*Geophysical Prospecting* **37**, 925–958, 1989

# 3D TABLE-DRIVEN MIGRATION[1]

G. BLACQUIÈRE[2], H. W. J. DEBEYE[3],
C. P. A. WAPENAAR[2] and A. J. BERKHOUT[2]

## ABSTRACT

BLACQUIÈRE, G., DEBEYE, H.W.J., WAPENAAR, C.P.A. and BERKHOUT, A.J. 1989. 3D table-driven migration. *Geophysical Prospecting* **37**, 925–958.

An efficient full 3D wavefield extrapolation technique is presented. The method can be used for any type of subsurface structure and the degree of accuracy and dip-angle performance are user-defined. The extrapolation is performed in the space–frequency domain as a space-dependent spatial convolution with recursive Kirchhoff extrapolation operators.

To get a high level of efficiency the operators are optimized such that they have the smallest possible size for a specified accuracy and dip-angle performance. As both accuracy and maximum dip-angle are input parameters for the operator calculation, the method offers the possibility of a trade-off between these quantities and efficiency. The operators are calculated in advance and stored in a table for a range of wavenumbers. Once they have been calculated they can be used many times.

At the basis of the operator design is the well-known phase-shift operator. Although this operator is exact for homogeneous media only, it is assumed that it may be applied locally in case of inhomogeneities. Lateral velocity variations can then be handled by choosing the extrapolation operator according to the local value of the velocity. Optionally the operators can be designed such that they act as spatially variant high-cut filters. This means that the evanescent field can be suppressed in one pass with the extrapolation. The extrapolation method can be used both in prestack and post-stack applications. In this paper we use it in zero-offset migration. Tests on 2D and 3D synthetic and 2D real data show the excellent quality of the method. The full 3D result is much better then the result of two-pass migration, which has been applied to the same data.

The implementation yields a code that is fully vectorizable, which makes the method very suitable for vector computers.

## INTRODUCTION

Many current 3D migration techniques, e.g. 3D time migration methods in general, two-pass (or 2 times 2D) depth migration and 3D depth migration based on oper-

[1] Received July 1988, revision accepted April 1989.

[2] Delft University of Technology, Faculty of Applied Physics, Group of Seismics and Acoustics, P.O. Box 5046, 2600 GA Delft, The Netherlands.

[3] Delft University of Technology, Faculty of Applied Physics, Group of Seismics and Acoustics. Present address: Jason Geosystems B.V., P.O. Box 596, 2600 AN Delft, The Netherlands.

ator splitting, are only approximately valid in case of inhomogeneous media. If strong velocity variations occur in the subsurface, a one-pass, true 3D, depth migration method should be used (Yilmaz *et al.* 1987). The quality of such a method is determined by the wavefield extrapolation, apart from the quality of the data and the availability of a good macro-model. To be able to handle the lateral velocity variations correctly, the extrapolation is performed as a laterally variant convolution in the space–frequency $x$, $y$, $\omega$ domain (Berkhout and Van Wulfften Palthe 1979; Berkhout 1984, 1985). Vertical velocity variations are taken into account by dividing the model into small layers and by applying the extrapolation recursively from one depth level to the next.

In practice some of the problems that have to be avoided in wavefield extrapolation are: instability, inaccuracy, poor performance if steep dips are present, and spatial operator aliasing. In this paper we describe a 3D table-driven extrapolation scheme in which these problems have been solved. We use a 3D version of the optimum operator design method as introduced by Holberg (1988) where full control of accuracy and dip-angle performance is available, and it also offers the option of space-dependent high-angle and evanescent field suppression. The advantage is that whatever the requirements are, the result is always the smallest possible stable extrapolation operator.

Full 3D wavefield extrapolation is numerically expensive, and therefore special attention must be paid to an efficient implementation of the scheme. Following an idea of Raoult and Dezard (1985), we calculate the 3D extrapolation operators in advance and store them in a table. Some aspects concerning the table size (the number of operators) will be discussed. Also, the symmetry properties of the operators – all information is actually contained in one quadrant of the 3D operator – are used to reduce the number of calculations. We use this wavefield extrapolation method in a recursive way together with the imaging principle for the development of a 3D zero-offset migration scheme.

The final section is dedicated to examples. Although we only discuss the 3D case in this paper, the method can be easily applied for the 2D case as well, and therefore we also give some 2D examples. The algorithm was tested on synthetic data from subsurface models with strong velocity variations. A 2D zero-offset section was modelled with an accurate finite-difference scheme in which, for example, diffraction energy is fully incorporated (Kelly, Seford and Whitmore 1982); we also show a 2D example of real data. The 3D examples were generated using optimized extrapolation operators both for the modelling and the migration. For comparison a result of two-pass migration is also presented.

## Wavefield Extrapolation Theory

### *Introduction*

3D downward wavefield extrapolation, from depth level $z_i$ to depth level $z_{i+1} = z_i + \Delta z$, in the $x$, $y$, $\omega$ domain can be formulated as

$$P(x, y, z_{i+1}, \omega) = F[x, y, \Delta z, k(x, y, z_i, \omega)] * P(x, y, z_i, \omega), \tag{1}$$

with

$$k(x, y, z_i, \omega) = \omega/c(x, y, z_i),$$

where $P$ denotes the temporally Fourier transformed upgoing pressure field, $F$ denotes the inverse wavefield extrapolation operator, $k$ is the local wavenumber, $c$ is the propagation velocity of the medium, and the symbol $*$ denotes a two-dimensional space-dependent spatial convolution along the $x$- and $y$-coordinates.

The extrapolation according to (1) should be applied recursively for all depth levels of interest. This way vertical velocity variations can be taken into account.

As can be seen from (1), a model of the subsurface containing the propagation velocity $c(x, y, z)$ is required. Following Berkhout (1985), we distinguish between the propagation properties and the reflectivity properties of the subsurface. The propagation depends on the macro seismic parameters of the medium, such as average velocity and average absorption. The reflectivity depends on the detailed seismic parameters of the medium such as local variations in velocity and density (see Fig. 1).

Because one-way wavefield extrapolation according to (1) compensates for the propagation effects of the medium, only the macro seismic parameters need to be known. A discussion on techniques to estimate or update macro models (Van der Made 1988) is beyond the scope of this paper.

Apart from the input data and the macro model that is available, the quality of the extrapolation is mainly determined by the quality of the recursive operator $F$. The number of arithmetic operations involved in the extrapolation is inversely proportional to the size of operator $F$. Therefore, in the following we will concentrate on the quality and the size of operator $F$.

### *Optimization of the downward extrapolation operator*

The operator optimization method, introduced by Holberg (1988) for the 2D

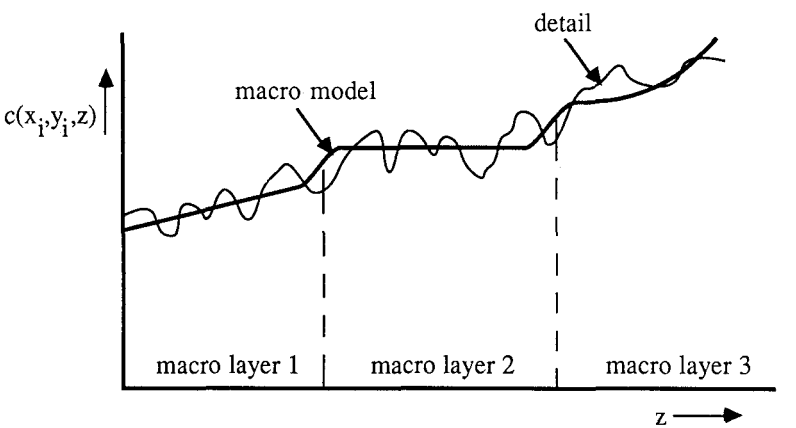


FIG. 1. Velocity profile of the macro model (thick line) and the detailed model (thin line). Given the macro-model and the seismic data, the detailed model can be obtained with seismic migration.

case, has some properties that make it very suitable for our purpose: (1) The operator size (number of operator points) and the maximum dip-angle are user-defined input parameters. (2) Optionally the operator can be designed such that it acts as a high-dip angle or evanescent field suppression filter. (3) The most accurate operator possible under the conditions in 1 and 2 will be found.

Hence, with the optimization method it is possible to find the operator that gives the most accurate results given the operator size, the maximum dip-angle and, optionally, a filter characteristic. However, one must keep in mind that our aim is to find the smallest operator given a criterion on the required accuracy, and again the maximum dip-angle and filter characteristic. This can be accomplished by iteratively applying the optimization method (see the flow-chart in Fig. 2). A description of the optimization method, which we have extended to the 3D case, will be given next.

The basic principle of the operator optimization is to minimize the phase and amplitude errors with a least-squares optimization algorithm. The option to make the operator also act as a high dip-angle and evanescent field suppressing filter can be included by defining a so-called constraint function, e.g. a cosine shaped filter.

The expression for wavefield extrapolation in the $x$, $y$, $\omega$ domain, equation (1),

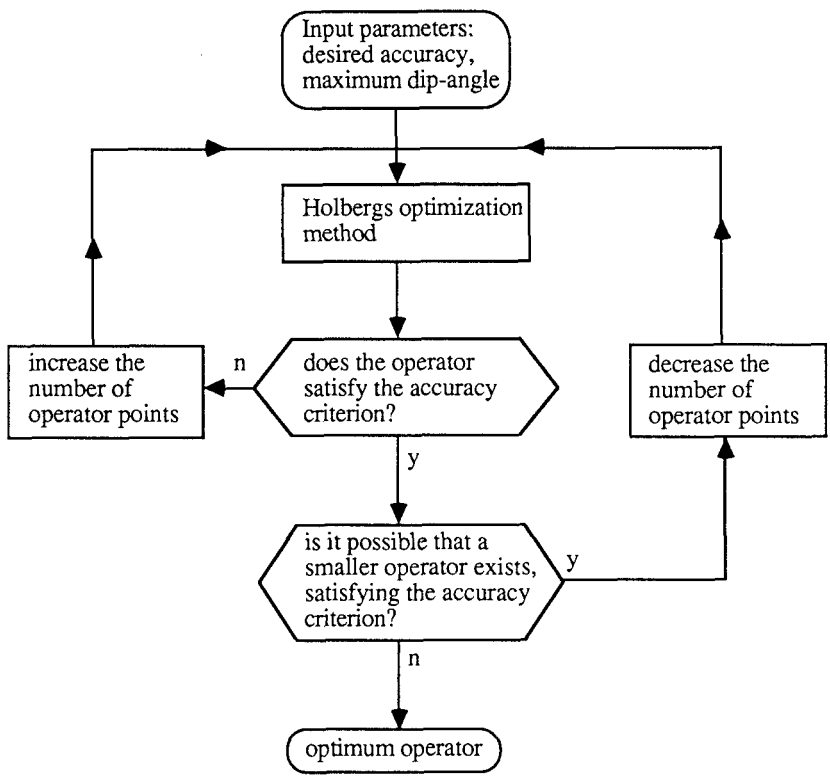


FIG. 2. Iterative application of Holbergs' operator optimization method, results in the smallest possible operator given the desired accuracy and maximum dip-angle.

can be written as

$$P(x, y, z_{i+1}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F[x', y', \Delta z, k(x, y, z_i, \omega)] \times P(x - x', y - y', z_i, \omega)\mathrm{d}x'\mathrm{d}y', \quad (2a)$$

or, after discretization in the $x$- and $y$-direction:

$$P(q\Delta x, r\Delta y, z_{i+1}, \omega) \approx \sum_{m=-M}^{M} \sum_{n=-N}^{N} \langle F[m\Delta x, n\Delta y, \Delta z, k(q\Delta x, r\Delta y, z_i, \omega)]\rangle \times P[(q - m)\Delta x, (r - n)\Delta y, z_i, \omega]\Delta x\Delta y, \quad (2b)$$

where $q = 1$, numx, and $r = 1$, numy. numx denotes the number of traces in the $x$-direction and numy denotes the number of traces in the $y$-direction.

