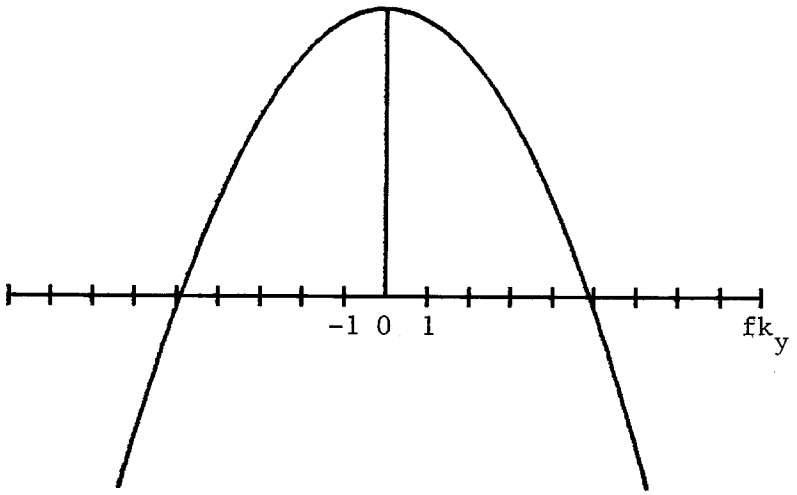

FIGURE 1.

Thus, at some particular value of fk_y, the filter will go to zero, making its inverse infinite and the process unstable. To make the process unconditionally stable we need to keep the curve from crossing zero. Since we do not want to alter the curve near $fk_y = 0$ (long wavelengths) we will consider making an adjustment of the form, $1 - (fk_y)^2/24 + \alpha(fk_y)^4$, where $\alpha > 0$. This is precisely what we would get if we had expanded $w(y)$ in a fourth order Taylor's series expansion in the original traveltime approximation. Assuming a straight ray path and expanding $w(y)$ about $w(y_o) = w_o$ we obtain

$$t = \frac{1}{\sin\theta} \int_{-f/2}^{f/2} (w_o + yw_o' + \frac{y^2}{2} w_o'' + \frac{y^3}{6} w_o''' + \frac{y^4}{24} w^{iv}) \, dy$$

where θ is the angle the ray makes with the vertical. So

$$\frac{t}{a} = w_o + \frac{f^2}{24} w_o'' + \frac{f^4}{1920} w_o^{iv} \qquad (5)$$

where a, again, is $(f^2 + 4z^2)^{1/2}$.

Fourier transforming Equation (5),

$$\text{F.T.}\left\{\frac{t(y)}{a(y)}\right\} = \left(1 - \frac{f^2 k_y^{\,2}}{24} + \frac{f^4 k_y^{\,4}}{1920}\right) W(K_y) \qquad (6)$$

A plot of the frequency domain operator is shown in Figure 2.

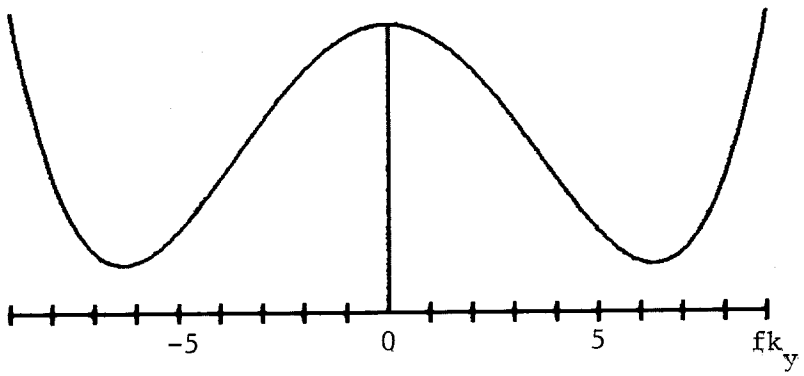

FIGURE 2.

The fact that the operator does not reach zero for any value of fk_y in the continuous case does not necessarily guarantee that the eigenvalues will not reach zero in the discrete case. To see what happens, let's just try it. We first need to put Equation (5) in finite difference form. Using $(-1, 4, 6, -4, 1)/\Delta y^4$ as the fourth derivative operator, we obtain

$$\frac{t_j}{a_j} = w_j + cw_{j-1} - 2cw_j + cw_{j+1} + dw_{j-2} - 4dw_{j-1} + 6dw_j$$

$$-4dw_{j+1} + dw_{j+2}$$

where

$$c = \frac{f^2}{24\Delta y^2}$$

and

$$d = \frac{f^4}{1920\Delta y^4}$$

Collecting terms together yields the pentagonal system of equations

$$\frac{t_j}{a_j} = dw_{j-2} + (c-4d)w_{j-1} + (1-2c+6d)w_j + (c-4d)w_{j+1} + dw_{j+2}, \quad j = 1,2,.....n \tag{7}$$

A subroutine to solve pentagonal systems of equations is given in the appendix. We will now apply this equation to the model used in SEP-14, as well as a new model which simulates a lateral gradient of velocity. We will refer to these as Model 1 and Model 2 respectively.

