Difference Approximations of Waves in Slanted Frames

by Bjorn Engquist

We will consider approximations of the one way wave equation in slanted coordinates using the following coordinate transformation for upcoming waves (SEP-7, p. 30):

$$\begin{aligned} t' &= t + \frac{z}{v}\cos\theta - \frac{x}{v}\sin\theta \\ x' &= x + z\tan\theta \\ z' &= z \end{aligned} \qquad (1)$$

$$\begin{aligned} \omega &= \omega' \\ k_x &= k'_x + \frac{\omega'}{v}\sin\theta \\ k_z &= k'_z + k'_x\tan\theta - \frac{\omega'}{v}\cos\theta \end{aligned} \qquad (2)$$

When this transformation is applied to the wave equation dispersion relation

$$k_z = -\left(\frac{\omega^2}{v^2} - k_x{}^2\right)^{1/2} \qquad (3)$$

we get

$$k'_z + k'_x\tan\theta - \frac{\omega'\cos\theta}{v} = -\left(\frac{\omega'^2}{v^2} - \left(k'_x + \frac{\omega'}{v}\sin\theta\right)^2\right)^{1/2} \qquad (4)$$

From now on we will only use the slanted coordinates, but for convenience drop all primes in the formulas.

By using rational approximations of the square root in (4) we can derive the following differential equations describing the inverse problem for upcoming waves [SEP-8, p. 23 or p. 60].

$$P_{tz} + \frac{v \tan\theta}{2\cos\theta} P_{xz} + \frac{v}{2\cos^3\theta} P_{xx} = 0 \tag{5}$$

$$P_{ttz} + \frac{v \tan\theta}{\cos\theta} P_{txz} - \frac{v^2}{4\cos^2\theta} P_{xxz} + \frac{v}{2\cos^3\theta} P_{txx} = 0 \tag{6}$$

$$0 \le t \le T \quad , \quad x_{min} \le x \le x_{max} \quad , \quad z \ge 0$$

The function P is given initially at $z = 0$. Boundary conditions are given at $t = T$, $x = x_{min}$ and $x = x_{max}$. For equation (6) we need two boundary conditions at $t = T$. See SEP-8, p. 61 for a discussion of the well posedness of these problems.

The dispersion relations of (5), (6) and other equations are studied elsewhere in this volume. There we see that (6) gives a good approximation to the wave equation in a wider range of angles $(\frac{k_x}{\omega})$ than what (5) does.

We will here derive difference approximations to (5) and (6) and analyze them with respect to accuracy and stability. The main purpose is to present the structure of some stable difference methods. They can then be changed in different ways to suit certain purposes. We can have other coefficients in (5) and (6) and also often modify the difference operators. In what follows we will think of the difference equations as approximations of (5) and (6). It is of course also possible to look at them as direct approximations of the scalar wave equation.

Let $P^n_{j,k}$ be the discrete approximation of $P(t, x, z)$ on a mesh at the point (t_j, x_k, z^n). In order to simplify the notation we define some difference operators (unchanging indices are dropped).

$$D^x_+ P_k = \frac{P_{k+1} - P_k}{\Delta x} \qquad (\sim P_x)$$

$$D^x_- P_k = \frac{P_k - P_{k-1}}{\Delta x} \qquad (\sim P_x)$$

$$D^x_0 P_k = \frac{P_{k+1} - P_{k-1}}{2\,\Delta x} \qquad (\sim P_x)$$

(D^t_+ , D^z_+ , ... denotes the analogous approximations of $\frac{\partial}{\partial t}, \frac{\partial}{\partial z}, \ldots$)

$$D_2 P_j{}^n = \frac{1}{2\Delta t^2}(P^n_{j+2} - P_j{}^n - P^{n+1}_{j+1}) \qquad (\sim P_{tt})$$

We will also use a number of averaging operators

$$Q_1 P_j{}^n = \frac{1}{2}(P^n_{j+1} + P^{n+1}_{j-1})$$

$$Q_2(\alpha) P_k = \alpha P_{k-1} + (1-2\alpha) P_k + \alpha P_{k+1}$$

$$Q_3 P_j = \frac{1}{2}(P_j + P_{j+1})$$

$$Q_4 P_j{}^n = \frac{1}{4}(P_j{}^n + P^n_{j+1} + P^{n+1}_j + P^{n+1}_{j+1})$$

$$Q_5 P_j = \frac{1}{2}(P_{j-1} - P_{j+1})$$

$$Q_6 P^n = \frac{1}{2}(P^n + P^{n+1})$$

The differential equation (5) can now be approximated by the following three schemes ($t_j = T - j\,\Delta t$)

$$\left(D_0^t D_+^z - \frac{v\tan\theta}{2\cos\theta} D_0^x D_+^z - \frac{v}{2\cos^3\theta} Q_1 D_+^x D_-^x \right) P^n_{j,k} = 0 \tag{7}$$

$$\left(Q_2(\alpha) D_+^t D_+^z - \frac{v\tan\theta}{2\cos\theta} Q_3 D_+^z D_0^x - \frac{v}{2\cos^3\theta} Q_4 D_+^x D_-^x \right) P^n_{j,k} = 0 \tag{8}$$

$$\left(Q_2(\alpha) \right) D_+^t D_+^z - \frac{v\tan\theta}{2\cos\theta} Q_3 D_+^z D_0^x - \frac{v}{2\cos^3\theta} \left(Q_4 - \frac{\Delta t^2}{12} D_2 \right) D_+^x D_-^x) P^n_{j,k} = 0 \tag{9}$$

and for equation (6) we will use

$$\left(D_+^t D_-^t D_+^z Q_2(\alpha) - \frac{v\tan\theta}{\cos\theta} D_0^t D_0^x D_+^z - \frac{v^2}{4\cos^2\theta} Q_5 D_+^x D_-^x D_+^z - \right.$$

$$\left. - \frac{v}{2\cos^3\theta} Q_6 D_0^t D_+^x D_-^x \right) P^n_{j,k} = 0 \tag{10}$$

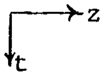
Let us write these schemes in the form of difference molecules with the undivided differences

$$d_0 = 2\,\Delta x\, D_0^x \quad , \quad d_2 = \Delta x^2 D_+^x D_-^x$$

and with

$$c_1 = -\frac{\Delta t\ v\tan\theta}{8\,\Delta x\cos\theta} \quad , \quad c_2 = -\frac{\Delta t\ \Delta z\ v}{8\Delta x^2\cos^3\theta}$$

$$c_3 = -\frac{\Delta t^2\ v}{8\Delta x^2\cos^2\theta}$$

The axes have the direction 

. We write the different t and z levels explicitly only for (7')

$$\begin{array}{r|c|c|} \hline j-1 \ldots & 1 & -1 \\ \hline j \ldots\ldots & & \\ \hline j+1 \ldots & -1 & 1 \\ \hline & n & n+1 \end{array} + 4c_1 \begin{array}{|c|c|} \hline & \\ \hline -1 & 1 \\ \hline & \\ \hline \multicolumn{1}{c}{n} & \multicolumn{1}{c}{n+1} \end{array} d_0 + 4c_2 \begin{array}{|c|c|} \hline & 1 \\ \hline & \\ \hline 1 & \\ \hline \multicolumn{1}{c}{n} & \multicolumn{1}{c}{n+1} \end{array} d_2 \qquad (7')$$

$$\begin{array}{|c|c|} \hline 1 & -1 \\ \hline -1 & 1 \\ \hline \end{array} Q_2(\alpha) + c_1 \begin{array}{|c|c|} \hline -1 & 1 \\ \hline -1 & 1 \\ \hline \end{array} d_0 + c_2 \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} d_2 \quad (8')$$

$$\begin{array}{|c|c|} \hline 1 & -1 \\ \hline -1 & 1 \\ \hline \end{array} Q_2(\alpha) + c_1 \begin{array}{|c|c|} \hline -1 & 1 \\ \hline -1 & 1 \\ \hline \end{array} d_0 + \frac{c_2}{6} \begin{array}{|c|c|} \hline & -1 \\ \hline 7 & 6 \\ \hline 6 & 7 \\ \hline -1 & \\ \hline \end{array} d_2 \quad (9')$$

$$\begin{array}{|c|c|} \hline -1 & 1 \\ \hline 2 & -2 \\ \hline -1 & 1 \\ \hline \end{array} Q_2(\alpha) + 2c_1 \begin{array}{|c|c|} \hline 1 & -1 \\ \hline & \\ \hline -1 & 1 \\ \hline \end{array} d_0 + c_3 \begin{array}{|c|c|} \hline -1 & 1 \\ \hline & \\ \hline -1 & 1 \\ \hline \end{array} d_2 + c_2 \begin{array}{|c|c|} \hline -1 & -1 \\ \hline & \\ \hline 1 & 1 \\ \hline \end{array} d_2 \quad (10')$$

The difference approximation (7) is explicit. We need to know the initial and boundary values: $P^0_{j,k}$, $P^n_{0,k}$ and $P^n_{1,k}$. Since every difference is centered it is easy to check that (7) has local truncation error of second order. That is, if the full difference operator is applied to any smooth function P , which is a solution to (5), the result will be of the order $O(\Delta t^2 + \Delta x^2 + \Delta z^2)$.

There are of course more sophisticated ways than using the order of approximation to describe the accuracy. In general, a higher order method is more accurate measured in different ways and requires fewer points per wave than a lower order one.

The formulas (8), (9) and (10) are implicit. In each step a tridiagonal system has to be solved. In (8) the truncation error has the form $O(\Delta t^2 + \Delta x^2 + \Delta z^2)$. The scheme (8) is more compact than (7) and it gives a better approximation in the t-direction.

$$D_+^t P(t_j) = \frac{\partial}{\partial t} P(t_{j+1/2}) + \frac{\Delta t^2}{24} \frac{\partial^3}{\partial t^3} P(t_{j+1/2}) + \ldots$$

$$D_0^t P(t_j) = \frac{\partial}{\partial t} P(t_j) + \frac{\Delta t^2}{6} \frac{\partial^3}{\partial t^3} P(t_j) + \ldots$$

$$Q_1 P(t_j) = P(t_{j+1/2}) + \frac{\Delta t^2}{4} \frac{\partial^2}{\partial t^2} P(t_{j+1/2}) + \ldots$$

$$Q_3 P(t_j) = P(t_j) + \Delta t^2 \frac{\partial^2}{\partial t^2} P(t_j) + \ldots$$

The t-discretization error is here a fourth of that in (7). When $\alpha = 0$, ∂_x is replaced by D_0^x and ∂_{xx} by $D_+^x D_-^x$. If we take $\alpha = \frac{1}{12}$ and disregard the P_{xz} term the approximation is of fourth order in x. This corresponds to the $D_+^x D_-^x (I + \frac{1}{12} D_+^x D_-^x)^{-1}$ approximation of ∂_{xx}.