In principle both $M$ and $N$ should be infinite. However, in practice $M$ and $N$ must be limited because they determine the size of the operator. Moreover, for efficiency, they should be as small as possible. Therefore, this expression is an approximation. Note that in (2b) we replaced the symbol $F$ for the extrapolation operator by $\langle F\rangle$, indicating that this is the approximated operator that has to be optimized.

We assume for the moment that the velocity is laterally invariant. In that case $k$ is constant within one extrapolation step, $k = k(z_i, \omega)$, and spatial convolution (2a) may be replaced by a simple multiplication in the $k_x$, $k_y$, $\omega$ domain, according to

$$\tilde{P}(k_x, k_y, z_{i+1}, \omega) = \tilde{F}(k_x, k_y, \Delta z, k)\, \tilde{P}(k_x, k_y, z_i, \omega). \quad (3)$$

Here the tilde ($\tilde{}$) denotes a 2D spatial Fourier transform, according to

$$\tilde{A}(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(x, y) \exp(\mathrm{j}k_x x) \exp(\mathrm{j}k_y y)\mathrm{d}x\mathrm{d}y,$$

or, after discretization in the $x$- and $y$-direction,

$$\tilde{A}(k_x, k_y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} A(m\Delta x, n\Delta y) \exp(\mathrm{j}k_x m\Delta x) \exp(\mathrm{j}k_y n\Delta y)\Delta x\Delta y.$$

We can now formulate the discretized and approximated version of (3), yielding the 2D spatial Fourier transform of (2b):

$$\tilde{P}(k_x, k_y, z_{i+1}, \omega) \approx \langle \tilde{F}(k_x, k_y, \Delta z, k)\rangle \tilde{P}(k_x, k_y, z_i, \omega), \quad (4)$$

with

$$\langle \tilde{F}(k_x, k_y, \Delta z, k)\rangle = \sum_{m=-M}^{M} \sum_{n=-N}^{N} f_{mn} \exp(\mathrm{j}k_x m\Delta x) \exp(\mathrm{j}k_y n\Delta y)\Delta x\Delta y.$$

Here the complex coefficients $f_{mn}$ are defined as

$$f_{mn} = \langle F(m\Delta x, n\Delta y, \Delta z, k)\rangle.$$

If we use the symmetry properties of the operator:

$$f_{mn} = f_{m(-n)},$$
$$f_{mn} = f_{(-m)n}, \tag{5}$$

we can write

$$\langle \tilde{F}(k_x, k_y, \Delta z, k) \rangle = \sum_{m=0}^{M} \sum_{n=0}^{N} 4 s_{mn} f_{mn} \cos(k_x m \Delta x) \cos(k_y n \Delta y) \Delta x \Delta y, \tag{6}$$

in which $s_{mn}$ is a symmetry factor defined as

$$\begin{aligned} s_{mn} &= 1 && \text{for} && m \geq 1 && \text{and} && n \geq 1, \\ s_{mn} &= \tfrac{1}{2} && \text{for} && (m = 0 && \text{and} && n \geq 1) \\ & && \text{or} && (m \geq 1 && \text{and} && n = 0), \\ s_{mn} &= \tfrac{1}{4} && \text{for} && m = 0 && \text{and} && n = 0. \end{aligned}$$

Note that the actual number of coefficients has reduced from $(2M + 1)(2N + 1)$ to $(M + 1)(N + 1)$.

In the $k_x$, $k_y$, $\omega$ domain, the expression for the operator is the well-known phase-shift operator

$$\tilde{F}(k_x, k_y, \Delta z, k) = \exp(\mathrm{j}\sqrt{k^2 - k_x^2 - k_y^2}\, \Delta z). \tag{7}$$

Note that this expression is exact for propagating waves ($k^2 \geq k_x^2 + k_y^2$). In practice we do not require operators that are accurate for all angles of propagation up to 90°. Therefore, we introduce $\alpha_{x,\max}$ as the maximum angle of propagation in the $x$–$z$ plane and $\alpha_{y,\max}$ as the maximum angle of propagation in the $y$–$z$ plane. The area in the $k_x$, $k_y$, $\omega$ domain limited by these angles, is given by the ellipse

$$\frac{k_x^2}{k^2 \sin^2(\alpha_{x,\max})} + \frac{k_y^2}{k^2 \sin^2(\alpha_{y,\max})} = 1. \tag{8}$$

Within the region bounded by this ellipse and the spatial Nyquist frequencies, $\pm k_{x,\mathrm{Nyq}} = \pm \pi/\Delta x$ and $\pm k_{y,\mathrm{Nyq}} = \pm \pi/\Delta y$, (see Fig. 3, light-shaded area), the approximate operator $\langle F \rangle$ must be as accurate as possible. Outside this region, (see Fig. 3, dark-shaded area), there are no special requirements for the phase; however, to avoid instability problems, the amplitude must be smaller than unity. As mentioned before, it is possible to use the extrapolation operator also for suppressing the waves with propagation angles larger than $\alpha_{x,\max}$ and $\alpha_{y,\max}$ respectively, and the evanescent waves ($k_x^2 + k_y^2 > k^2$). In that case the amplitudes (in the dark-shaded area) should be smaller than some user-defined window, $g_{\mathrm{user}}(k_x, k_y)$, e.g. a cosine-window.

The optimization problem can now be formulated as follows:

Find the coefficients $f_{mn}$ of approximated operator $\langle F \rangle$, such that (1) in the light-shaded area as depicted in Fig. 3, the errors (the differences between $\langle \tilde{F} \rangle$ and the exact operator $\tilde{F}$) are as small as possible; (2) in the dark-shaded area of Fig. 3, the

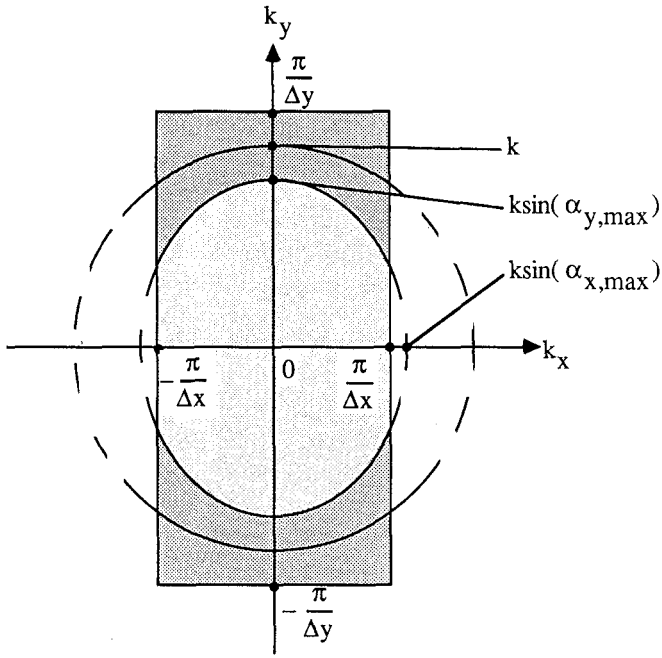


FIG. 3. In the light-shaded area the approximated operator should match the exact phase-shift operator as accurately as possible. In the dark-shaded area the amplitude of the approximate operator should be smaller than unity or some user defined window.

operator satisfies the requirements concerning stability ($\|\langle \tilde{F} \rangle\| \leq 1$) or suppression ($\|\langle \tilde{F} \rangle\| < g_{\text{user}} < 1$).

In the implementation of the optimization method two functions are important: the objective function, which handles requirement 1 and constraint function, which takes care of requirement 2. These will now be discussed.

The error function that is minimized, requirement 1, is called the objective function and is given by

$$\text{OBJ}(F) = \sum_{k_x,\, k_y} (\varepsilon^2 + \lambda \delta^2), \tag{9}$$

for

$$\frac{k_x^2}{k^2 \sin^2(\alpha_{x,\,\max})} + \frac{k_y^2}{k^2 \sin^2(\alpha_{y,\,\max})} \leq 1$$

$$\text{and} \quad |k_x| \leq k_{x,\,\text{Nyq}} \quad \text{and} \quad |k_y| \leq k_{y,\,\text{Nyq}}. \tag{10}$$

Here the symbol $F$ represents the coefficients $f_{mn}$ of extrapolation operator $\langle F \rangle$:

$$F = (f_{00}, f_{10}, \ldots, f_{M0}; f_{01}, f_{11}, \ldots, f_{M1}; \ldots; f_{0N}, f_{1N}, \ldots, f_{MN}),$$

$\varepsilon$ represents the phase error

$$\varepsilon(k_x, k_y, \Delta z, k) = \sqrt{k^2 - k_x^2 - k_y^2}\, \Delta z - \tan^{-1}(\text{Im}\, \langle \tilde{F} \rangle / \text{Re}\, \langle \tilde{F} \rangle),$$

$\delta$ represents the amplitude error

$$\delta(k_x, k_y, \Delta z, k) = 1 - \|\langle \tilde{F} \rangle\|,$$

and $\lambda$ is an optional weighting factor which can be used to control the emphasis put on the amplitude errors relative to the phase errors. In practice we use $\lambda = 1$.

The summation in (9) is carried out over all $k_x$, $k_y$ samples within the area as defined by (10), (light-shaded area in Fig. 3).

In our version of the objective function (9) we did not explicitly include the derivatives of the phase and amplitude errors with respect to $k$, $k_x$ and $k_y$, respectively. This reduces the cost of computation. Instead, we use the optimum operator for wavenumber $k$ as initial estimate for the optimization of the operator for wavenumber $k + \Delta k$ (see also the next section). This has the effect that rapid oscillations in the errors in the $k$-direction generally do not occur and therefore no spurious energy is generated (Holberg 1988). Also rapid oscillations in the errors in the $k_x$- and $k_y$-direction will not occur because of the small number of operator points in the $x$, $y$, $\omega$ domain.

Requirement 2, concerning stability or suppression for the larger wavenumbers, is formulated in a constraint function, which is given by

$$\mathrm{CON}(F) = \sum_{k_x, k_y} \{\max(\|\langle \tilde{F} \rangle\| - g_{\mathrm{user}}, 0)\}^2 = 0, \tag{11}$$

for

$$\frac{k_x^2}{k^2 \sin^2(\alpha_{x,\max})} + \frac{k_y^2}{k^2 \sin^2(\alpha_{y,\max})} > 1$$

$$\text{and} \quad |k_x| \leq k_{x,\mathrm{Nyq}} \quad \text{and} \quad |k_y| \leq k_{y,\mathrm{Nyq}}. \tag{12}$$

The summation in (11) is carried out over all $k_x$, $k_y$ samples within the area as defined by (12), (see the dark shaded area in Fig. 3).

Note that we have formulated our problem as a minimization problem which can be solved with standard minimization software (e.g. NAG). We used a sequential quadratic programming algorithm which requires as input: the objective function, the constraint function and their first derivatives (Gill, Murray and Wright 1981). The algorithm will not be discussed in this paper.

As mentioned before, the procedure should be repeated until the smallest operator with errors below a user specified level is found. After this the optimum operators are ready to be applied.

In Fig. 4 an optimized 3D operator can be seen. Fig. 4a shows the amplitude of the operator in the $x$, $y$, $\omega$ domain, Figs 4b and 4c show the amplitude and the phase of this operator in the $k_x$, $k_y$, $\omega$ domain, and Figs 4d and 4e show the amplitude error and phase error, respectively. The errors are plotted in the region in the $k_x$, $k_y$, $\omega$ domain as defined by (10). The parameters of this operator are $k = 0.2$ $\mathrm{m}^{-1}$, $M = 12$, $N = 12$; number of points in the $k_x$, $k_y$, $\omega$ domain $64 \times 64$, $\Delta x = \Delta y = 7$ m, $\Delta z = 5$ m, $\alpha_{x,\max} = \alpha_{y,\max} = 45°$.