Model 1

For convenience, Model 1 is shown again in Figure 3. The cause of the lateral inhomogeneity is a truncated low velocity layer of 5400 feet/sec imbedded between 5000 and 5200 feet. Details of the model are given in the figure caption and are discussed in detail in the SEP-14 report. The results of using a tridiagonal or second derivative scheme to estimate the velocity to the interfaces at 6000, 7000, and 8000 feet are shown in Figure 4 as the fat line. The velocity was estimated using the traveltimes to the 2000-foot offset, and the depth to the reflectors was assumed to be known. The correct vertical rms velocity is shown as the thin straight line. In this example, the second derivative was estimated using midpoints separated by $5\Delta y$, *i.e.*

$$w_j'' \cong \frac{1}{5\Delta y^2}(w_{j-5} - 2w_j + w_{j+5})$$

The results of the second derivative method are encouraging but show some short wavelength jitter due to small eigenvalues for large fk_y. A further inconvenience of the second derivative method is having to impose boundary conditions on each set of tridiagonal equations. In this example, five different sets of equations were solved for the velocity, so at the five midpoints at each end the velocity is determined by boundary conditions. If we had used a larger offset to determine the velocity, the number of equations would have been greater (because $k\Delta y$ has to be greater to insure stability), and even more midpoints at the end of the section would be determined by the boundary conditions. Clearly this is undesirable but it

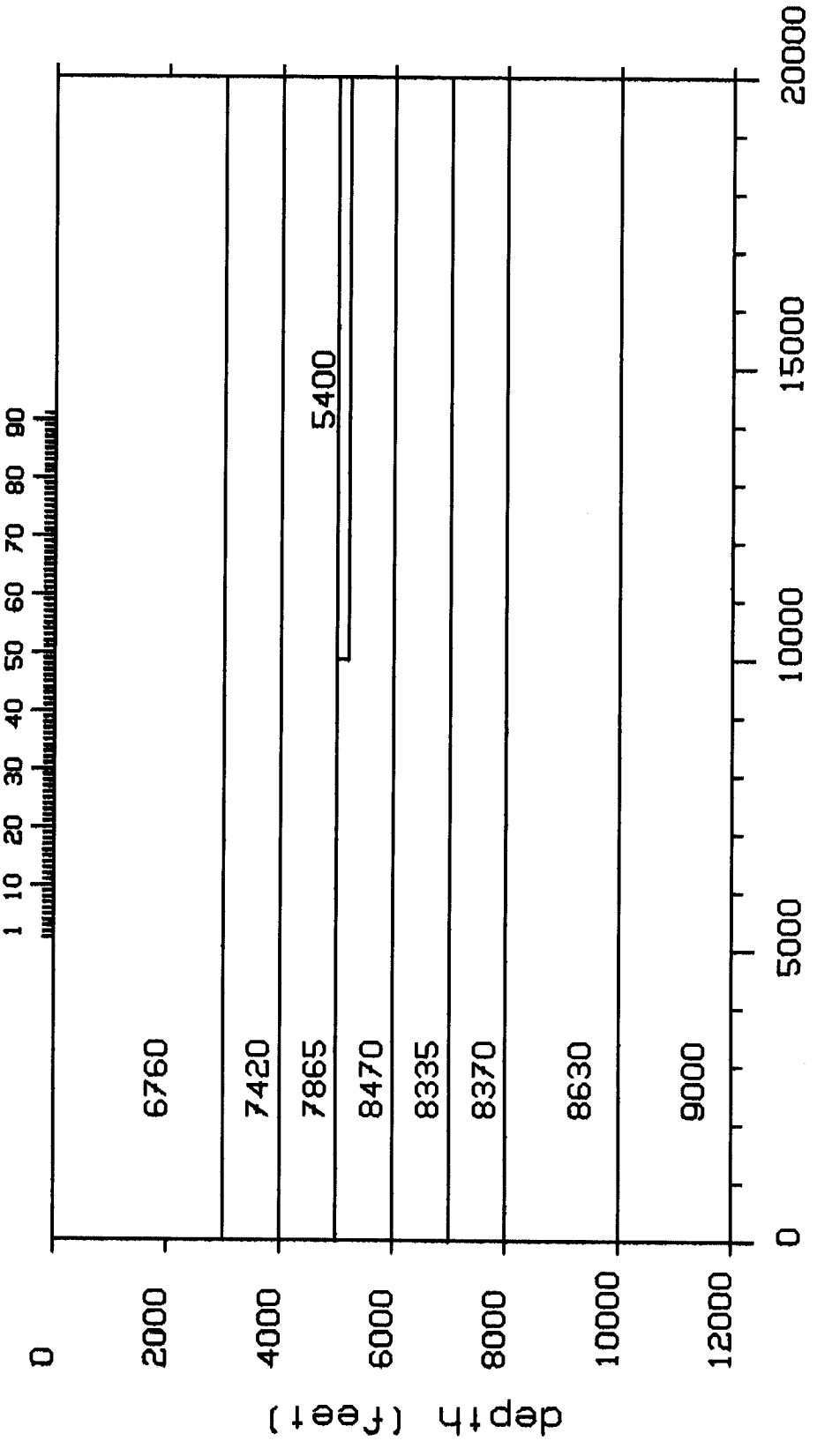

FIGURE 3.--Model 1. Model used to generate synthetic seismograms for velocity estimation. A total of 91 cmp gathers were generated using a ray-tracing program with locations shown at the top of the figure. The cmp gathers are 24-fold, with a midpoint spacing of 100 feet and a group spacing of 200 feet. Distance to the far offset is 5000 feet. The synthetic traces consist of 3 sec of 4 msec data. The velocities are given in ft/sec. The discontinuity occurs at cmp #49.