For $\alpha = \frac{1}{4}$ all x-derivatives are replaced by the corresponding bilinear approximations. That is, $\partial_x \sim \frac{2}{\Delta x} \frac{X-I}{X+I}$, where X is the translation operator $(X P_k = P_{k+1})$. If we discretize only in the x direction we have:

$$\partial_{tz} P_k + \frac{v \tan\theta}{2\cos\theta} \frac{2}{\Delta x} \frac{X-I}{X+I} \partial_z P_k + \frac{v}{2\cos^3\theta} \frac{4}{\Delta x^2} \frac{(X-I)^2}{(X+I)^2} P_k =$$

$$= \frac{X^2+2X+I}{4} \partial_{tz} P_k + \frac{v \tan\theta}{2\cos\theta} \frac{X^2-I}{2\Delta x} \partial_z P_k +$$

$$+ \frac{v}{2\cos^3\theta} \frac{X^2-2X+I}{\Delta x^2} P_k =$$

$$= Q_2(\tfrac{1}{4}) \partial_{tz} P_{k+1} + \frac{v \tan\theta}{2\cos\theta} D_0^{\,x} P_{k+1} + \frac{v}{2\cos^3\theta} D_+^{\,x} D_-^{\,x} P_{k+1}$$

In (9) we have used an idea from SEP-7, p. 137 to get an implicit scheme which is fourth order accurate in t. As it is written it is only second order in x and z. We can change the difference operators to improve the approximation in x. This is discussed later.

We finally approximate (6) with the compact and implicit scheme (10). It has the truncation error $O(\Delta t^2 + \Delta x^2 + \Delta z^2)$.

Let us now consider the stability of these difference equations. We want to solve them with z as evolution direction. For each n-level we also would like to calculate the new values recursively marching in the $-t$ direction (i.e., for increasing values of j). We will limit ourselves to analyzing the stability of the schemes first as pure initial value problems in z and then the stability of one sweep in t when z is fixed. The latter corresponds to a normal mode analysis of the full initial boundary value problem for infinitely rapid increasing modes in the z-direction. We saw in SEP-7, p. 146, that these two cases were essential.

In order to show stability of the initial value problem we Fourier transform the difference equation in t and x . The scheme (7) will then have the form

$$2\ i \sin(-\omega\Delta t)\ (\hat{P}^{n+1} - \hat{P}^{n}) + 8\, i\, c_1 \sin(k_x \Delta x)\ (\hat{P}^{n+1} - \hat{P}^{n}) -$$

$$- 16\, c_2 \sin^2\left(\frac{k_x \Delta x}{2}\right)\left(e^{i\omega\Delta t}\ \hat{P}^{n+1} + e^{-i\omega\Delta t}\ \hat{P}^{n}\right) = 0$$

$$\Big(i\,\big(2 \sin(-\omega\Delta t) + 8\, c_1 \sin(k_x\Delta x) - 16\, c_2 \sin^2\left(\frac{k_x\Delta x}{2}\right)$$

$$\sin(\omega\Delta t)\big) - 16\, c_2 \sin^2\left(\frac{k_x\Delta x}{2}\right)\cos(\omega\Delta t)\Big)\ \hat{P}^{n+1} =$$

$$\Big(i\,\big(2 \sin(-\omega\Delta t) + 8\, c_1 \sin(k_x\Delta x) - 16\, c_2 \sin^2\left(\frac{k_x\Delta x}{2}\right)$$

$$\sin(\omega\Delta t)\big) + 16\, c_2 \sin^2\left(\frac{k_x\Delta x}{2}\right)\cos(\omega\Delta t)\Big)\ \hat{P}^{n}$$

This formula has the structure

$$(iA + B)\ \hat{P}^{n+1} = (iA - B)\ \hat{P}^{n} \tag{11}$$

where A and B are real. Hence, $|\hat{P}^{n+1}|$ equals $|\hat{P}^{n}|$ which means that no Fourier mode can increase.

We now turn to the second question of stability in t for a fixed z . When we have transformed in x and when we only consider the n+1 level, (7) becomes

$$\hat{P}^{n+1}_{j+1} - \hat{P}^{n+1}_{j-1} + 8\, i\, c_1 \sin(k_x\Delta x)\ \hat{P}^{n+1}_{j} - 16\, c_2 \sin^2\left(\frac{k_x\Delta x}{2}\right)\hat{P}^{n+1}_{j-1} = 0$$

with the corresponding characteristic equation

$$s^2 + 8\, i\, c_1 \sin(k_x \Delta x)\, s - (1 + 16\, c_2 \sin^2(\frac{k_x \Delta x}{2})) = 0 \qquad (12)$$

The stability condition $|s| \leq 1$ ($|s| < 1$ for coinciding roots) is fulfilled for an equation $s^2 + i\,a\,s + b = 0$ (real a,b) if $|b| \leq 1$, $|a| < 1 - b$. Hence, we have stability if the following is valid.

$$|1 + 16\, c_2| \leq 1 \; ; \quad |8\, c_1| < 2 + 16\, c_2$$

or

$$\frac{\Delta t\, \Delta z\, v}{\Delta x^2 \cos^3\theta} \leq 1 \; ; \quad \frac{\Delta t\, v\, |\tan\theta|}{\Delta x\, \cos\theta} < 2 - \frac{2\, \Delta t\, \Delta z\, v}{\Delta x^2 \cos^3\theta} \qquad (13)$$

Let us change (7) a little to see how easy it is to get an unstable scheme. For the last term we will now use the operator

$$4\, c_2 \begin{array}{|c|c|} \hline & \\ \hline 1 & 1 \\ \hline & \\ \hline \end{array}\; d_0$$

It looks even more natural than the one we had in (7') and the truncation error is still of second order. We Fourier transform in x and consider the n+1 level.

$$\hat{P}^{n+1}_{j+1} - \hat{P}^{n+1}_{j-1} + (\, 8\, i\, c_1 \sin(k_x \Delta x) - 16\, c_2 \sin^2(\frac{k_x \Delta x}{2})\,)\, \hat{P}^{n+1}_{j} = 0$$

The corresponding characteristic equation is

$$s^2 + (\, 8\, i\, c_1 \sin(k_x \Delta x) - 16\, c_2 \sin^2(\frac{k_x \Delta x}{2})\,)\, s - 1 = 0$$

It has one root $|s| > 1$ for all combinations of Δt, Δx and Δz, when $\sin^2 (\frac{k_x \Delta x}{2}) \neq 0$, and hence it is unstable.

We do not write out the stability analysis for the schemes (8), (9) and (10). For all of them we can derive the structure (11). It is also possible to show that they are unconditionally stable in the $-t$ direction for a fixed z if $\alpha \leq \frac{1}{4}$. For (10) we may have a linear growth in n which is consistent with the differential equation (6).

The only properties we need for stability in the x differencing are the following: The Fourier transform of the approximation of ∂_x is pure imaginary and the transform of the approximation of ∂_{xx} is real and negative. Hence, it is easy to improve the approximation in the x direction. We can, for example, use the following operators (the superscript x is omitted):

$$\partial_x \sim D_0 - \frac{\Delta x^2}{6} D_+ D_- D_0 \qquad \text{(4th order)}$$

$$\partial_x \sim D_0 (I + \frac{\Delta x^2}{6} D_+ D_-)^{-1} \qquad \text{(4th order)}$$

$$\partial_x \sim D_0 - \frac{\Delta x^2}{6} D_+ D_- D_0 + \frac{\Delta x^4}{30} (D_+ D_-)^2 D_0 \qquad \text{(6th order)}$$

$$\partial_x \sim D_0 (I + \frac{\Delta x^2}{6} D_+ D_- - \frac{\Delta x^4}{180} D_+ D_-)^{-1} \qquad \text{(6th order)}$$

$$\partial_{xx} \sim D_+ D_- - \frac{\Delta x^2}{12} (D_+ D_-)^2 \qquad \text{(4th order)}$$

$$\partial_{xx} \sim D_+ D_- (I + \frac{\Delta x^2}{12} D_+ D_-)^{-1} \qquad \text{(4th order)}$$

$$\partial_{xx} \sim D_+ D_- - \frac{\Delta x^2}{12} (D_+ D_-)^2 + \frac{\Delta x^4}{90} (D_+ D_-)^3 \qquad \text{(6th order)}$$

$$\partial_{xx} \quad D_+ D_- (I + \frac{\Delta x^2}{12} D_+ D_- - \frac{\Delta x^4}{240} (D_+ D_-)^2)^{-1} \qquad \text{(6th order)}$$

We can also try to derive approximations where the truncation error compensates for the error we get when approximating the wave equation with (5) or (6).

We have tested four subroutines SM7, SM8, SM9 and SM10 corresponding to the methods (7), (8), (9) and (10). The code for the algorithms are given on the last pages of this paper.

We checked the stability of these routines by using random initial data. The L_2-norm was conserved as the theory indicates.

In the following tables we compared the approximations with plane wave solutions. This was done partly to check the programs and partly to study the effeciency of the schemes. It is otherwise better to analyze the accuracy in the transformed space when approximating plane waves. The values preserved in Table 1 and 2 are for a special choice of Δt, Δx, Δz, θ and $\frac{k'}{\omega'}$ and can only be used as a complement when judging the quality of the methods.

In the test run we had $\theta = 30$, $\Delta x = \Delta t = 1$, $\Delta z = 0.2$ and $v = 1$. The number of points per wavelength (NPW) was 6 and 12, and the number of steps in z was respectively 10 and 20. The initial values were sin waves and the analytic values were given at the boundaries. We had $\alpha = \frac{1}{12}$ in SM9 and SM10. The norm of the initial values was one.

We can see in Table 1 that the convergence from NPW = 6 to 12 is about a factor of 4. That is what we expect from second order methods. The scheme SM9, which treats several terms with higher accuracy, has the best performance. The truncation error in time for SM7 is four times that of SM8 and so is the error we can see here. On the other hand, since (7) is explicit it is faster and simpler to use.