In the derivation of the operator optimization method, we had to assume a laterally constant velocity. However, the actual extrapolation is performed in the $x$,

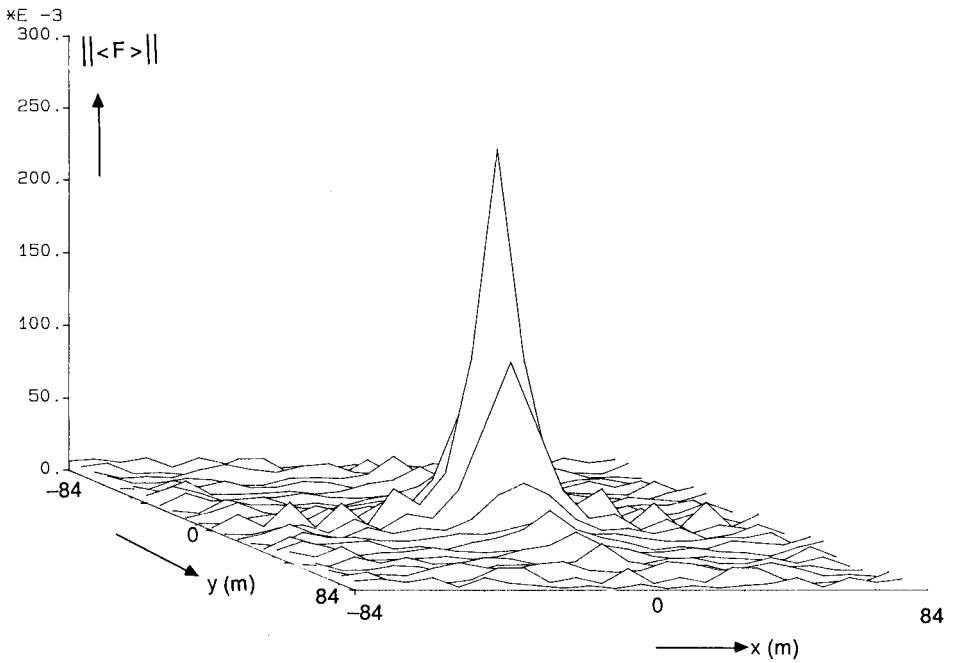


FIG. 4a. Amplitude of optimum 3D operator in the $x$, $y$, $\omega$ domain, 25 * 25 points ($M = 12$, $N = 12$).

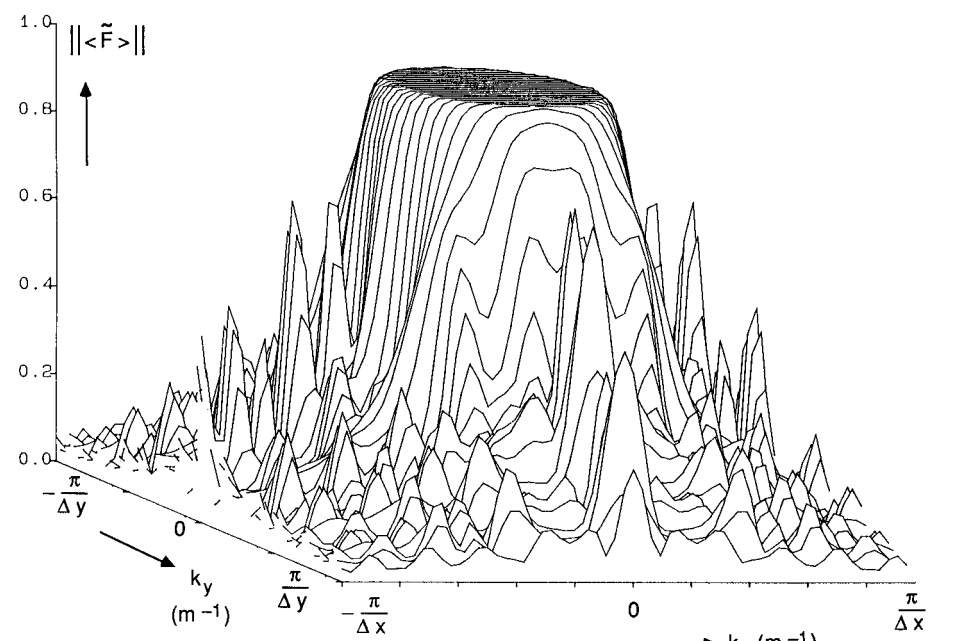


FIG. 4b. Amplitude of optimum 3D operator in the $k_x$, $k_y$, $\omega$ domain.

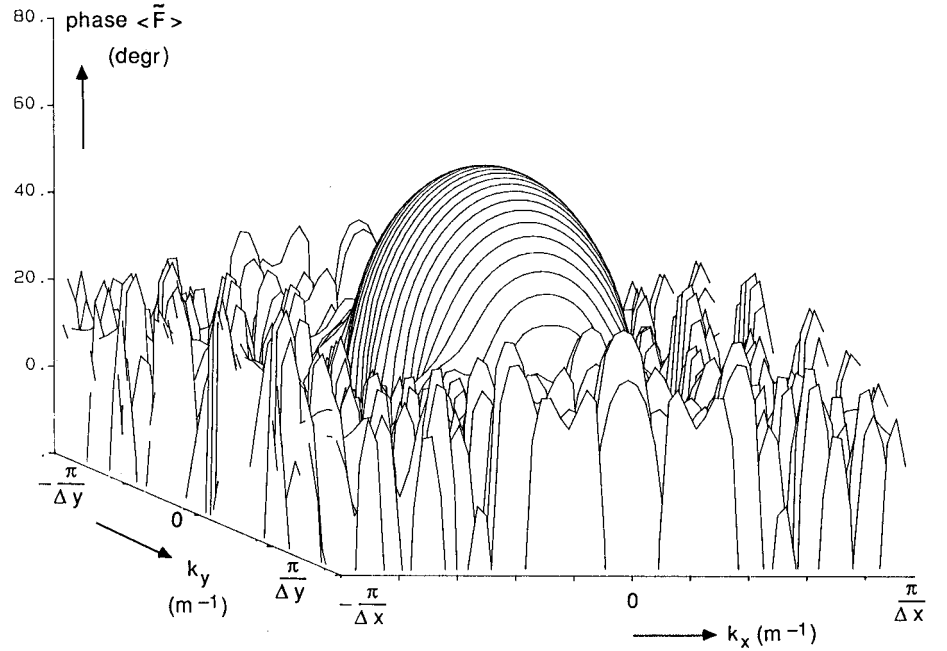


FIG. 4c. Phase of optimum 3D operator in the $k_x$, $k_y$, $\omega$ domain.

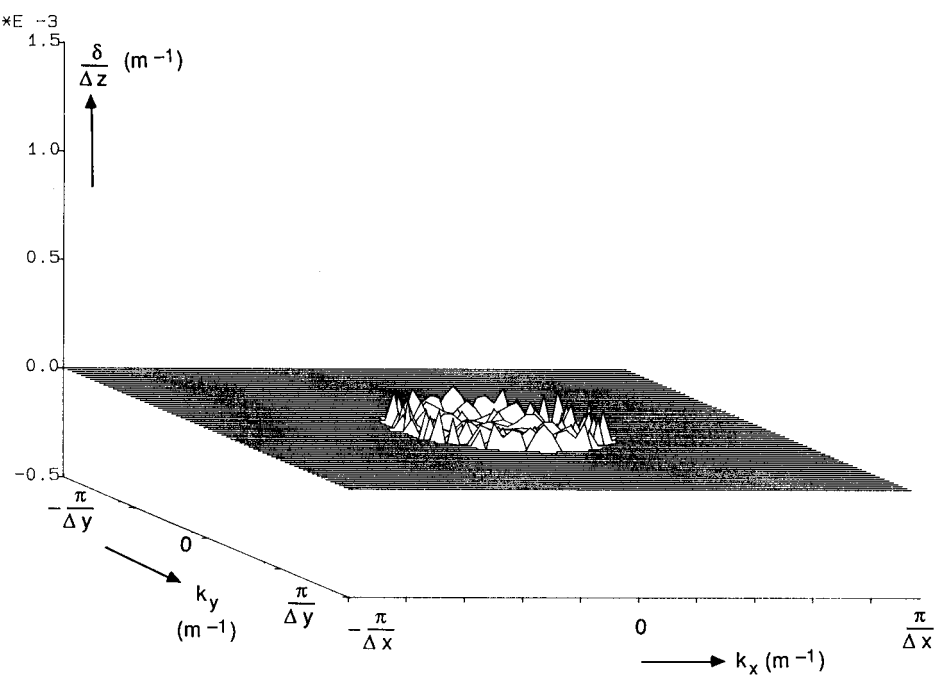


FIG. 4d. Amplitude error of 3D operator in the $k_x$, $k_y$, $\omega$ domain.

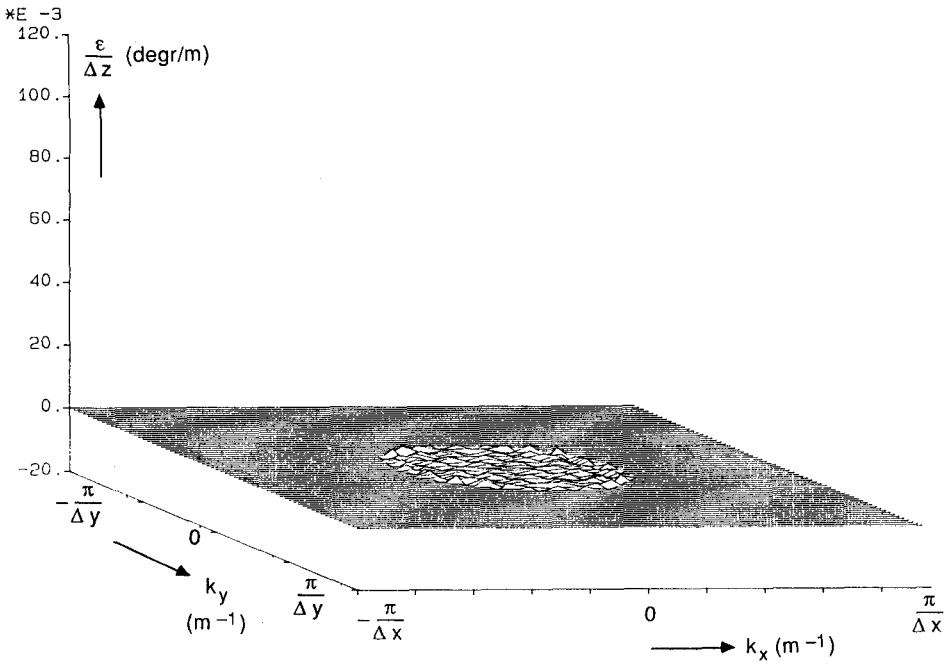


FIG. 4e. Phase error of 3D operator in the $k_x$, $k_y$, $\omega$ domain.

$y$, $\omega$ domain, which allows for lateral velocity variations to be taken into account. Therefore, we apply our extrapolation operators in a space-variant manner, i.e. each output point of the extrapolated data is computed using an operator based on one local value of the wavenumber $k(x, y, z_i, \omega)$ being the ratio $\omega/c(x, y, z_i)$ of the currently treated frequently $\omega$ and the local propagation velocity $c(x, y, z_i)$.

## OPERATOR TABLE

The method of operator generation as described in the previous section is computationally rather intensive. Calculation of the operators during the extrapolation would slow down the speed of the algorithm too much. Furthermore, a specific operator is likely to be used many times, either within one migration process, due to the recursive character of the extrapolation, or in other migration processes. Therefore, the operators are calculated in advance for a range of wavenumber values and stored in a table. During the extrapolation, the local value of the wavenumber, $k = \omega/c(x, y, z_i)$, is determined after which the appropriate operator is selected from the table and applied to the data. However, this procedure could easily lead to very large operator tables, because the maximum number of different $k$-values is equal to the product of the number of subsurface gridpoints and the number of frequency components. (This maximum will occur in the extreme case that the velocity changes from gridpoint to gridpoint in the subsurface.) Therefore, it was decided not

to keep one operator in the table for every $k$-value that occurs, but to define a wavenumber sampling interval $\Delta k$ and to store operators in the table for the following regular range of $k$-values:

$$k_{\min}, k_{\min} + \Delta k, \ldots, k_{\max} - \Delta k, k_{\max},$$

in which $k_{\min} = \omega_{\min}/c_{\max}$ and $k_{\max} = \omega_{\max}/c_{\min}$.