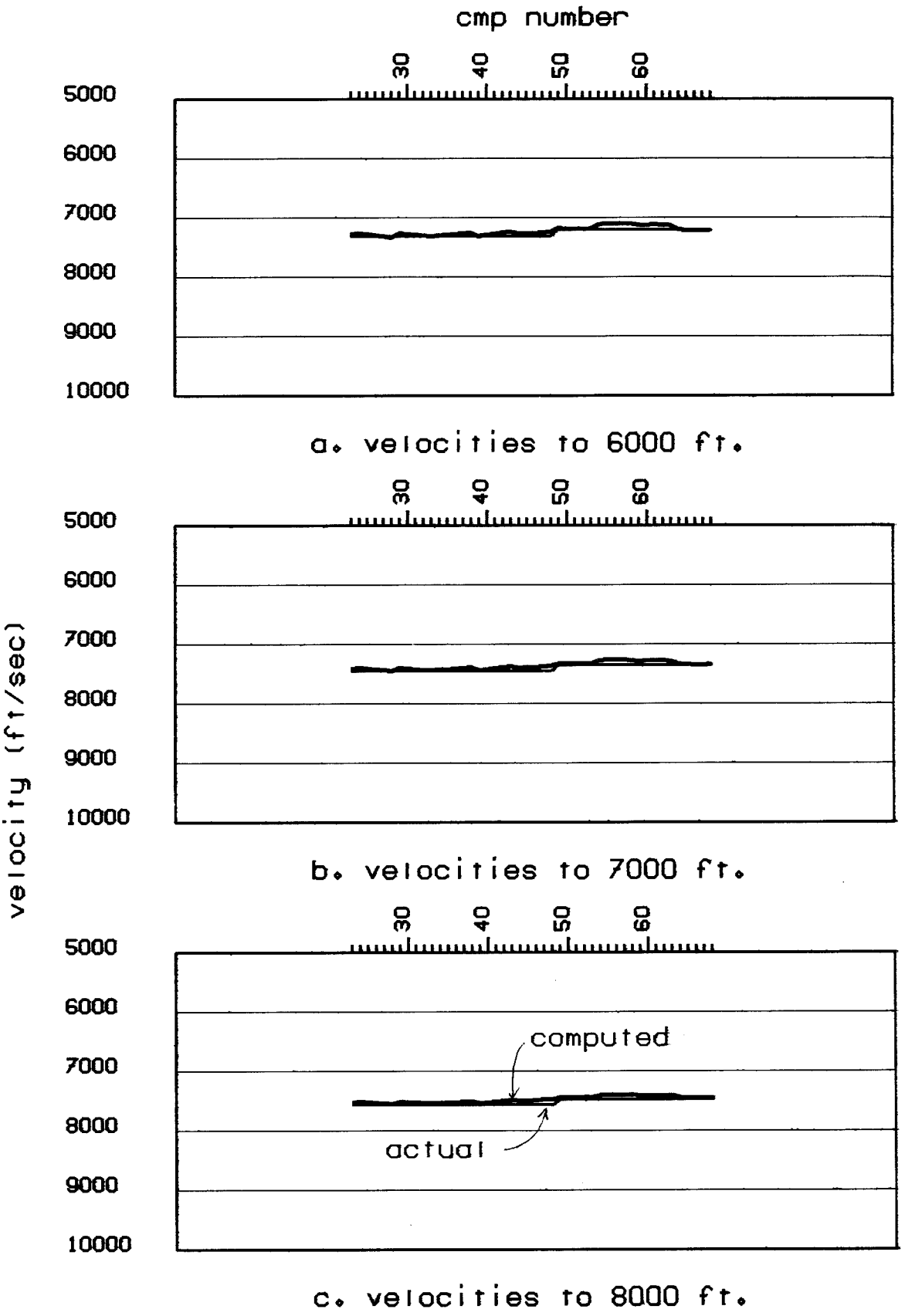

FIGURE 4.--Rms velocities to the interfaces at 6000, 7000, and 8000 feet of Model 1, using the second derivative method. The actual rms velocity is shown as the light line in each figure.

is necessary in the tridiagonal scheme. Going to a fourth derivative scheme avoids this problem as the pentagonal system of equations is stable for any Δy. Thus, in the fourth derivative scheme, only the velocities at two midpoints at each end are determined by the boundary conditions.

The results of estimating the velocity to the same interfaces of Model 1 by the fourth derivative scheme [(Equation (7)] are shown in Figure 5. We see that all of the ripple in the estimated velocity from the second derivative method is now gone. The estimated velocity is very close to the actual velocity at all midpoints except in the area of the discontinuity, where it is slightly low. The discrepancy occurs because we are trying to fit a step function with a fourth order polynomial.

Model 2

To see if the derivative methods would work over an area with a smoother lateral velocity gradient we devised the model shown in Figure 6. As with Model 1, a ray-tracing program was used to generate the common midpoint gathers. The gathers are 24-fold with a near and far offset of 400 and 9600 feet respectively.

A standard, semblance-type velocity estimation was used to estimate the velocity to the horizontal interface at 8000 feet, and the result is shown in Figure 7. The true vertical rms velocity to the interface is shown as the lighter line. Although the interface is horizontal in depth it is not on a time section, which accounts for most of the discrepancy between the true and measured rms velocities between midpoints 70 to 120 and 137 to 154. Around midpoints 50 to 130 the semblance measure gives a very poor estimate of the true velocity, due to the abrupt lateral changes in velocity.