TABLE 1

$\frac{k'_x}{\omega'} =$	sin 5°		sin 20°	
NPW =	6	12	6	12
SM7	.228	.069	3.75	1.42
SM8, $\alpha=0$	.057	.018	.932	.269
SM8, $\alpha=\frac{1}{12}$	.056	.011	.862	.237
SM8, $\alpha=\frac{1}{4}$	.058	.011	.709	.184
SM9	.009	.003	.266	.039
SM10	.114	.038	1.93	.722

L_2-error (times 100) when comparing the approximations with the analytic solutions of the corresponding differential equations (5) or (6).

TABLE 2

$\frac{k'_x}{\omega'} =$	sin 5°		sin 20°	
NPW =	6	12	6	12
SM7	.248	.093	6.14	4.07
SM8, $\alpha = 0$	.077	.041	3.13	2.77
SM8, $\alpha=\frac{1}{12}$	.079	.034	3.10	2.75
SM8, $\alpha = \frac{1}{4}$	.077	.035	3.03	2.71
SM9	.026	.020	2.62	2.68
SM10	.131	.051	3.15	2.19

L_2-error (times 100) when comparing the approximations with the solution to the scalar wave equation.

Table 2 shows that the main error for NPW = 6 and 5° comes from the discretization of (5) and (6). For NPW = 12 and 20° it is the other way around. The error originates from the difference between (5) or (6) and the wave equation. There is almost no improvement from SM8 to SM9 . Here the third order differential equation (6), used in SM10 starts to pay off.

In the final sequence of tests we propagated a wave form (Fig. 1) to the surface. In Fig. 2, 6 we used SM10 and in in Fig. 10, 15 the algorithm SM9. We then migrated these synthetic seismograms on twice as coarse a grid. This was done to get a more realistic situation where the truncation errors in the forward and backward problems did not cancel.

The computations were carried out in the slanted frome with $\theta = 20°$, $\alpha = \frac{1}{12}$. The grid sizes are displayed fairly well in the plots ($v \Delta t \Delta z / \Delta x^2 = 0.2$) . The initial values were products of sine waves in their area of support.

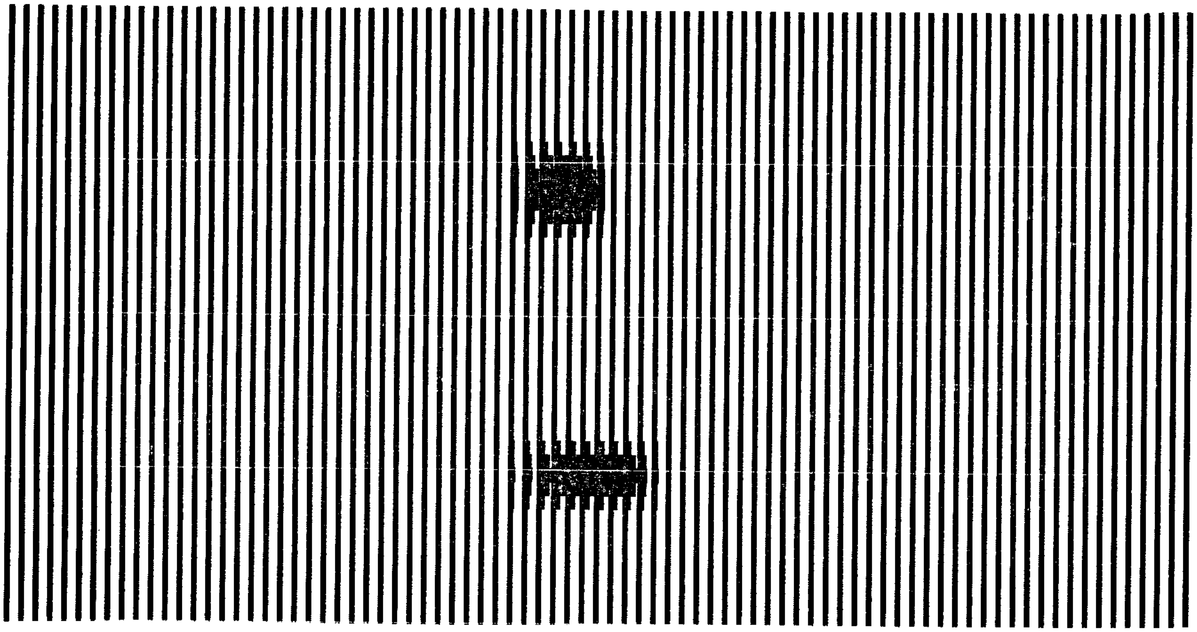


Fig. 1. Initial wave form.

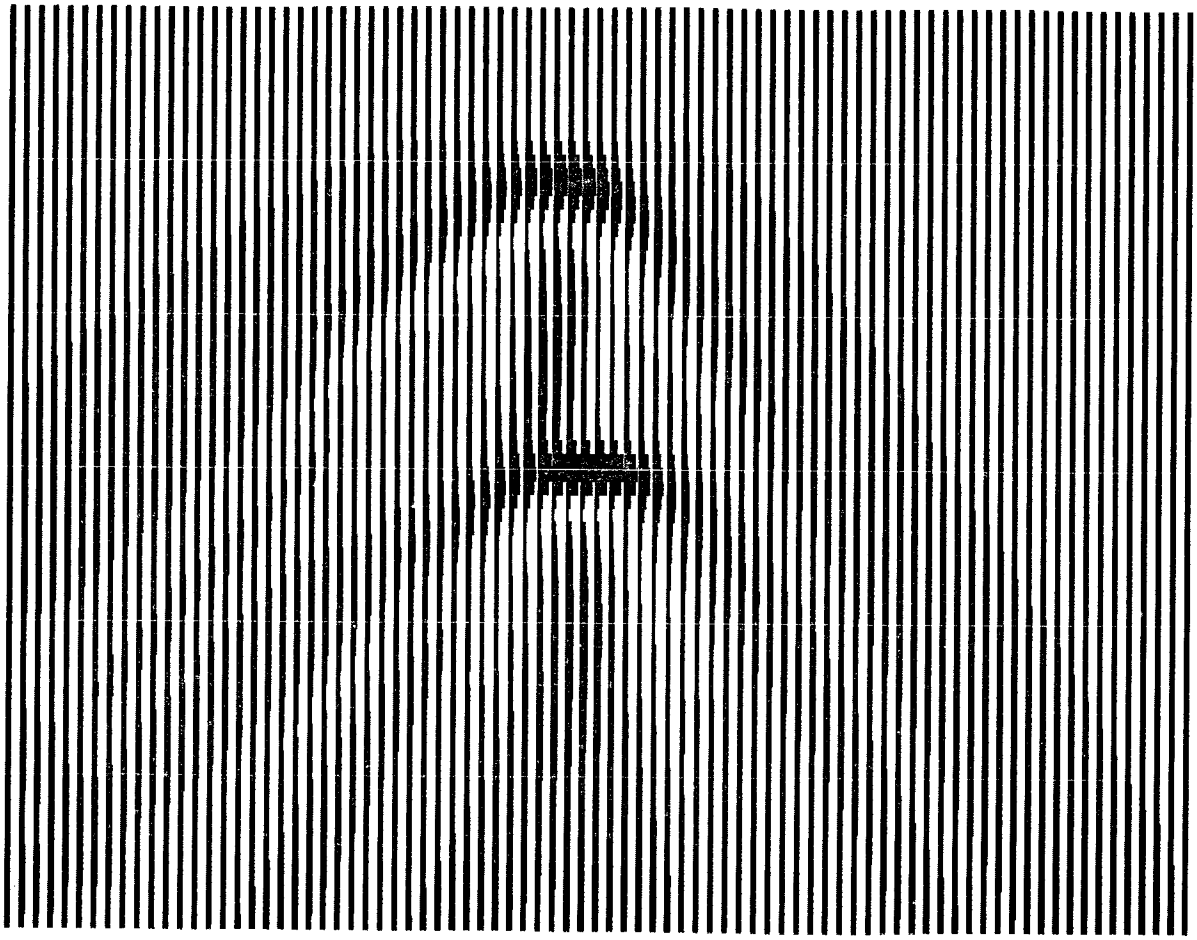


Fig. 2. SM10, surface data from $z = 20 \left(\frac{\Delta z}{2}\right)$.