Note that the number of operators in the table is inversely proportional to $\Delta k$. The advantage of such a table is its limited size. The disadvantage is that an actually required operator is not likely to be present. The easiest solution to the problem of 'computing' a required operator from table-operators is rounding, i.e. to select the operator with a $k$-value nearest to the required one. It will be clear that a significant error may be introduced this way. In practice a very small value for $\Delta k$, or, equivalently, a very large number of operators, would be necessary and therefore this method is not recommended.

A better solution is to compute the required operators by interpolation. We found that in general linear interpolation of real and imaginary parts gives good results if $\Delta k$ is in the order of $\Delta\omega/2c_{\max}$. A more sophisticated interpolation method is not recommended because this would deteriorate the efficiency of the migration.

A given table with operators can be used whenever seismic data with the appropriate acquisition parameters ($\Delta x$, $\Delta y$, etc.) need be extrapolated. In the ideal situation one has an operator table available for the most common acquisition parameters. Therefore we do not consider the time consuming optimization procedure as a disadvantage.

## Wavefield Extrapolation Applied in Zero-Offset Migration

We used the wavefield extrapolation method as the basis for a 3D zero-offset depth migration algorithm. It consists of the following two steps:

1. Downward extrapolation, according to

$$P_{zo}(q\Delta x, r\Delta y, z_{i+1}, \omega) \approx \sum_{m=-M}^{M} \sum_{n=-N}^{N} \langle F[m\Delta x, n\Delta y, \Delta z, k_{1/2}(q\Delta x, r\Delta y, z_i, \omega)]\rangle \times P_{zo}[(q-m)\Delta x, (r-n)\Delta y, z_i, \omega]\Delta x\Delta y. \quad (13)$$

where $q = 1$, num$x$ and $r = 1$, num$y$, with

$$k_{1/2}(x, y, z, \omega) = \omega/c_{1/2}(x, y, z) = 2\omega/c(x, y, z).$$

In (13) $P_{zo}$ denotes the temporal Fourier transform of the upgoing pressure field (zero-offset data, based on the 'exploding reflector' model), and the index 1/2 refers to the half-velocity substitution used in the exploding reflector model.

Note that (13) is the same as (2b) except for the so-called half velocity substitution (exploding reflector assumption).

2. Imaging by performing a weighted summation of all frequency components in the frequency band of interest. This yields an estimate of the zero-offset reflectivity ($t = 0$), according to

$$\langle R_{zo}(x, y, z_{i+1})\rangle = \frac{\Delta\omega}{\pi} \operatorname{Re} \sum_{\omega} G(z_{i+1}, \omega) P_{zo}(x, y, z_{i+1}, \omega). \tag{14}$$

$\langle R_{zo}\rangle$ denotes the band-limited estimate of the zero-offset reflectivity and $G$ is an optional weighting function. It can be used for depth-variant spectral shaping, e.g. to compensate for absorption or to suppress coherent noise.

These two steps should be applied recursively for all depth levels of interest. The procedure is visualized in Fig. 5. If a good macro model is used the result will be a detailed zero-offset reflectivity map of the subsurface.

## Computational Aspects

In 3D depth migration large amounts of data are involved, even in the case of zero-offset data. In this section we pay attention to the efficient implementation of the migration scheme. The extrapolation is computationally its most intensive part, and therefore we first discuss how the implementation is optimized by exploiting the symmetry properties of the extrapolation operators. As a measure for the efficiency of our algorithm we use the floating point operation count. After that we give a schematic overview of the data management of our scheme together with a brief discussion on the required core memory space.

Finally a comparison with another full 3D migration algorithm, reverse-time migration, is made.

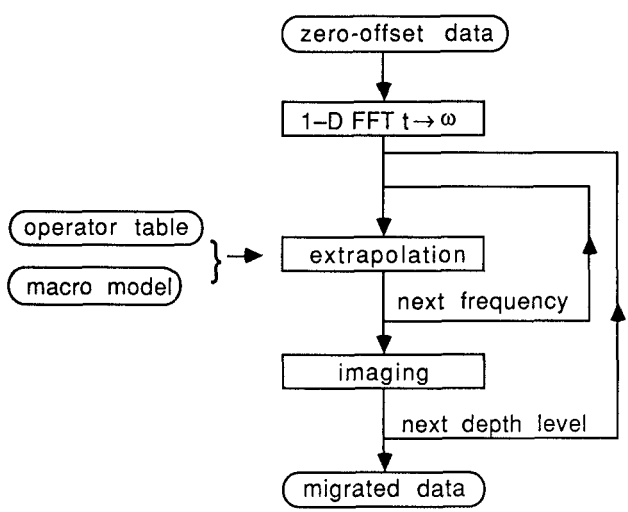


FIG. 5. Flow chart of 3D zero-offset migration scheme.

*Floating point operation count*

One monochromatic extrapolation step according to (13) involves

(numx)(numy)$(2M + 1)(2N + 1)$ complex additions, and

(numx)(numy)$(2M + 1)(2N + 1)$ complex multiplications. (15)

One complex addition is equivalent to two real additions; one complex multiplication is equivalent to four real multiplications and two real additions. Using this, we can express the total number of floating point (*fp*) operations for one monochromatic extrapolation step as

*8(numx)(numy)(2M + 1)(2N + 1) fp-operations.* (16)

Such a result must be interpreted with care, because it does not take into account the machine architecture (scalar $\Leftrightarrow$ vector, add multiply overlap, parallel processing, etc.); however, we consider the *fp*-operation count a satisfactory tool with which to compare different algorithms with similar structures.

In the operator design we made use of the symmetry properties of the operators, (5); if these properties are also used in the application of the operator, (13) becomes

$$\begin{aligned} &P_{zo}(q\Delta x, r\Delta y, z_{i+1}, \omega) \\ &\approx \sum_{n=0}^{N}\sum_{m=0}^{M} s_{mn}\langle F[m\Delta x, n\Delta y, \Delta z, k_{1/2}(q\Delta x, r\Delta y, z_i, \omega)]\rangle \\ &\times \{P_{zo}[(q-m)\Delta x, (r-n)\Delta y, z_i, \omega] + P_{zo}[(q-m)\Delta x, (r+n)\Delta y, z_i, \omega] \\ &+ P_{zo}[(q+m)\Delta x, (r-n)\Delta y, z_i, \omega] + P_{zo}[(q+m)\Delta x, (r+n)\Delta y, z_i, \omega]\}\Delta x\,\Delta y \end{aligned} \tag{17}$$

with $q = 1$, numx and $r = 1$, numy.

One monochromatic extrapolation step according to (17) involves

4(numx)(numy)$(M + 1)(N + 1)$ complex additions, and

(numx)(numy)$(M + 1)(N + 1)$ complex multiplications, or

*14(numx)(numy)(M + 1)(N + 1) fp-operations.* (18)

Note that in general the number of operations in (18) is less than 50% of the number in (16).

However, a further reduction of the number of *fp*-operations is possible if (17) is split into two parts, as follows:

$$\begin{aligned} &P_{zo}(q\Delta x, r\Delta y, z_{i+1}, \omega) \\ &\approx \sum_{n=0}^{N}\sum_{m=0}^{M} s_{mn}\langle F[m\Delta x, n\Delta y, \Delta z, k_{1/2}(q\Delta x, r\Delta y, z_i, \omega)]\rangle \\ &\times \{P'_{zo}[(q-m)\Delta x, (r-n)\Delta y, z_i, \omega] + P'_{zo}[(q-m)\Delta x, (r+n)\Delta y, z_i, \omega]\}\Delta x\,\Delta y, \end{aligned} \tag{19}$$

with $q = 1$, numx and $r = 1$, numy, in which

$$P'_{zo}[(q-m)\Delta x, l\Delta y, z_i, \omega] = P_{zo}[(q-m)\Delta x, l\Delta y, z_i, \omega] + P_{zo}[(q+m)\Delta x, l\Delta y, z_i, \omega]. \quad (20)$$

First the summation according to (20) is performed for $1 = (1 - N)$, (numy + $N$) and $m = 0, M$. This yields

(numy + $2N$)($M$ + 1) complex additions,

application of (19) – in which (20) is included – results in

(numx)[2(numy)($M$ + 1)($N$ + 1) + (numy + $2N$)($M$ + 1)] complex additions,

and

(numx)(numy)($M$ + 1)($N$ + 1) complex multiplications, or, together

*(numx)(M + 1)[10(numy)(N + 1) + 2(numy + 2N)] fp-operations.* (21)

Compared with (18), the number of *fp*-operations in (21) is reduced by about 25%. A high degree of vectorization is possible on vector computers for the implementation of (19) and (20); this is a very important property because the extrapolation is the heart of the migration algorithm.

The floating point operation count for the operator interpolation is not a fixed number. It depends on the structure of the macro model. For one monochromatic extrapolation step it is somewhere in between

*0 fp-operations*, for a homogeneous macro model, and

*6(numx)(numy)(N + 1)(M + 1) fp-operations*, for a fully inhomogeneous model.

In a practical subsurface macro model there will be large homogeneous areas, and therefore we neglect the operator interpolation with respect to the extrapolation of the data. We also neglect the imaging step because its contribution to the *fp*-operation count is only in the order of 0.1%.

Thus, an estimate of the total number of *fp*-operations of the migration scheme can be obtained by multiplying the number of operations in (21) by the number of frequency components and the number of extrapolation steps:

*(numz)(numf)(numx)(M + 1)*

*[10(numy)(N + 1) + 2(numy + 2N)] fp-operations.* (22)

numz is the number of extrapolation steps, numf is the number of frequency components; in practice numf $\approx \frac{1}{4}$ numt, where numt is the number of recorded time samples.

## *Data management*

In this section a brief description of the data flow of our migration algorithm is given. In Fig. 6 the procedure is visualized. We start with a Fourier transformed zero-offset data set ($x$, $y$, $\omega$ domain), stored in a data file on disk. A table containing

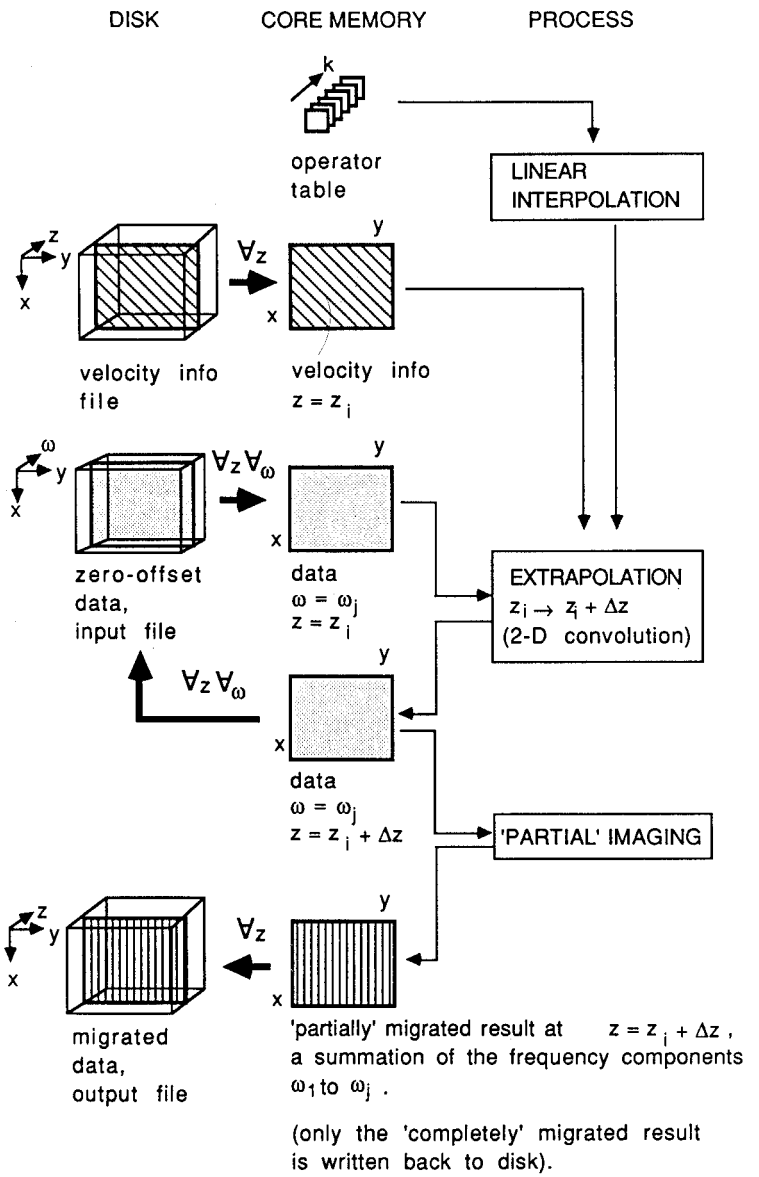


FIG. 6. Data management of the 3D table-driven migration algorithm.

a set of optimized extrapolation operators must be available in the core memory. A velocity information file (macro model) must be present on disk. It is possible to use this file also as output file. This means a reduction of the required disk space. However, the original velocity information in the file will get lost.