Assuming a knowledge of the depth to the reflector, the velocity was estimated three different times by using the traveltimes from three different common offset sections corresponding to 1600, 3200, and 6400 feet. The 1600-foot common offset section is shown in Figure 8. The gaps seen in some of the reflectors are due to limitations in the ray-tracing program. The second derivative method was tried first to estimate the velocity giving the results displayed in Figure 9. The number of independent equations solved for the three cases were 3, 5, and 10. The velocity estimates are in close agreement with the true velocity shown in Figure 7. The short wavelength jitter is still

present, however. The flattening of the velocity curve at the right is due to the zero-slope boundary conditions. Note how the flattening spans over more midpoints for the larger offset.

The corresponding results using the fourth derivative method are shown in Figure 10. The velocity profile is again in close agreement with the true rms velocity and the curves are much smoother than those in the second derivative method.

A close comparison of the three different results in Figure 10 shows that the computed velocities are not quite the same, although the differences are small. This is due to the straight ray approximation made in the derivation of the equations.

In all of the cases shown we assumed that the depth to the reflector was known. Obviously, in many cases this is an unfair assumption, since obtaining the reflector depth from a seismic section requires some knowledge of the velocity. In such cases the reflector depth must be iteratively solved for along with the velocity. A first guess might be the near-offset traveltime divided by 2 * 8000 ft/sec. In order to insure stability the reflector depths should be smoothed before re-estimating the velocity. In a future paper we will show how to minimize the depth dependence by either working with w^2 and two different common offset sections or by using two different common midpoint slant stack sections.

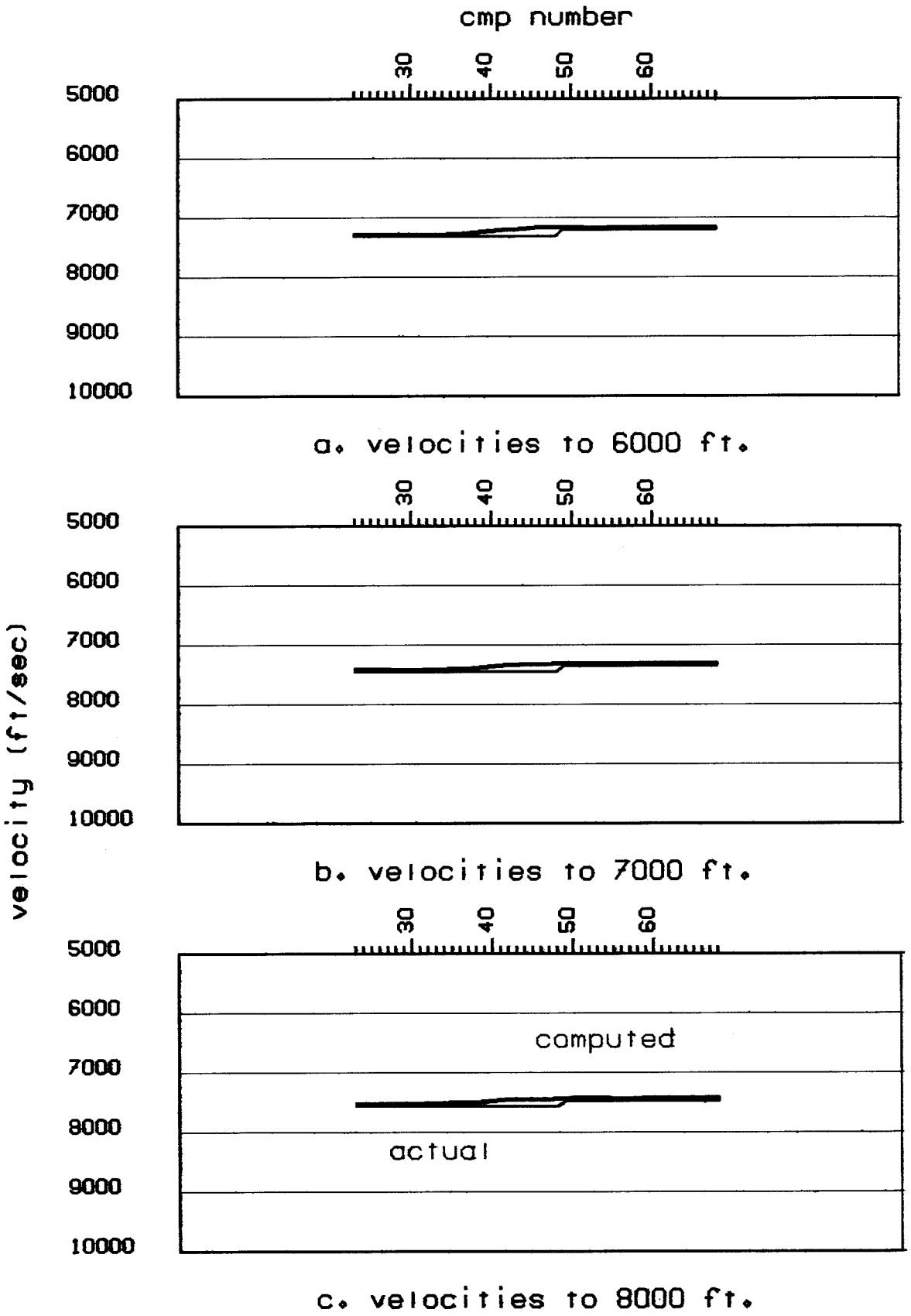

FIGURE 5.--Rms velocities to the interfaces at 6000, 7000, and 8000 feet of Model 1, using the fourth derivative method. Actual rms velocity is shown on the light line.