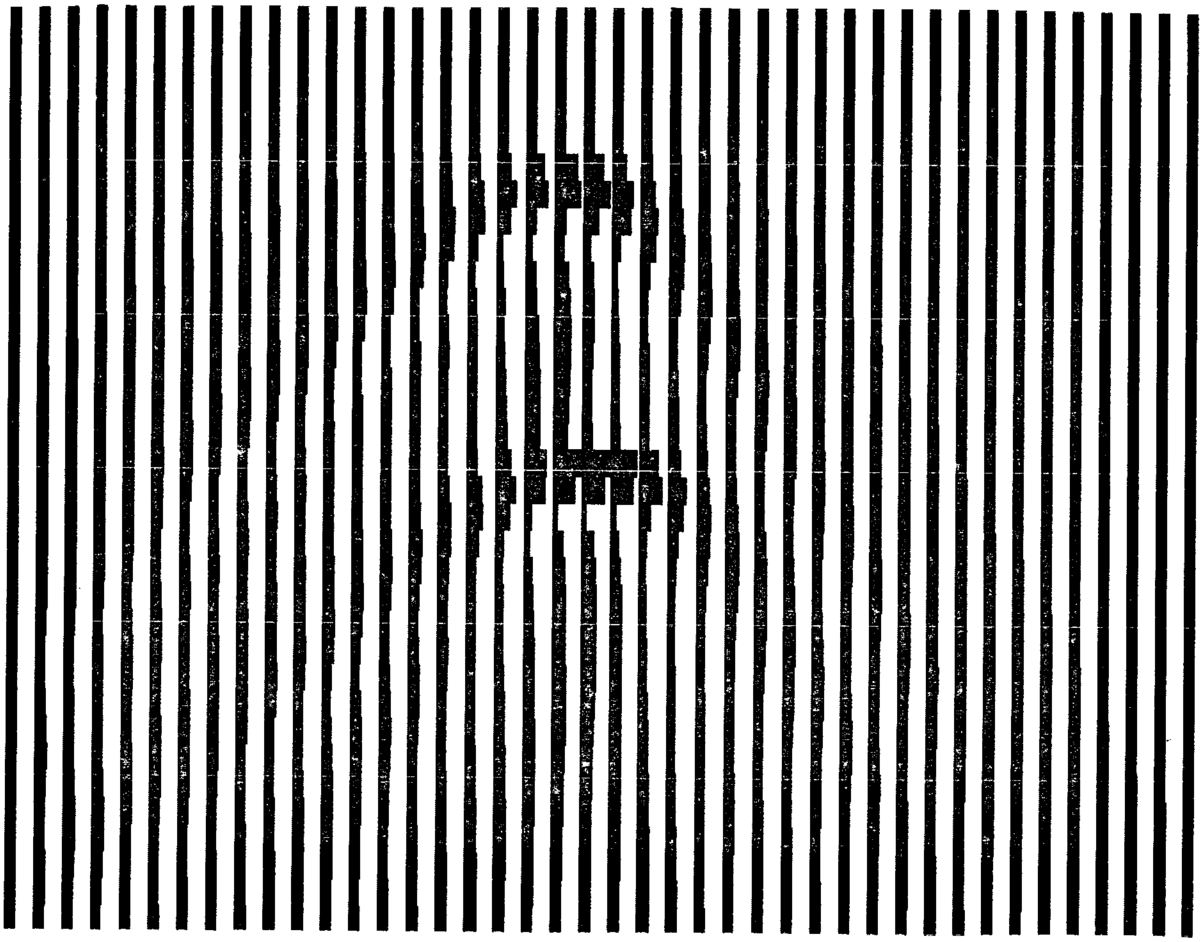


Fig. 3. Coarser sampling of data from Fig. 2.

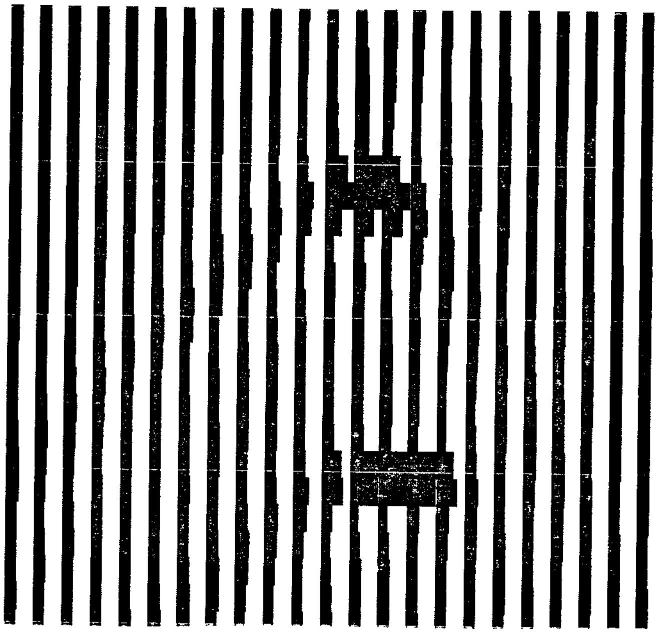


Fig. 4. SM10, migration to 10 Δz .

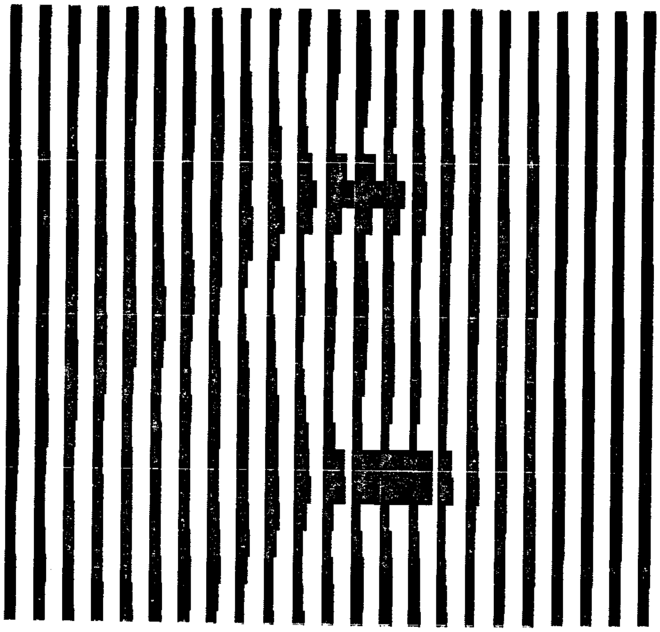


Fig. 5. SM10 (without P_{txz}-term), migration to 10 Δz .

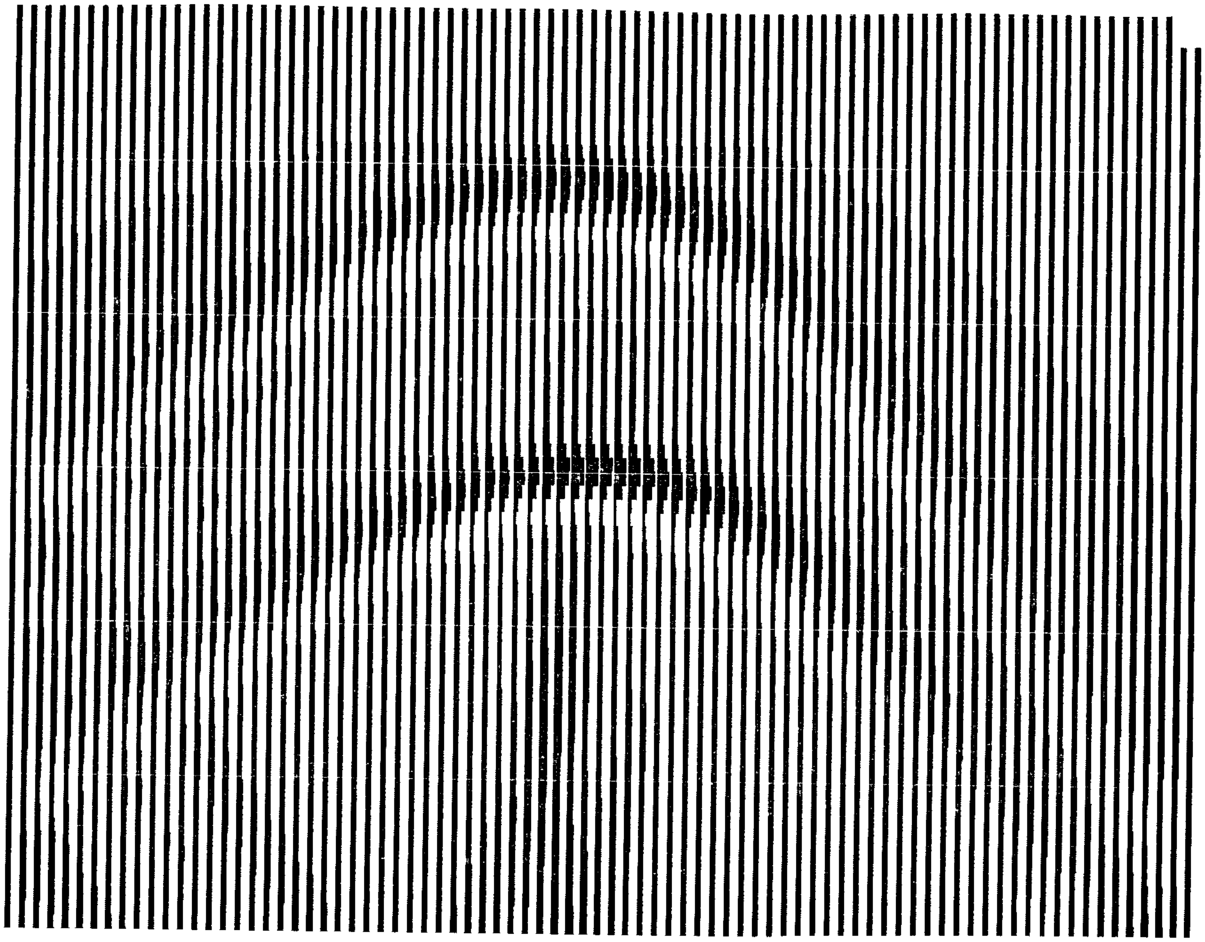


Fig. 6. SM10, surface data from $z = 80\ (\frac{\Delta z}{2})$.