*Depth loop:*

— A 2D $x$–$y$ panel, constant depth $z = z_i$, containing velocity information of the current depth level $z_i$ is read from the velocity information file on disk and stored in the core memory. The extrapolation from the current depth level to the next, $z_i \rightarrow z_i + \Delta z$, can now be performed.

*Frequency loop:*

— A 2D $x$–$y$ data panel, depth level $z = z_i$ and frequency $\omega = \omega_j$, is read from the data file on disk and stored in core memory.

*X, Y loop:*

— The local $k$-value is determined: $k = \omega_j / c_{1/2}(x, y, z_i)$.

— Only if this value differs from the previous one, a new extrapolation operator for this $k$-value is calculated by means of linear interpolation between two operators from the table.

— The operator is applied to the data, according to (19).

*End of X, Y loop:*

Once all points have been extrapolated, the result is an extrapolated $x$–$y$ data panel at $z = z_i + \Delta z$ for frequency component $\omega_j$.

— This panel is written back to the data file on disk, where it replaces the $x$–$y$ data panel at $z = z_i$ for frequency component $\omega_j$.

— Also this panel is used to carry out a part of the imaging step: we don't need to wait until all frequency components have been treated to do the summation. Instead we can simply add the real parts of the extrapolated result for the current frequency component, $\omega = \omega_j$, to the 'partially' imaged result of the previously treated frequency components $\omega_1$ to $\omega_{j-1}$. This yields a partially imaged $x$–$y$ panel at $z = z_i + \Delta z$, which is the result of a summation of the frequency components $\omega_1$ to $\omega_j$.

It is called partial because not all frequencies have contributed to the result yet.

*End of frequency loop.*

Once all frequencies have been treated, the output is:

1. the migrated $x$–$y$ panel at $z = z_i + \Delta z$.

— this panel is written to the output file on disk.

2. a 3D zero-offset data set in the $x$, $y$, $\omega$ domain at $z = z_i + \Delta z$, stored in the data file on disk, which will be used as input for the extrapolation to the next depth level.

*End of depth loop.*

Once this procedure has been repeated for all depth levels, the result is the migrated data section in the $x$, $y$, $z$ domain.

Experience taught us that the speed of the migration algorithm was never bounded by the I/O between disk and core memory. The wavefield extrapolation almost completely determines the calculation time. It should be noted that during the migration process the extrapolated zero-offset data set in the $x$, $y$, $\omega$ domain is available as an intermediate result for any depth level. Examination of those intermediate results, after inverse Fourier transformation to the $x$, $y$, $t$ domain, may contribute to a better understanding of the migrated result.

The migration can be easily implemented in parallel, because the extrapolation is performed per frequency component. Working per frequency component offers a natural way of dividing a problem into independent parts, each of which can be treated by a separate processor.

### *Required core memory space*

During the migration process the next amounts of data are kept in core memory (see also Fig. 6)

— a 2D $x$–$y$ panel, $z$ constant, containing velocity information, size:

*(numx)(numy) real numbers.*

— a 2D $x$–$y$ panel, $\omega$ constant, containing monochromatic zero-offset data, size:

(numx)(numy) complex numbers, or

*2(numx)(numy) real numbers.*

— a 2D $x$–$y$ panel, $\omega$ constant, containing extrapolated monochromatic zero-offset data, size:

(numx)(numy) complex numbers, or

*2(numx)(numy) real numbers.*

— a 2D $x$–$y$ panel, $z$ constant, containing (partially imaged) migrated data, size:

*(numx)(numy) real numbers.*

— a table containing the extrapolation operators, size:

(numop)$(N + 1)(M + 1)$ complex numbers, or

*2(numop)(N + 1)(M + 1) real numbers,*

numop represents the number of operators in the table.

Hence, the core memory space must be large enough for

$$6(\text{numx})(\text{numy}) + 2(\text{numop})(N + 1)(M + 1) \text{ real numbers.} \tag{23}$$

*Example.* With the next parameters: numx = 500, numy = 100, $M = 12$, $N = 12$, a table containing 400 operators and 4 bytes per real number, we find that a core memory of 1.7 Mbyte is required.

### *Cost comparison with other migration methods*

The use of '2 times 2D migration' methods, or methods based on operator splitting, reduces the computational costs considerably. However, as these methods do not have the quality of full 3D migration (see also 3D example 2), a fair cost comparison cannot be made.

Therefore, we compare the cost of our migration with that of the acoustic 3D

reverse-time migration algorithm as discussed by Chang and McMechan (1989). In reverse-time migration the recursive extrapolation is performed backwards in time without dip limitations. At each extrapolation step the zero-offset data provide the boundary conditions at the surface. Starting at the maximum registration time, the extrapolation is continued to zero time. The extrapolated section at zero time is the migrated result: all depths are imaged simultaneously. The expression for reverse-time extrapolation is given by

$$\begin{aligned} P^{t-\Delta t}_{k,m,n} &= 2(1-3a^2)P^{t}_{k,m,n} - P^{t+\Delta t}_{k,m,n} \\ &\quad + a^2[P^{t}_{k+1,m,n} + P^{t}_{k-1,m,n} + P^{t}_{k,m+1,n} \\ &\quad + P^{t}_{k,m-1,n} + P^{t}_{k,m,n+1} + P^{t}_{k,m,n-1}], \end{aligned} \tag{24}$$

where $P^{t}_{k,m,n}$ is an abbreviation for the pressure field $p(k\Delta x_{rt}, m\Delta y_{rt}, n\Delta z_{rt}, t)$, $\Delta x_{rt} = \Delta y_{rt} = \Delta z_{rt} = h$ is the grid spacing, the index rt refers to reverse-time and $a = a(x, y, z) = c_{1/2}(x, y, z)\Delta t/h$. From (24) it follows that the extrapolation of one sample requires 9 *fp*-operations (in case of a homogeneous medium). The total number of *fp*-operations is found by multiplying this with the number of gridpoints and the number of time steps $\text{numt}_{rt}$:

$$9(numx_{rt})(numy_{rt})(numz_{rt})(numt_{rt}). \tag{25}$$

To be able to compare this result with the one in (22), one should keep in mind that the spatial and temporal intervals in reverse-time migration are small in order to preclude instability and/or grid dispersion:

$$\begin{aligned} numx_{rt} &\approx 5numx, \\ numy_{rt} &\approx 5\ numy, \\ numz_{rt} &\approx 5\ numz, \\ numt_{rt} &\approx 5\ numt \approx 20\ numf. \end{aligned} \tag{26}$$

Using this we find for the total number of *fp*-operations in reverse-time migration:

$$22\,500\ (numx)(numy)(numz)(numfr) \tag{27}$$

This is in the order of 10 times larger than the number of *fp*-operations for our method (22).

The required memory space for reverse-time migration is computed as follows:
— a 3D $x$–$y$–$z$ volume containing velocity information, size:

$$125(numx)(numy)(numz)\ real\ numbers.$$

— two 3D $x$–$y$–$z$ volumes containing data at two times, size:

$$250(numx)(numy)(numz)\ real\ numbers.$$

Hence, unless one accepts an enormous amount of I/O, the core memory space must be large enough for

$$375(numx)(numy)(numz)\ real\ numbers. \tag{28}$$

*Example.* With the next parameters: numx = 500, numy = 100, numz = 150 and 4 bytes per real number, we find that the reverse-time migration requires a core memory of more than 10 Gbyte. This number should be compared with the 1.7 Mbyte that is required by table-driven migration (see the previous example). Note that 10 Gbyte is too much for an implementation on present computers.

## RESULTS/EXAMPLES

### *2D example 1*

Although we described the migration method for the 3D case, we first implemented the scheme for the 2D case. To test it, synthetic exploding reflector data were generated with an accurate finite-difference scheme, in which diffraction energy is correctly incorporated (Kelly, Alford and Whitmore 1982). The subsurface model is shown in Fig. 7a. It consists of a reflector below a pinch-out structure and significant lateral and vertical velocity variations are present. The zero-offset response of only the 'exploding reflector' was modelled, see Fig. 7b. The shape of the reflector is distorted by the propagation effects of the overburden. In Fig. 7c the migrated result is shown. Notice the excellent positioning and focussing of the reflector.

Some parameters: temporal sampling interval $\delta t = 4$ ms, frequency content $f_{\min} = 10$ Hz, $f_{\max} = 70$ Hz, number of frequency components numf = 62, minimum

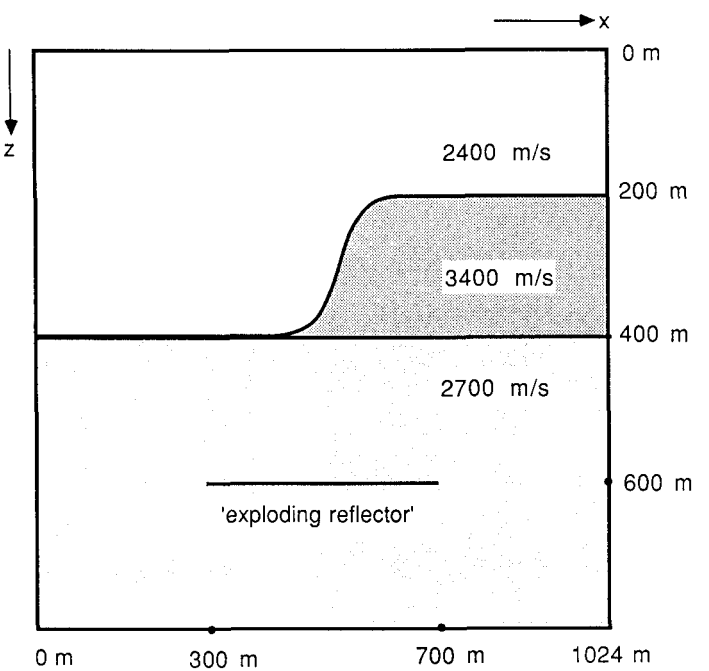


FIG. 7a. Subsurface model used in the generation of zero-offset data. Only the response of the 'exploding reflector' was modelled.