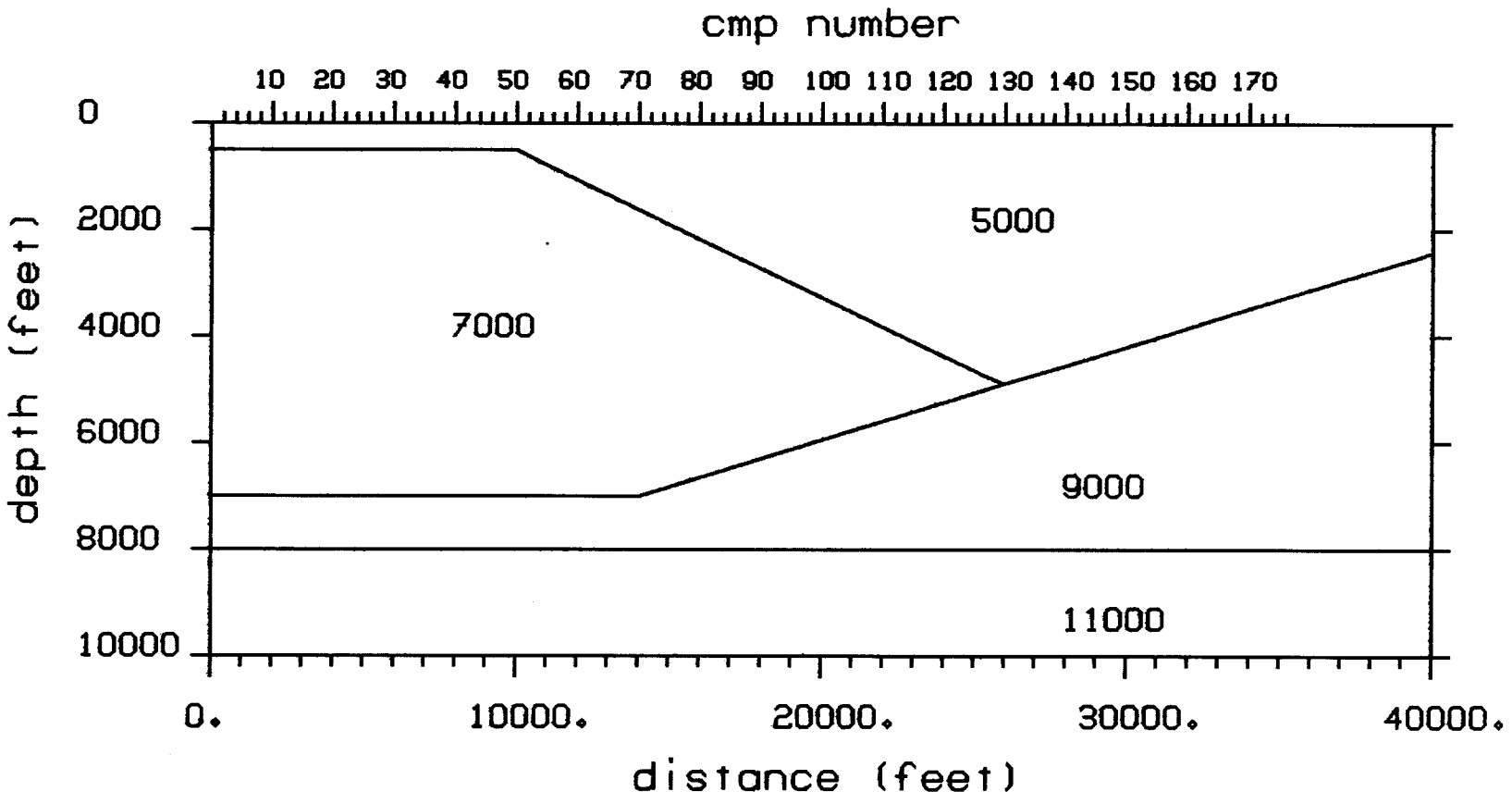

FIGURE 6.--Model 2. Cmp gathers are 24-fold with a midpoint spacing of 200 feet and a group spacing of 400 feet. Velocities are given in ft/sec.