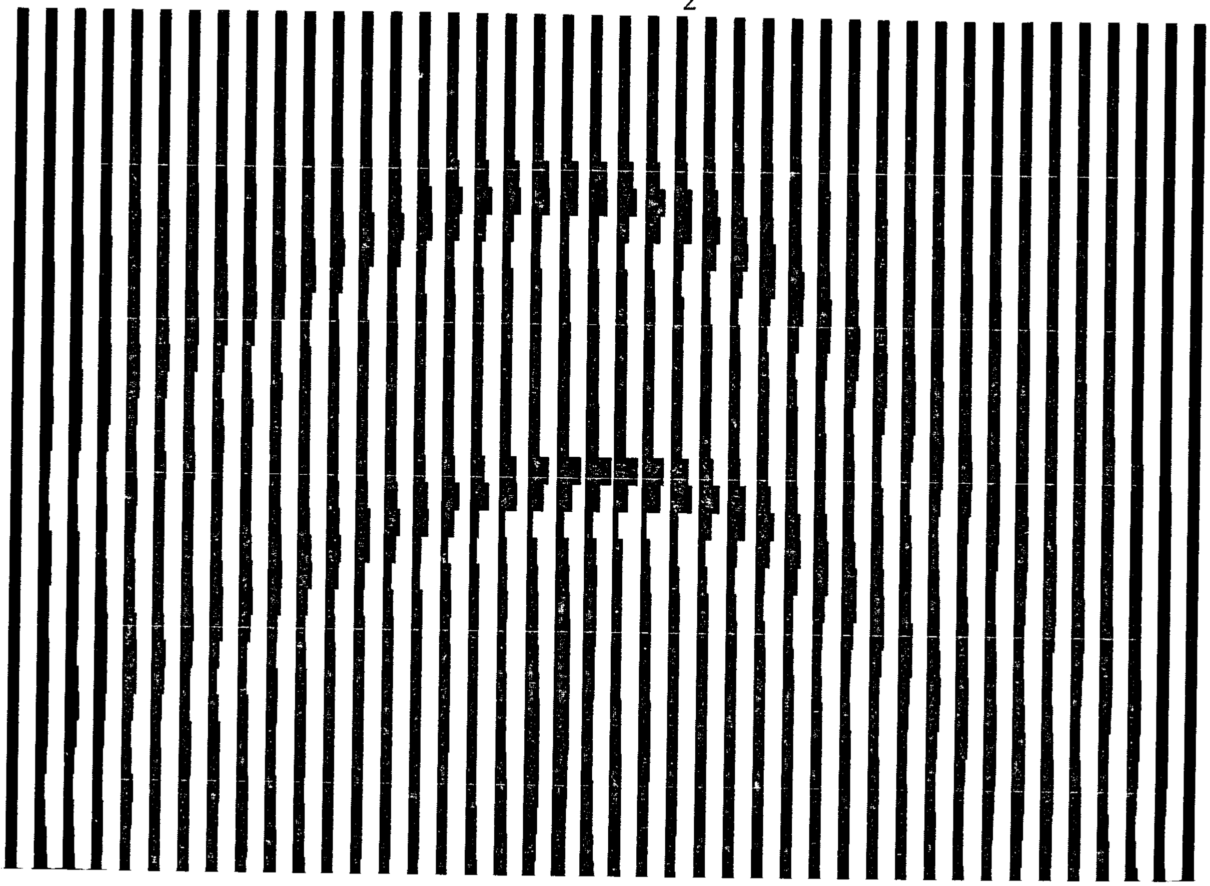


Fig. 7. Coarser sampling of data from Fig. 6.

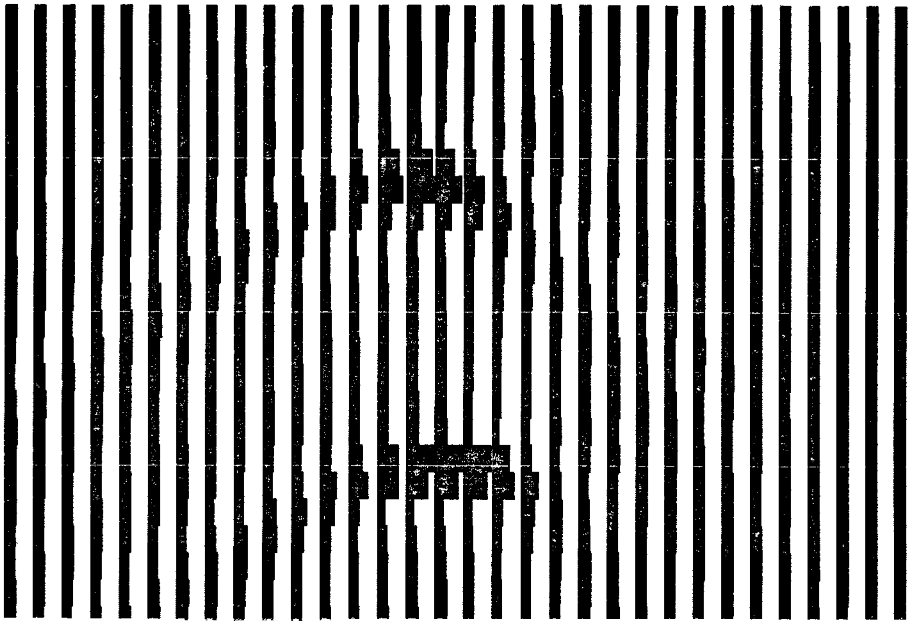


Fig. 8. SM10, migration to 40 Δz .

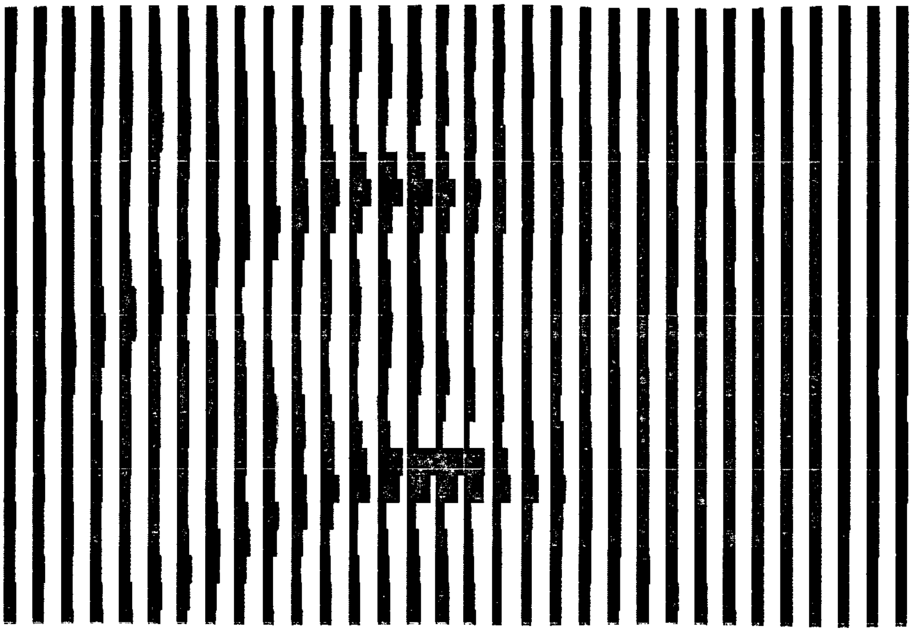


Fig. 9. SM10 (without P_{txz}-term), migration to 40 Δz .

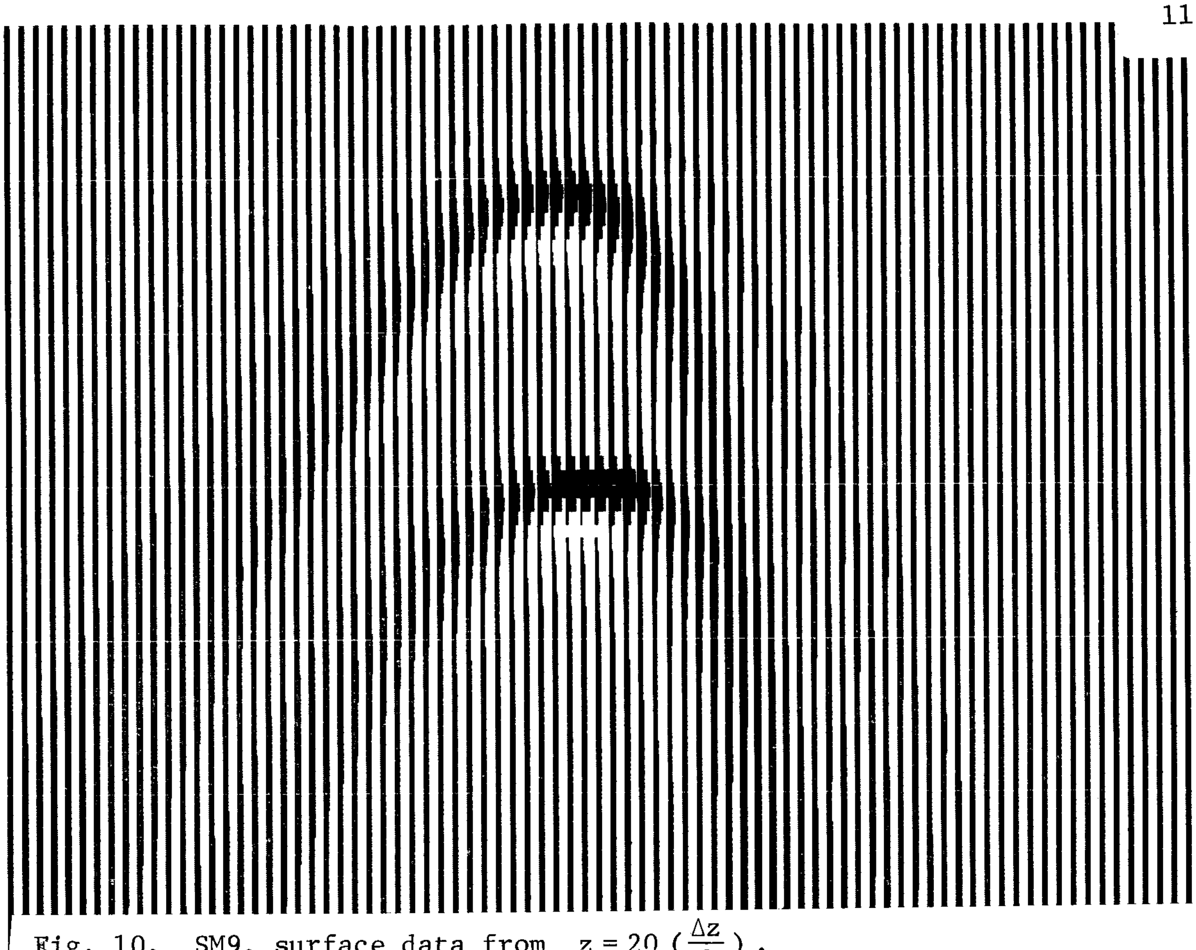


Fig. 10. SM9, surface data from $z = 20(\frac{\Delta z}{2})$.