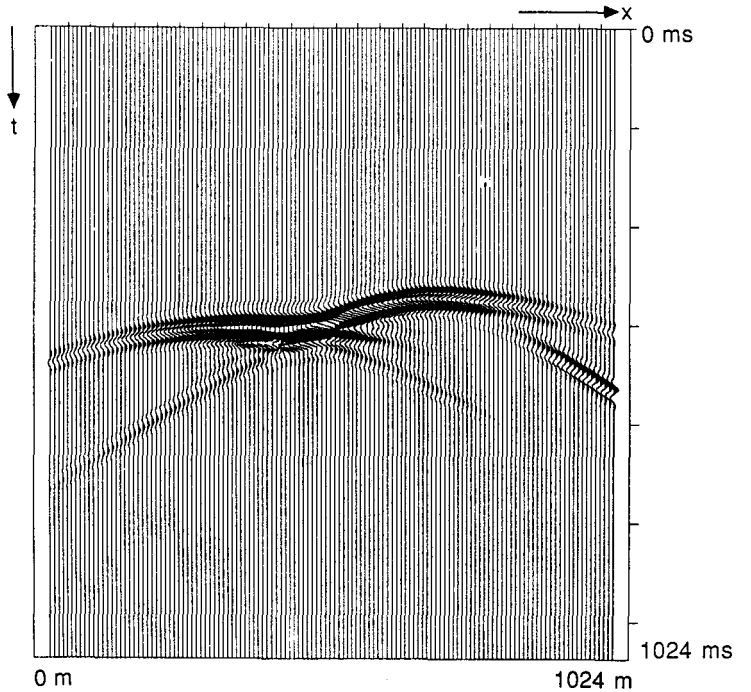


FIG. 7b. Zero-offset response of the reflector.

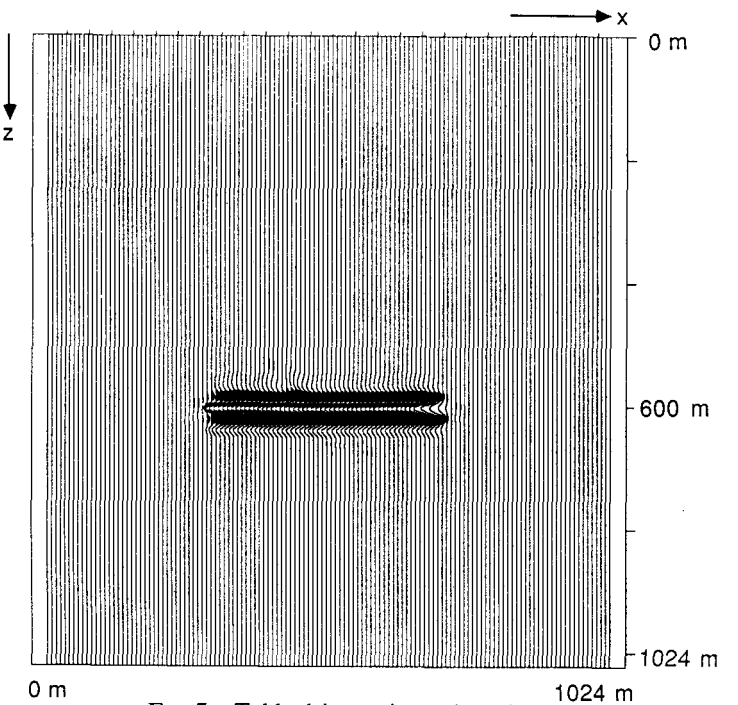


FIG. 7c. Table-driven migrated result.

velocity $c_{min} = 2400$ m/s, maximum velocity $c_{max} = 3400$ m/s, grid size 256 × 256 (numx, numz), horizontal grid spacing $\Delta x = 8$ m, vertical grid spacing $\Delta z = 4$ m, number of operator points 19 ($M = 9$), maximum angle of propagation used in the operator design $\alpha_{max} = 65°$, number of operators 185 (in the range from $k_{min} = 0.036$ $m^{-1}$ to $k_{max} = 0.368$ $m^{-1}$).

### *2D example 2*

In Fig. 8a a real stacked section is shown. In this section a fault can be observed from which diffraction tails originate. We found that satisfactory migration results were reached using a simple constant velocity macro model. As can be seen from the depth section in Fig. 8b, the result shows a very good definition of the fault.

Some parameters: temporal sampling interval $\delta t = 4$ ms, frequency content $f_{min} = 15$ Hz, $f_{max} = 100$ Hz, number of frequency components numf = 175, velocity $c = 2025$ m/s, grid size 256 × 256 (numx, numz), horizontal grid spacing $\Delta x = 5$ m, vertical grid spacing $\Delta z = 4$ m, number of operator points 23 ($M = 11$), maximum angle of propagation used in the operator design $\alpha_{max} = 65°$, number of operators 177 (in the range from $k_{min} = 0.094$ $m^{-1}$ to $k_{max} = 0.627$ $m^{-1}$).

### *3D example 1*

The 3D migration scheme was first tested by examination of its impulse response. We therefore placed a wavelet at $t = 300$ ms on the middle trace of an

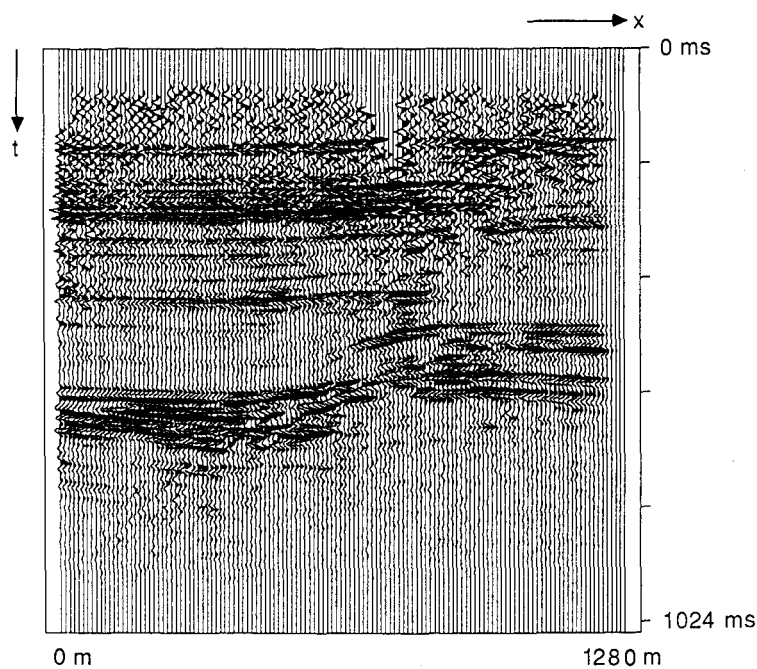


FIG. 8a. Real stacked data. Notice the diffraction tails that originate from the fault.

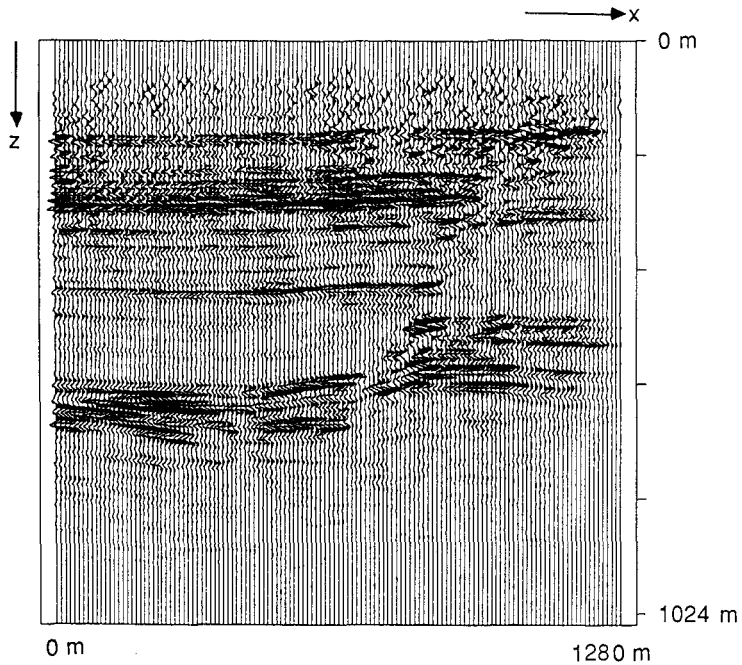


FIG. 8b. Table-driven migrated result.

'empty' zero-offset data set. In Fig. 9a a vertical ($x$–$t$) cross-section at $y = 0$ m is shown. After 3D migration, a vertical ($x$–$z$) cross-section at $y = 0$ m, and a horizontal ($x$–$y$) depth slice at $z = 230$ m can be seen in Figs 9b and 9c, respectively. As expected the migrated result is a half-sphere.

Some parameters: temporal sampling interval $\delta t = 4$ ms, frequency content $f_{\min} = 20$ Hz, $f_{\max} = 100$ Hz, number of frequency components numf = 32, velocity $c = 2000$ m/s, grid size 128 × 64 × 70 (numx, numy, numz), horizontal grid spacing $\Delta x = 6$ m and $\Delta y = 7$ m, vertical grid spacing $\Delta z = 6.25$ m, number of operator points 19 × 17 ($M = 9$, $N = 8$), maximum angles of propagation used in the operator design $\alpha_{x,\max} = 50°$ and $\alpha_{y,\max} = 50°$, number of operators 54 (ranging from $k_{\min} = 0.123\ \text{m}^{-1}$ to $k_{\max} = 0.515\ \text{m}^{-1}$).

For this configuration, the program required 35 min user time at a rate of 8.2 Mflop on a Convex C1-XP.

### *3D example 2*

The impulse response in example 1 was made for a homogeneous medium. To test the performance of the migration also for a 3D inhomogeneous medium, the following example was made. We modelled the response of a horizontal square reflector situated below a 3D synclinal structure (see Fig. 10a). At its boundary, the velocity changes from 2400 m/s in the upper part to 3600 m/s in the lower part. A time slice at $t = 0.28$ s, a vertical ($x$–$t$) cross-section of the zero-offset data at

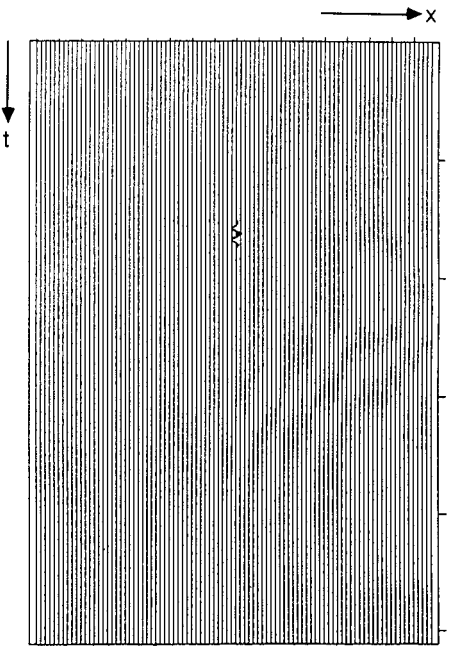


FIG. 9a. 'Zero-offset' section, used as input data to determine the impulse response of the 3D migration algorithm.

$y = 455$ m and a vertical ($y$–$t$) cross-section at $x = 520$ m and are shown in Figs 10b, c and d respectively. The synclinal structure has a strong focussing effect on the reflected energy. In order to see also the diffracted energy, the pictures of the zero-offset data have been clipped 15 dB. The migrated result can be seen in Fig. 11. The vertical ($x$–$z$) and ($y$–$z$) cross-sections confirm that the reflector is horizontal whereas

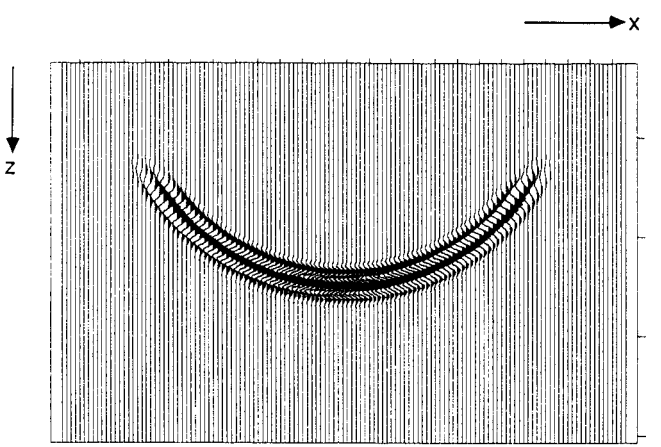


FIG. 9b. Vertical cross-section, ($x$–$z$) panel at $y = 0$ m, of the 3D migrated result.