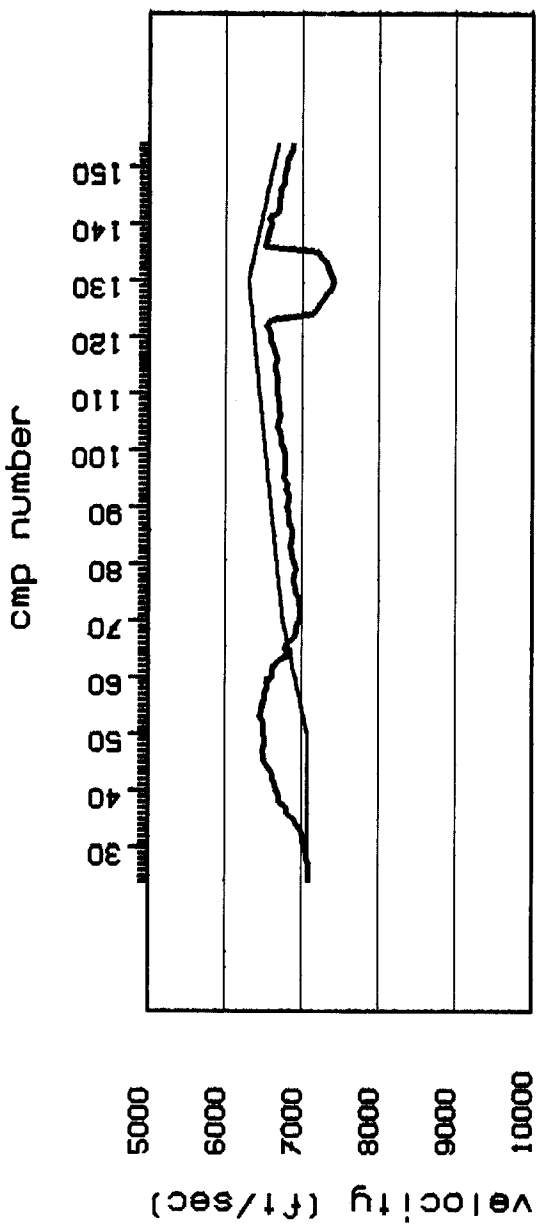

FIGURE 7.--Correct vertical rms velocity (light line) and stacking velocity computed using a semblance technique (heavy line) for Model 2.

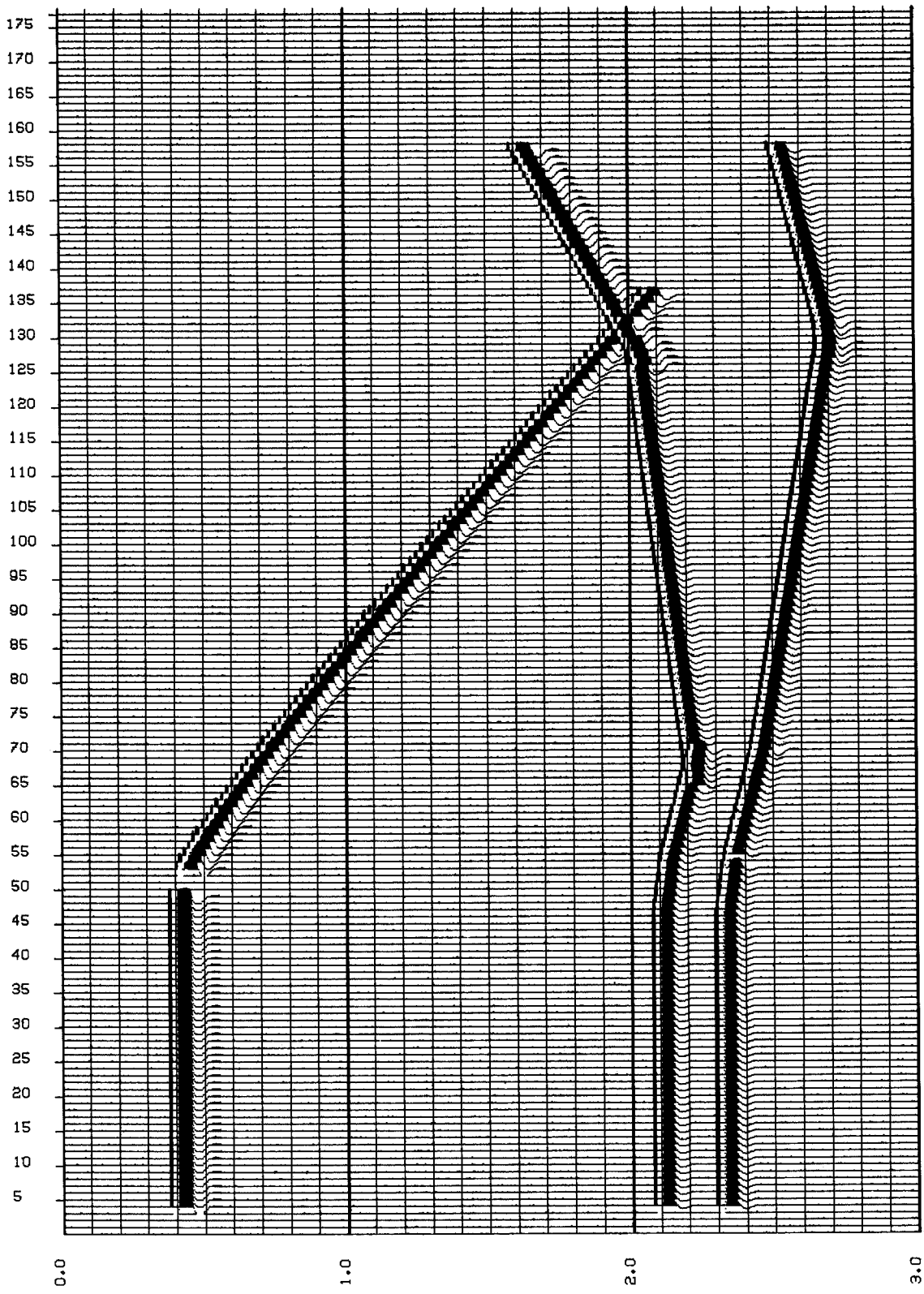

FIGURE 8.--Constant offset section (1600 feet) of Model 2.