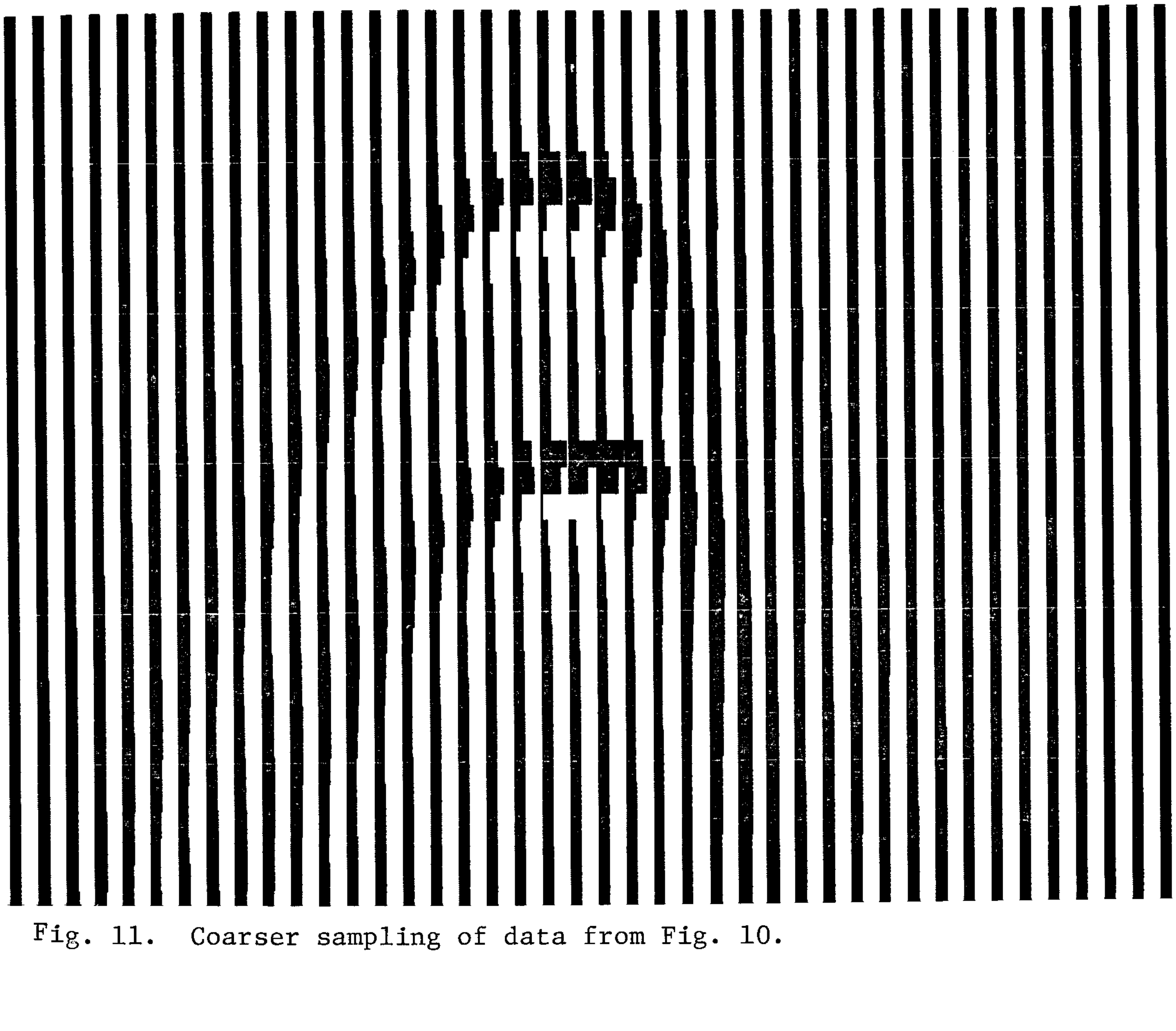


Fig. 11. Coarser sampling of data from Fig. 10.

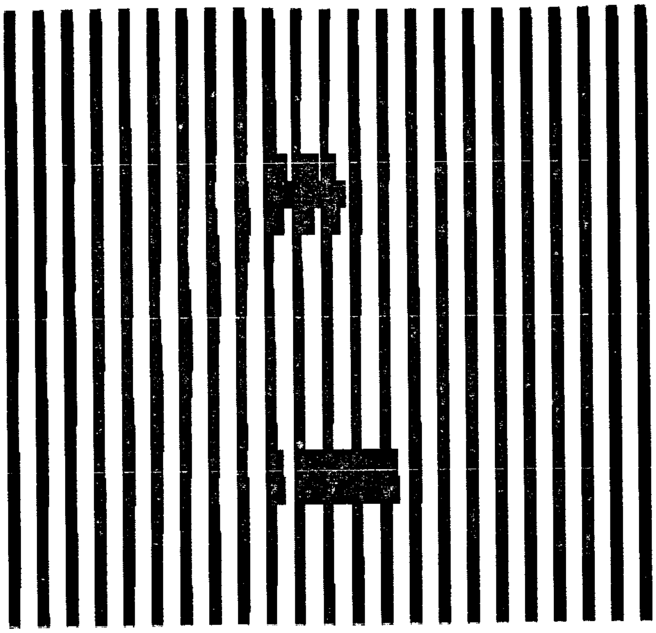


Fig. 12. SM9, migration to 10 Δz .

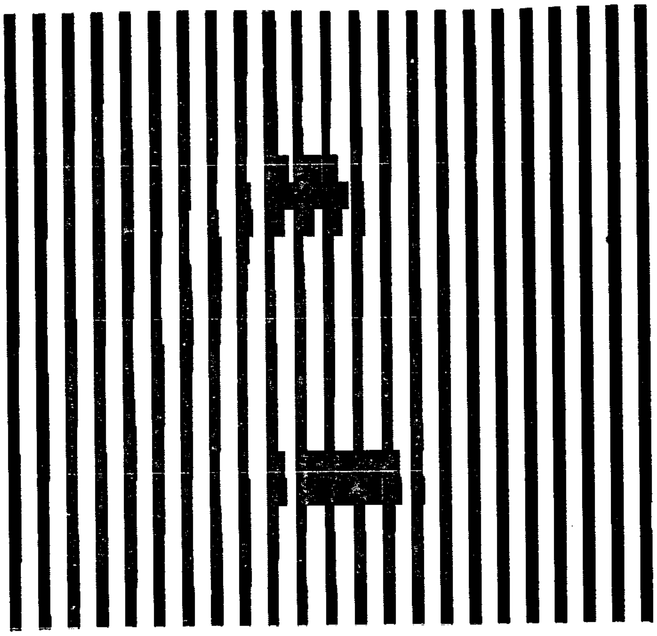


Fig. 13. SM8, migration to 10 Δz .

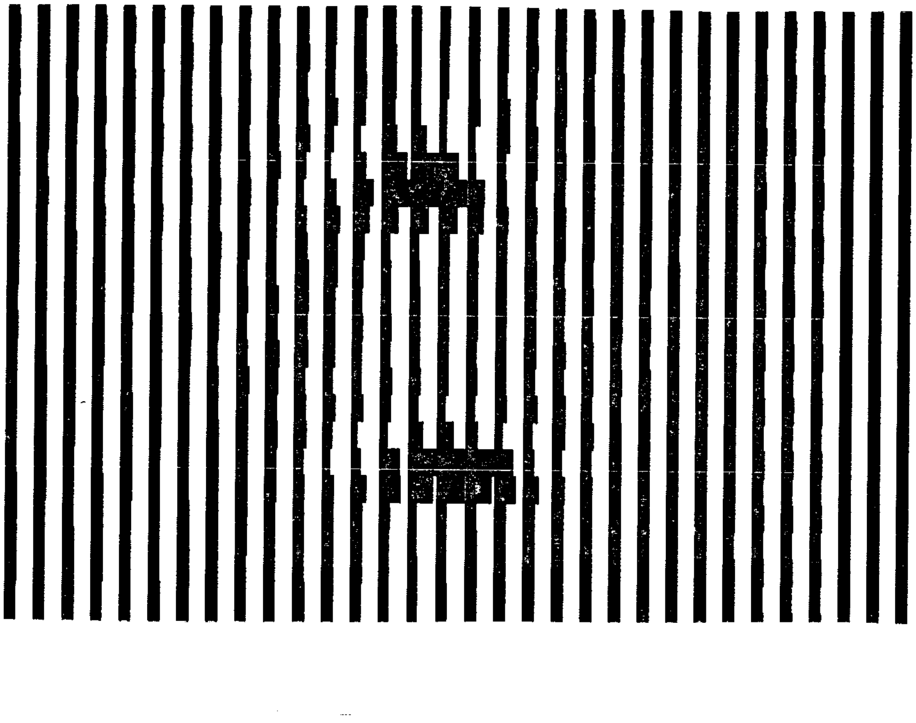


Fig. 14. SM7, migration to 10 Δz .

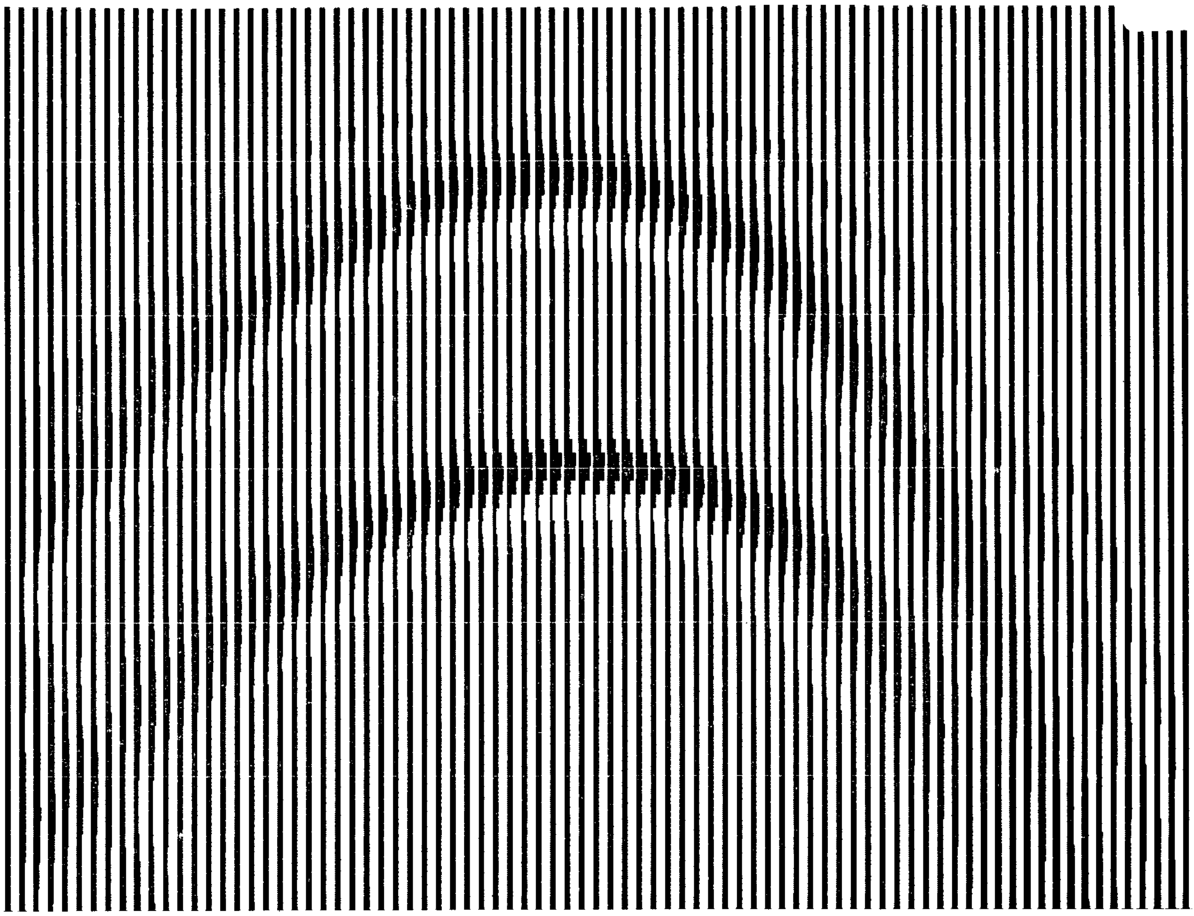


Fig. 15. SM9, surface data from $z = 80(\frac{\Delta z}{2})$.