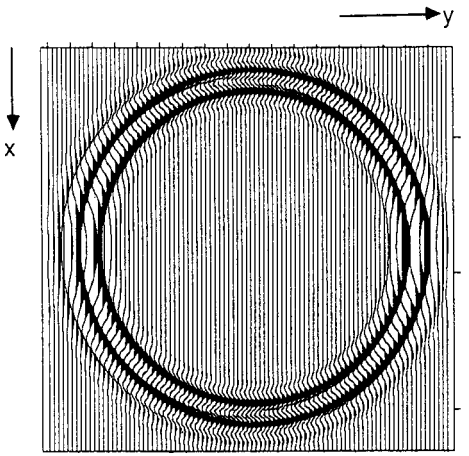


FIG. 9c. Depth slice at a depth of 230 m of the 3D migrated result.

the depth slice at $z = 350$ m shows the correct square shape of the reflector. We also made amplitude cross-sections of Figs. 11b and c, which are shown in Figs 11d and e. Notice the constant amplitude along the reflector.

The parameters for the migration are: temporal sampling interval $\delta t = 4$ ms, number of time samples 128, frequency content $f_{\min} = 20$ Hz, $f_{\max} = 70$ Hz, number of frequency components numf = 28, minimum velocity $c_{\min} = 2400$ m/s, maximum velocity $c_{\max} = 3600$ m/s, grid size 128 × 128 × 100 (numx, numy, numz), horizontal grid spacing $\Delta x = 8$ m and $\Delta y = 7$ m, vertical grid spacing $\Delta z = 5$ m, number of operator points for migration 23 × 27 ($M = 11$, $N = 13$), maximum angles of propagation used in the operator design $\alpha_{x,\max} = 50°$ and $\alpha_{y,\max} = 50°$, number of operators 191 (ranging from $k_{\min} = 0.069\ \text{m}^{-1}$ to $k_{\max} = 0.366\ \text{m}^{-1}$).

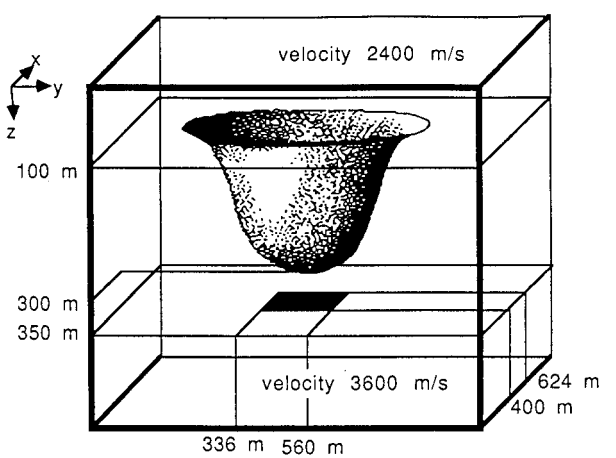


FIG. 10a. 3D subsurface model. The response of the square reflector below the 3D synclinal structure was modelled.

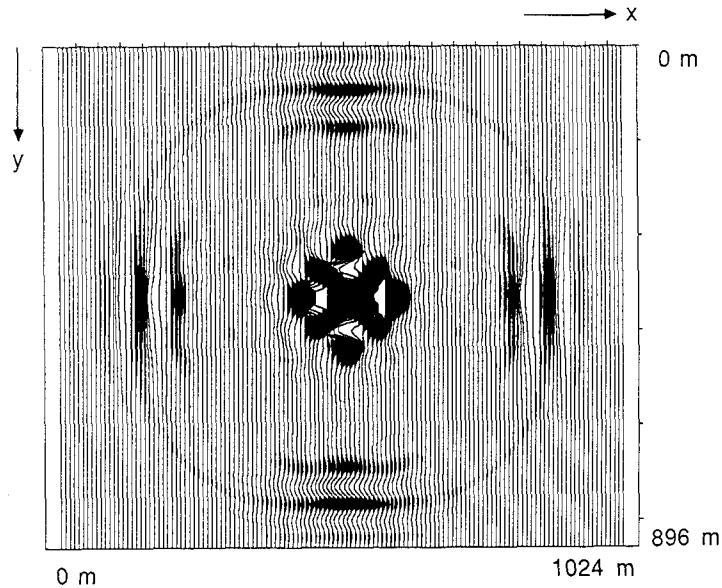


FIG. 10b. Zero-offset data, time slice at 280 ms.

The number of floating point operations for this example is $7.84 \times 10^{10}$, and it took 156 min user time on a Convex C1-XP (at a rate of 8.4 Mflop).

*Comparison with '2 times 2D migration'.* In the seismic industry 3D post-stack migration is usually carried out as a sequence of 2D migrations. In two-pass migration (or '2 times 2D migration') all 2D cross-sections in one lateral direction of a 3D data set are migrated first. Next, all 2D cross-sections of the result are migrated in the perpendicular direction. In case of a homogeneous medium, the results of two-pass migration and full 3D migration are practically equivalent. To examine the

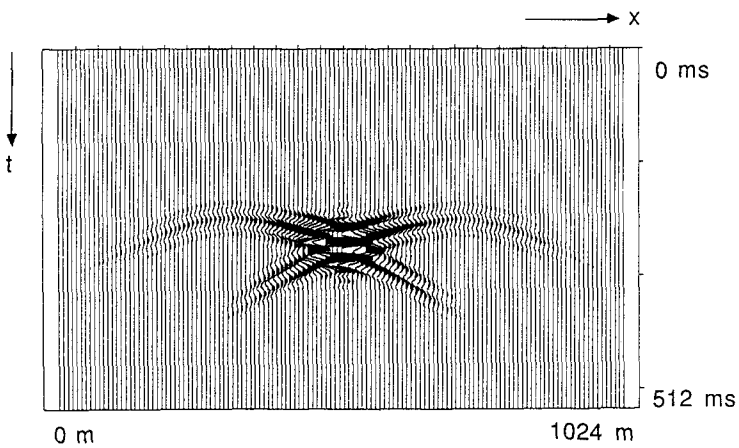


FIG. 10c. Zero-offset data, vertical cross-section, ($x$–$y$) panel at $y = 455$ m.

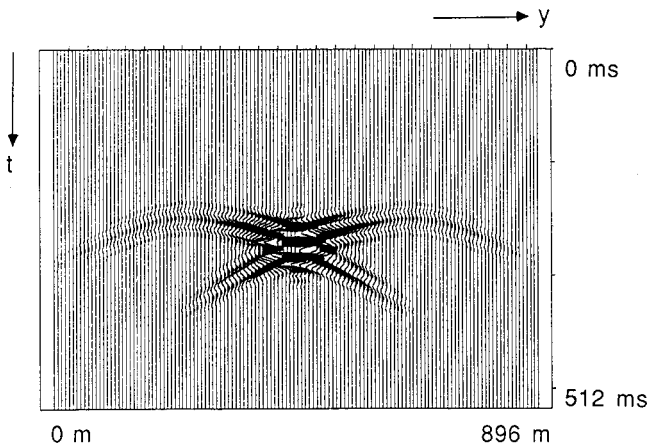


FIG. 10d. Zero-offset data, vertical cross-section, ($y$–$t$) panel at $x = 520$ m.

results of two-pass migration in case of lateral velocity variations we did the following experiment, starting with the same zero-offset data as in the previous example. The first pass was carried out in the $x$-direction using 2D time migration with $\delta t = 4$ ms. The second pass was carried out in the $y$-direction using 2D depth migration with $\Delta z = 5$ m. In Fig. 12 the results are shown. A depth slice at 350 m, a vertical $x$–$z$ cross-section and a vertical $y$–$z$ cross-section of the migrated result can

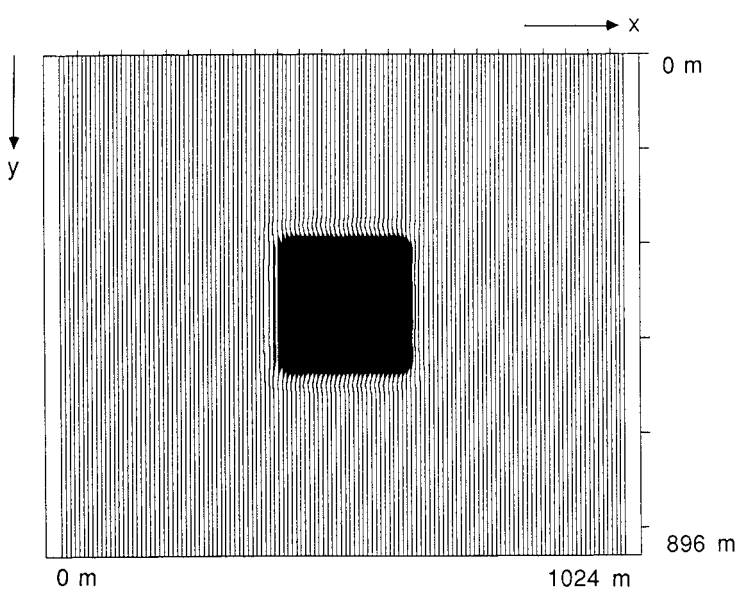


FIG. 11a. 3D migrated data, depth slice at a depth of 350 m.

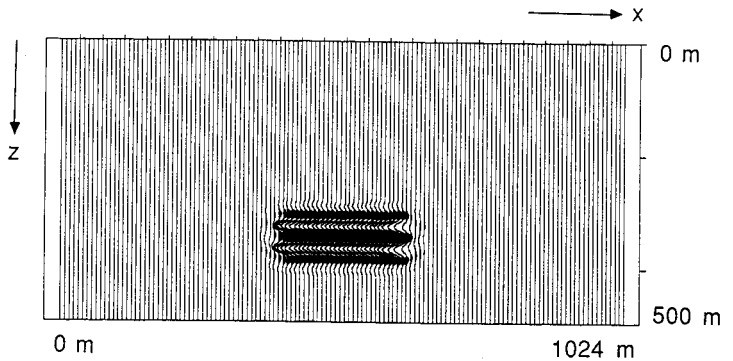


FIG. 11b. 3D migrated data, vertical cross-section, $(x–z)$ panel at $y = 455$ m.

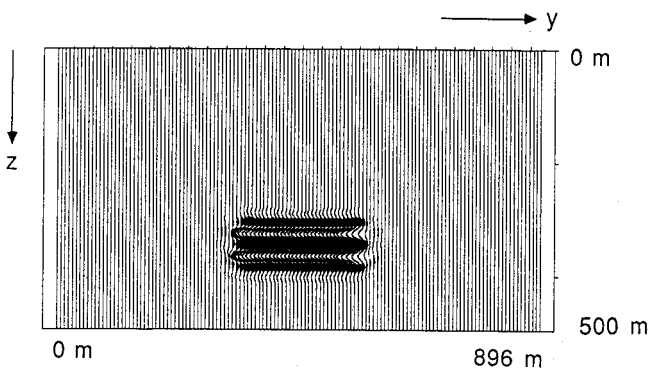


FIG. 11c. 3D migrated data, vertical cross-section, $(y–z)$ panel at $x = 520$ m.