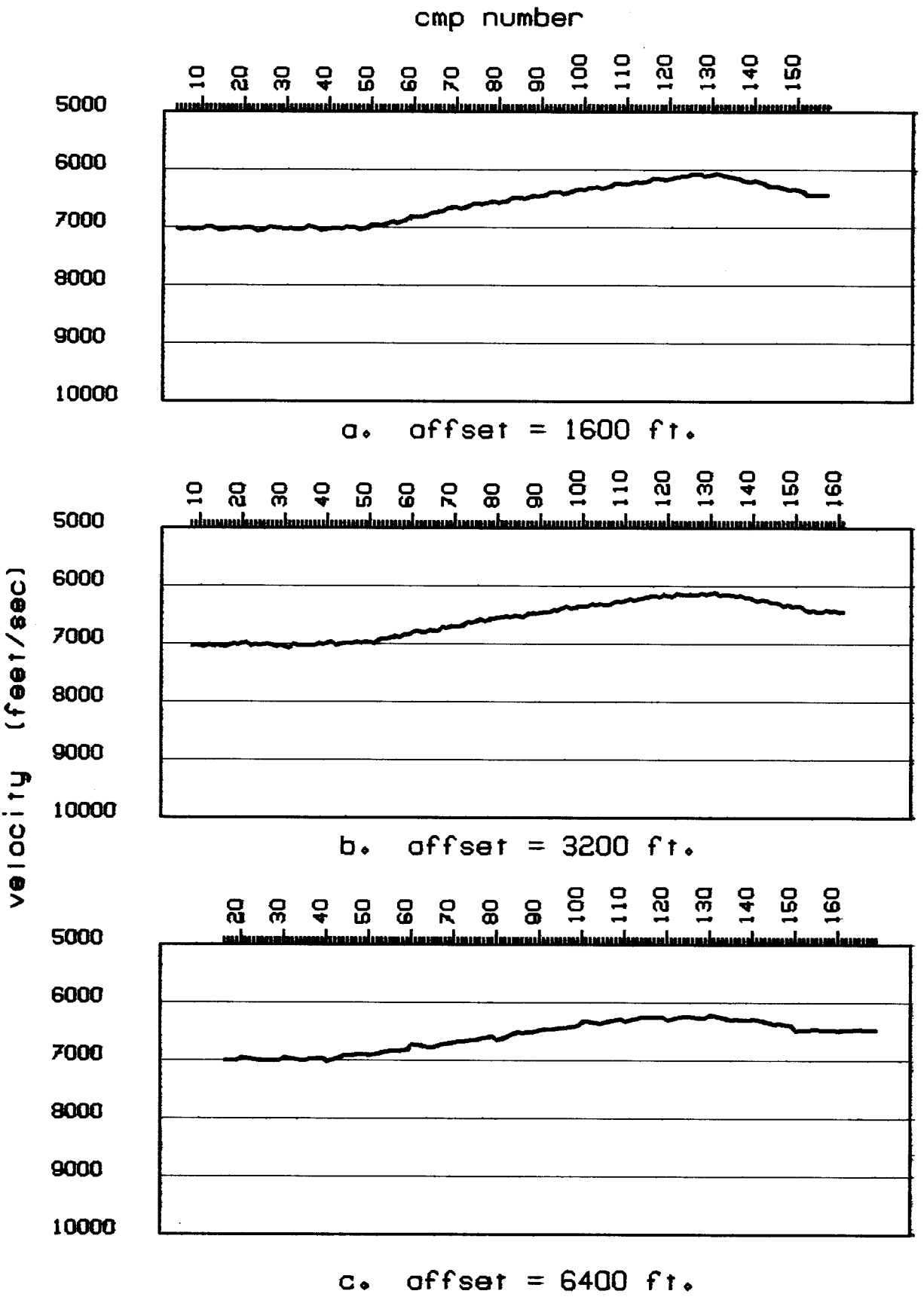

FIGURE 9.--Velocity estimates to the 8000-foot interface of Model 2 using the second derivative method on constant offset sections of 1600, 3200, and 6400 feet.