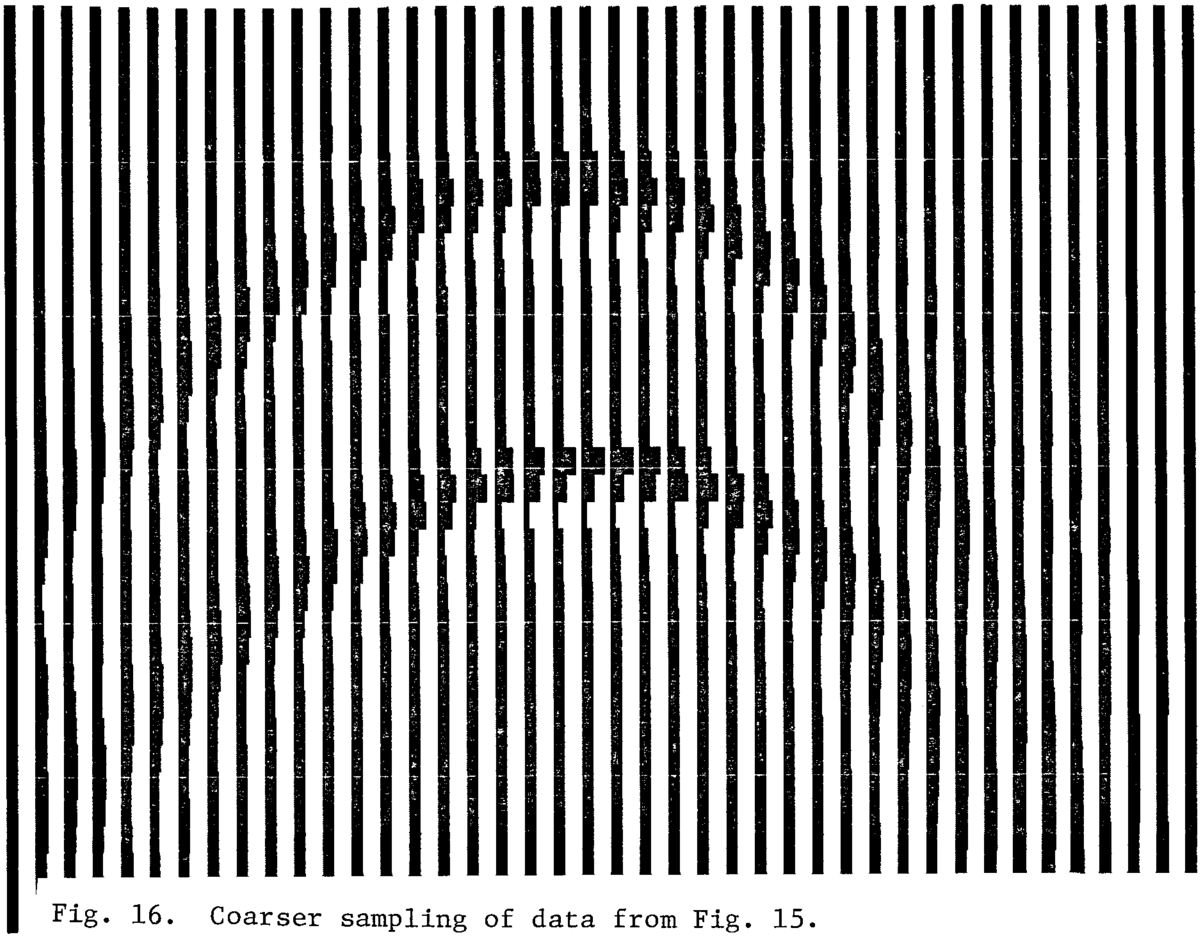


Fig. 16. Coarser sampling of data from Fig. 15.

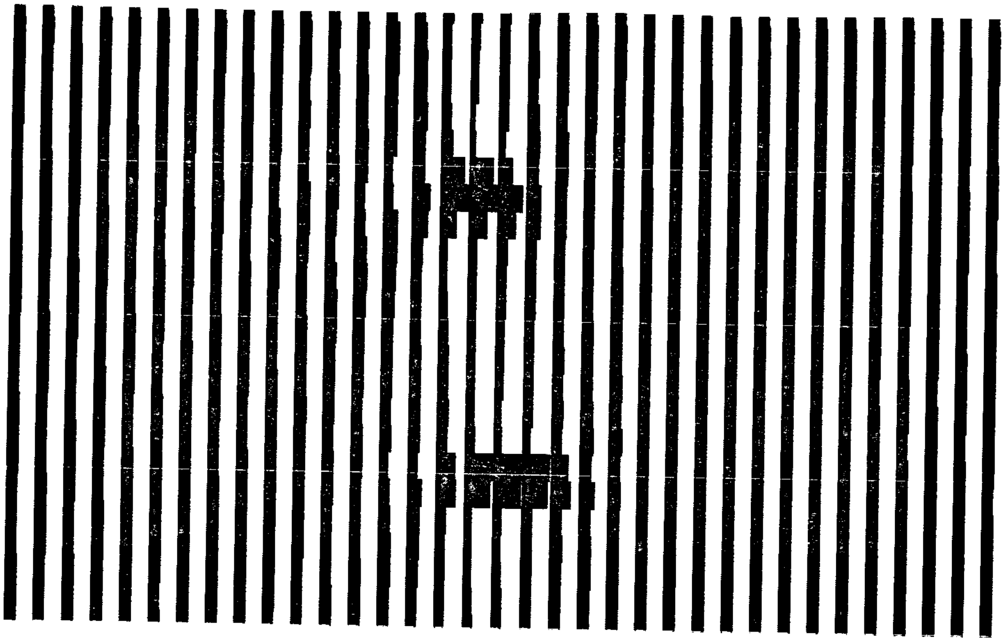


Fig. 17. SM9, migration to 40 Δz .

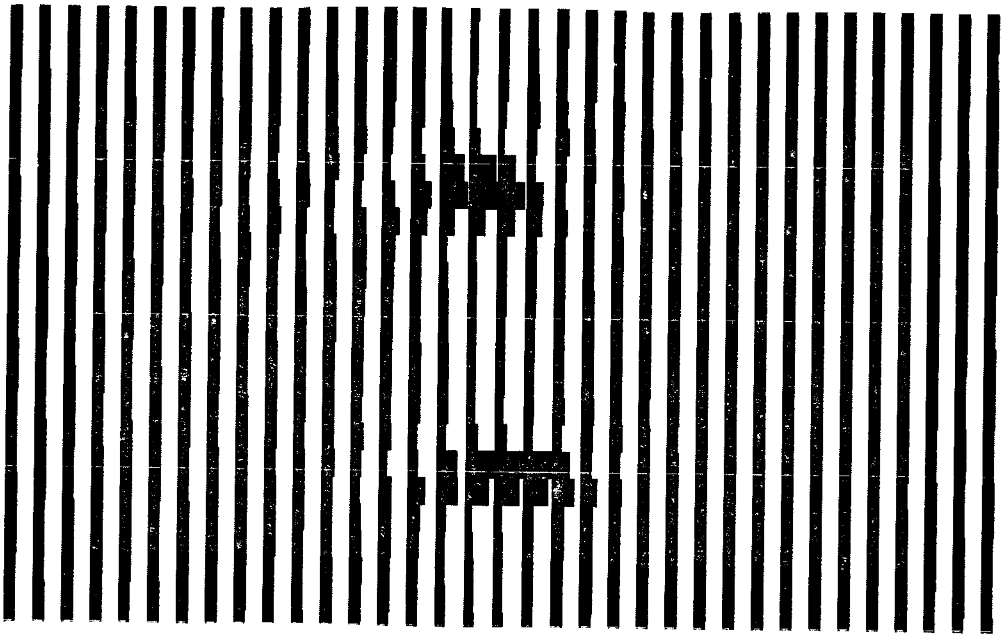


Fig. 18. SM8, migration to 40 Δz .

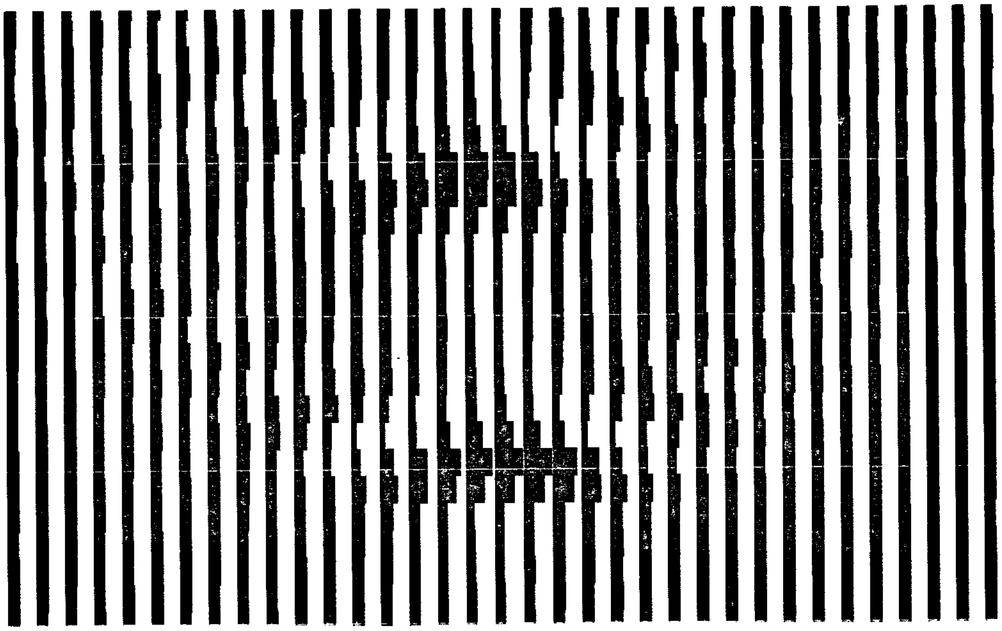


Fig. 19. SM7, migration to 40 Δz .