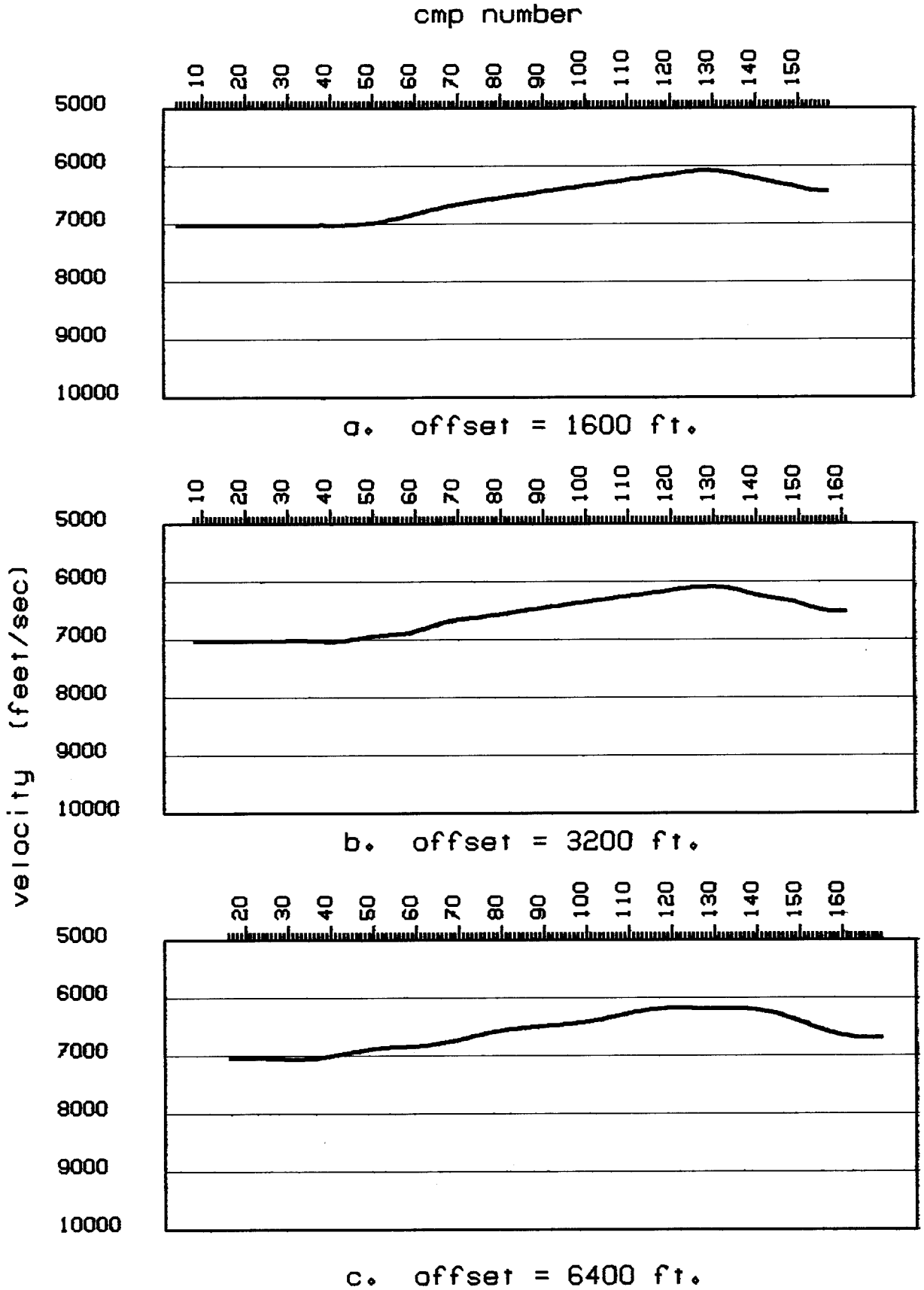

FIGURE 10.--Velocity estimates to the 8000-foot interface of Model 2 using the fourth derivative method on constant offset sections of 1600, 3200, and 6400 feet.

APPENDIX

```
c       subroutine pent(a,b,c,d,e,f,n,t,q,r,s,ibc)
c
c       solves the pentagonal system of equations:
c
c         a(k)t(k-2) + b(k)t(k-1) + c(k)t(k) + d(k)t(k+1)
c
c                + e(k)t(k+2) = f(k),
c
c       where, k = 1,2,...,n
c
c       using the recursion relations:
c
c         t(k)   = q(k)t(k+2)   + r(k)t(k+1)   + s(k)
c
c         t(k-1) = q(k-1)t(k+1) + r(k-1)t(k)   + s(k-1)
c
c         t(k-2) = q(k-2)t(k)   + r(k-2)t(k-1) + s(k-2)
c
c       set ibc = 0 to solve the n equations exactly (i.e. no b.c.'s)
c               = 1 for zero first and second derivative b.c.'s.
c
        subroutine pent(a,b,c,d,e,f,n,t,q,r,s,ibc)
        dimension a(n),b(n),c(n),d(n),e(n),f(n),t(n),q(n),r(n),s(n)
c
c
c       left boundary conditions
c
        if(ibc.eq.1) goto 3
c
c       solve the set of equations exactly.  No b.c.'s.
c
        q(1) = -e(1)/c(1)
        r(1) = -d(1)/c(1)
        s(1) =  f(1)/c(1)
        den  = c(2) - d(1)*b(2)/c(1)
        q(2) = -e(2)/den
        r(2) = (e(1)*b(2)/c(1) - d(2))/den
        s(2) = (f(2) - f(1)*b(2)/c(1))/den
        goto 5
c
c       zero slope and zero second derivative b.c.'s
c       t(1) = t(2) = t(3)
c
c
  3     q(1) = 0.
        r(1) = 1.
        s(1) = 0.
        q(2) = 0.
        r(2) = 1.
        s(2) = 0.
c
```

```
  5      n1 = n-1
         n2 = n-2
         n3 = n-3
         do 10 k=3,n
           den = a(k)*q(k-2) + r(k-1)*( a(k)*r(k-2) + b(k)) + c(k)
           q(k) = -e(k)/den
           r(k) = -( q(k-1)*(b(k) + a(k)*r(k-2)) + d(k) )/den
           s(k) = ( f(k) - ( s(k-1)*(b(k) + a(k)*r(k-2)) + a(k)*s(k-2)))/den
  10     continue
c
c        right boundary conditions
c
         if(ibc.eq.1) goto 13
c
c        solve the equations exactly
c
         g1 = a(n)*q(n2) + c(n)
         g2 = a(n)*r(n2) + b(n)
         g3 = f(n) - a(n)*s(n2)
         h1 = a(n1)*r(n3)*q(n2) + b(n1)*q(n2) + d(n1)
         h2 = a(n1)*q(n3) + a(n1)*r(n3)*r(n2) + b(n1)*r(n2) + c(n1)
         h3 = f(n1) - a(n1)*r(n3)*s(n2) - a(n1)*s(n3) - b(n1)*s(n2)
         t(n) =  (g3 - g2*h3/h2)/(g1 - g2*h1/h2)
         t(n1) = (g3 - g1*h3/h1)/(g2 - g1*h2/h1)
         goto 15
c
c        zero slope and zero second derivative.
c        t(n2) = t(n1) = t(n)
c
  13     t(n) = s(n2)/(1.-(q(n2)+r(n2)))
         t(n1) = t(n)
c
  15     continue
         do 20 j=1,n2
           k = n1-j
  20     t(k) = q(k)*t(k+2) + r(k)*t(k+1) + s(k)
         return
         end
```