The schemes (8) and especially (9) did a good job in restoring the original picture. We can also see that the skew term P_{txz} in (10) had some positive effect. These plots mainly show the discretization error. It would have been comparatively less favorable to the methods based on the second order differential equation if the seismograms were produced by the wave equation.

```
      SUBROUTINE S47(N1, N2)
C***  SLANTED MIGRATION FROM Z-LEVEL N1 TO N2
C***  INITIAL VALUES IN P, BOUNDARY CONDITIONS IN BC
C***  SECOND ORDER PDE, IMPLICIT SCHEME
      COMMON NT,NX,NZ,DT,DX,DZ,P(64,64),Q(64,64),THETA
      N11 = N1+1
      NXS1 = NX-1
       CO = COS(THETA)
      A = -TAN(THETA)*DT/(2*CO*DX)
      B = DT*DZ/(2*CO**3*DX**2)
      DO 10 N=N11,N2
      DO 20 I=1,NT
      DO 20 J=1,NX
 20   Q(I,J) = P(I,J)
      DO 30 I=1,2
      DO 30 J=1,NX
 30   P(I,J) = BC(I,J,N)
      DO 40 I=3,NT
 40   P(I,1) = BC(I,1,N)
      DO 50 I=3,NT
 50   P(I,NX) = BC(I,NX,N)
      DO 60 I=3,NT
      DO 60 J=2,NXS1
 60   P(I,J) = A*(P(I-1,J+1)-P(I-1,J-1))+B*(P(I-2,J+1)+P(I-2,J-1))
     1   +(1.-2.*B)*P(I-2,J)+B*(Q(I,J+1)+Q(I,J-1))+(1.-2.*B)*Q(I,J)
     2   -A*(Q(I-1,J+1)-Q(I-1,J-1))-Q(I-2,J)
 10   CONTINUE
      RETURN
      END

      SUBROUTINE S13(N1, N2,D)
C***  SLANTED MIGRATION FROM Z-LEVEL N1 TO N2
C***  INITIAL VALUES IN P, BOUNDARY CONDITIONS IN BC
C***  D IS THE PARAMETER ALPHA
C***  SECOND ORDER PDE,IMPLICIT SCHEME
      COMMON NT,NX, NZ,DT,DX,DZ,P(64,64),Q(64,64),THETA
      DIMENSION X(64),Y(64)
      N11 = N1+1
      NXS2 = NX-2
      CO = COS(THETA)
      C1 = TAN(THETA)*DT/(4*CO*DX)
      C2 = DT*DZ/(3*CO**3*DX**2)
      A= -C1-C2*D
      B = 1.-2*D*2*C2
      C = C1-C2*D
      DO 10 N=N11,N2
      DO 20 I=1,NT
      DO 20 J=1,NX
 20   Q(I,J) = P(I,J)
      DO 30 J=1,NX
 30   P(1,J) = BC(1,J,N)
      DO 40 I=2,NT
      HL = BC(I,1,N)
      HR = BC(I,NX,N)
      DO 50 J=1,NXS2
      Y1 = (C2+C1+D)*P(I-1,J)+(1.-2*D-2*C2)*P(I-1,J+1)
      Y2 = (C2-C1+D)*P(I-1,J+2)
      Y3 = (C2-C1+D)*Q(I,J)+(1.-2*D-2*C2)*Q(I,J+1)
      Y4 = (C2+C1+D)*Q(I,J+2)
      Y5 = (C2-C1-D)*Q(I-1,J)-(1.-2*D+2*C2)*Q(I-1,J+1)
      Y6 = (C2+C1-D)*Q(I-1,J+2)
      Y(J) = Y1+Y2+Y3+Y4+Y5+Y6
50    CONTINUE
      Y(1) = Y(1)-A*HL
      Y(NXS2) = Y(NXS2)-C*HR
      CALL TRID(X,Y,NXS2,A,B,C)
      DO 60 J=1,NXS2
60    P(I,J+1) = X(J)
      P(I,1) = HL
      P(I,NX) = HR
40    CONTINUE
10    CONTINUE
      RETURN
      END

      SUBROUTINE SLO(N1,N2)
C***  SLANTED ITERATION FROM Z-LEVEL N1 TO N2
C***  INITIAL VALUES IN P, BOUNDARY CONDITIONS IN BC
C***  SECOND ORDER PDE,IMPLICIT SCHEME
      COMMON NT,NX,NZ,DT,DX,DZ,P(64,64),Q(64,64),THETA
      DIMENSION X(64),Y(64)
      N11 = N1+1
      NTS1 = NT-1
      NXS2 = NX-2
      CO = COS(THETA)
      C1 = TAN(THETA)*DT/(2*CO*DX)
      C2 = DT*DZ/(2*CO**3*DX**2)
      C3 = 7*C2/6
      C4 = C2/6
      D = (2.+DT*DZ/DX**2)/24
      A= -C1-C3+D
      B = 1.-2*D+2*C3
      C = C1-C3+D
      DO 10 N=N11,N2
      DO 20 I=1,NT
      DO 20 J=1,NX
20    Q(I,J) = P(I,J)
      DO 30 J=1,NX
      P(1,J) = BC(1,J,N)
30    P(2,J) = BC(2,J,N)
      DO 40 I=3,NTS1
      HL = BC(I,1,N)
      HR = BC(I,NX,N)
      DO 50 J=1,NXS2
      Y1 = (C2+C1+D)*P(I-1,J)+(1.-2*D-2*C2)*P(I-1,J+1)
      Y2 = (C2-C1+D)*P(I-1,J+2)
      Y3 = (C2-C1+D)*Q(I,J)+(1.-2*D-2*C2)*Q(I,J+1)
      Y4 = (C2+C1+D)*Q(I,J+2)
      Y5 = (C3-C1-D)*Q(I-1,J)-(1.-2*D+2*C3)*Q(I-1,J+1)
      Y6 = (C3+C1-D)*Q(I-1,J+2)
      Y7 = -C4*(P(I-2,J)-2*P(I-2,J+1)+P(I-2,J+2))
      Y8 = -C4*(Q(I+1,J)-2*Q(I+1,J+1)+Q(I+1,J+2))
      Y(J) = Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8
50    CONTINUE
      Y(1) = Y(1)-A*UL
      Y(NXS2) = Y(NXS2)-C*UR
      CALL TRID(X,Y,NXS2,A,B,C)
      DO 60 J=1,NXS2
60    P(I,J+1) = X(J)
      P(I,1) = UL
      P(I,NX) = UR
40    CONTINUE
      DO 45 J=1,NX
45    P(NT,J) = BC(NT,J,N)
10    CONTINUE
      RETURN
      END

      SUBROUTINE S3IC(N1,N2)
C***  SLANTED ITERATION FOR 2-LEVEL N1 TO N2
C***  INITIAL VALUES IN P, BOUNDARY CONDITIONS IN BC
C***  THIRD ORDER PDE,IMPLICIT SCHEME
      COMMON NT,NX,NZ,DT,DX,DZ,P(64,64),Q(64,64),THETA
      DIMENSION X(64),Y(64)
      N11 = N1+1
      NXS2 = NX-2
      CO = COS(THETA)
      C1 = TAN(THETA)*DT/(4*CO*DX)
      C2 = DT*DZ/(8*CO**3*DX**2)
      C3 = DT**2/(8*CO**2*DX**2)
      A = -C1-(C3+C2)
      B = 1.+2*(C3+C2)
      C = C1-(C3+C2)
      DO 10 N=N11,N2
      DO 20 I=1,NT
      DO 20 J=1,NX
20    Q(I,J) = P(I,J)
      DO 30 J=1,NX
      P(1,J) = BC(1,J,N)
30    P(2,J) = BC(2,J,N)
      DO 40 I=3,NT
      UL = BC(I,1,N)
      UR = BC(I,NX,N)
      DO 50 J=1,NXS2
      Y1 = (-C1+C3-C2)*P(I-2,J)+(-1-2*C3+2*C2)*P(I-2,J+1)
      Y2 = (C1+C3-C2)*P(I-2,J+2)
      Y4 = (-C1-C3+C2)*P(I,J)+(1+2*C3-2*C2)*P(I,J+1)
      Y5 = (C1-C3+C2)*P(I,J+2)
      Y6 = (C1-C3-C2)*P(I-2,J)+(1+2*C3+2*C2)*P(I-2,J+1)
      Y7 = (-C1-C3-C2)*P(I-2,J+2)
      Y(J) = Y1+Y2    +Y4+Y5+Y6+Y7+2.*P(I-1,J+1)-2*P(I-1,J+1)
 50   CONTINUE
      Y(1) = Y(1)-A*PL
      Y(NXS2) = Y(NXS2)-C*PR
      CALL TRID(X,Y,NXS2,A,B,C)
      DO 60 J=1,NXS2
 60   P(I,J+1) = X(J)
      P(I,1) = PL
      P(I,NX) = PR
 40   CONTINUE
 10   CONTINUE
      RETURN
      END

      SUBROUTINE TRID(X,Y,N,A,B,C)
C***  TRIDIAGONAL SYSTEM SOLVER, DIAGONAL ELEMENTS IN A, B AND C
      DIMENSION X(N),Y(N),Z(100),W(100)
      DIAG = B
      Z(1) = Y(1)/B
      NS1 = N-1
      DO 10 I=1,NS1
      W(I) = C/DIAG
      DIAG = B-A*W(I)
 10   Z(I+1) = (Y(I+1)-A*Z(I))/DIAG
      X(N) = Z(N)
      DO 20 I=1,NS1
      J = N-I
 20   X(J) = Z(J)-W(J)*X(J+1)
      RETURN
      END
```