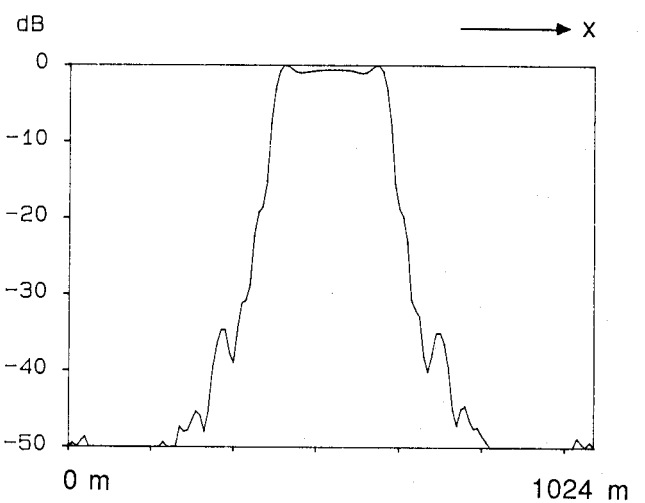


FIG. 11d. Amplitude cross-section of Fig. 11b.

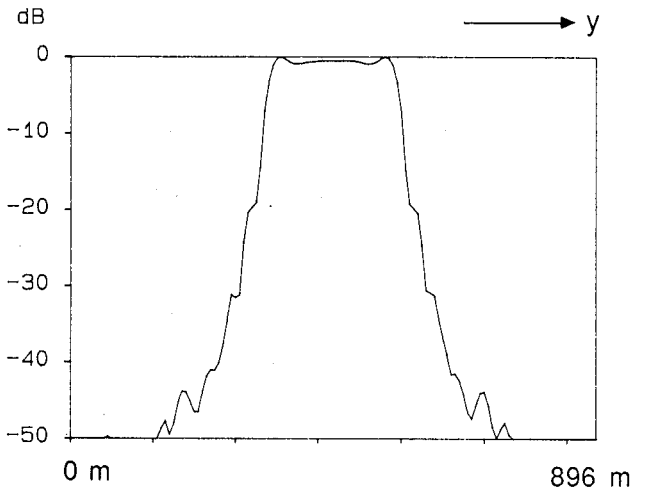


FIG. 11e. Amplitude cross-section of Fig. 11c.

be seen in Figs 12a, b and c respectively. Amplitude cross-sections of Figs 12b and c are shown in Figs 12d and e. The migrated result in the $x$–$z$ cross-section, Fig. 12b, is not correct: the horizontal reflector is imaged as an anticlinal structure. Also its size is too large (compare with Fig. 11b). Thr $y$–$z$ cross-section in Fig. 12c is better: it is very similar to the result of full 3D migration. However, the amplitude cross-section in Fig. 12e is not as regular as the amplitude cross-section of the 3D migrated result, Fig. 11e.

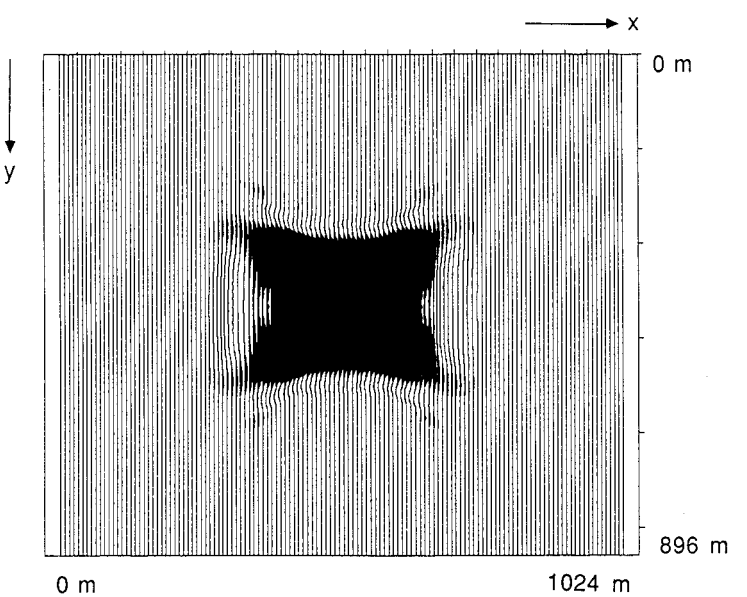


FIG. 12a. Two-pass migrated data, first pass in $x$-direction, second pass in $y$-direction, depth slice at 350 m.

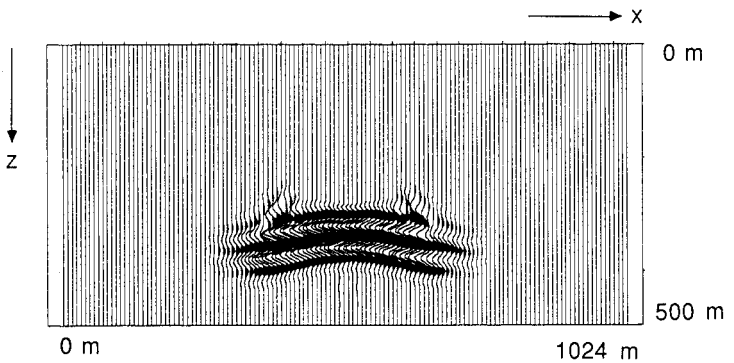


FIG. 12b. Two-pass migrated data, first pass in $x$-direction, second pass in $y$-direction, vertical cross section, ($x$–$z$) panel at $y = 455$ m.

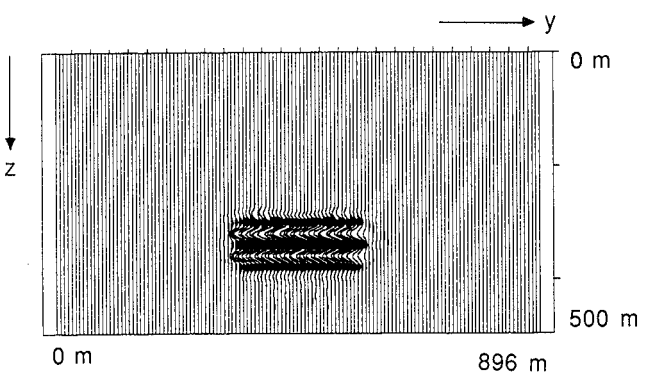


FIG. 12c. Two-pass migrated data, first pass in $x$-direction, second pass in $y$-direction, vertical cross-section, ($y$–$z$) panel at $x = 520$ m.

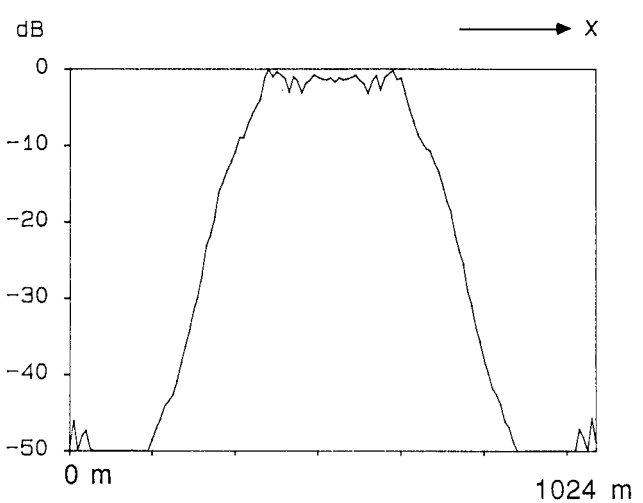


FIG. 12d. Amplitude cross-section of Fig. 12b.

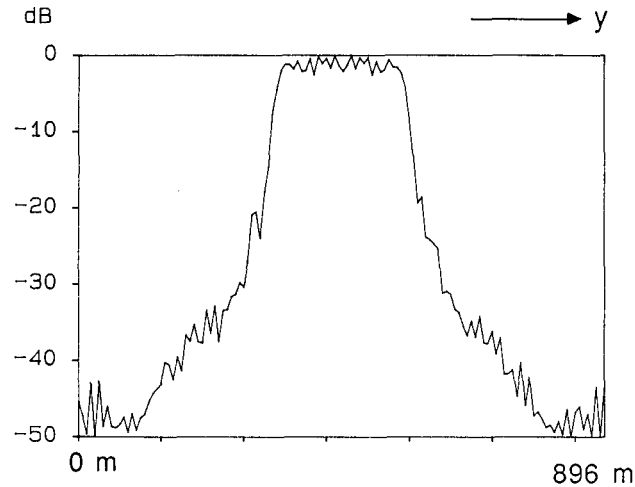


FIG. 12e. Amplitude cross-section of Fig. 12c.

The result of two-pass migration is not unique; it depends on the direction in which the first pass is carried out. To show this, the previous experiment was repeated, this time with the first pass carried out in the $y$-direction and the second pass in the $x$-direction. The results are shown in Fig. 13. Notice the big differences between Fig. 12 and Fig. 13. As expected, this time the $x$–$z$ cross-section of the migrated result is the best, see Fig. 13 b.

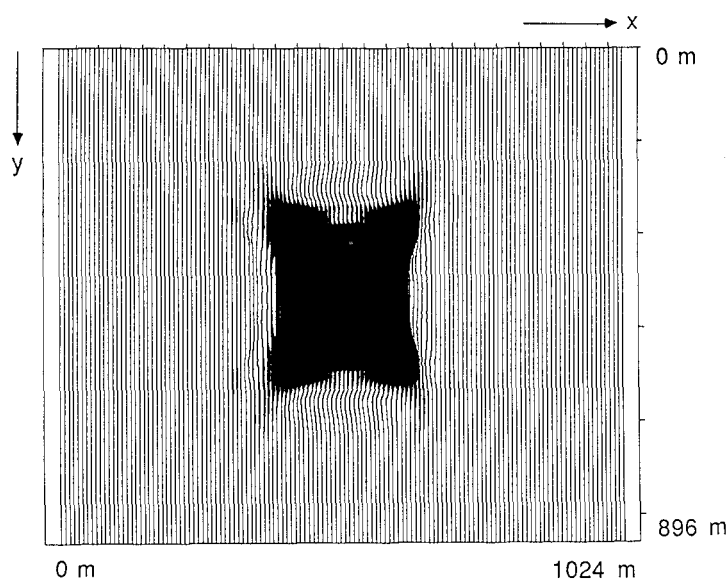


FIG. 13a. Two-pass migrated data, first pass in $y$-direction, second pass in $x$-direction, depth slice at 350 m.

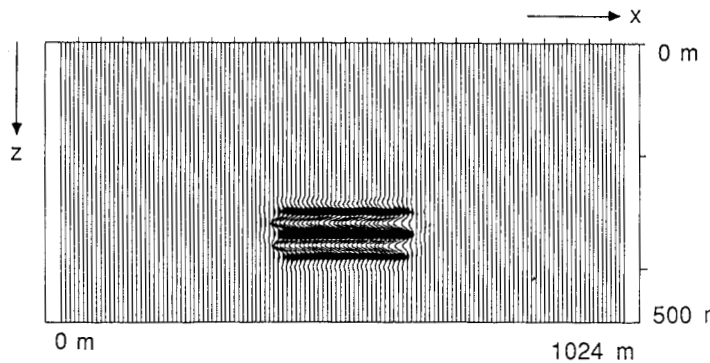


FIG. 13b. Two-pass migrated data, first pass in $y$-direction, second pass in $x$-direction, vertical cross-section, ($x$–$z$) panel at $y = 455$ m.

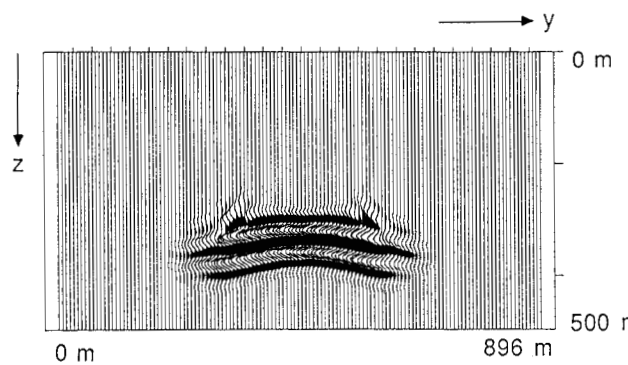


FIG. 13c. Two-pass migrated data, first pass in $y$-direction, second pass in $x$-direction, vertical cross-section, ($y$–$z$) panel at $x = 520$ m.

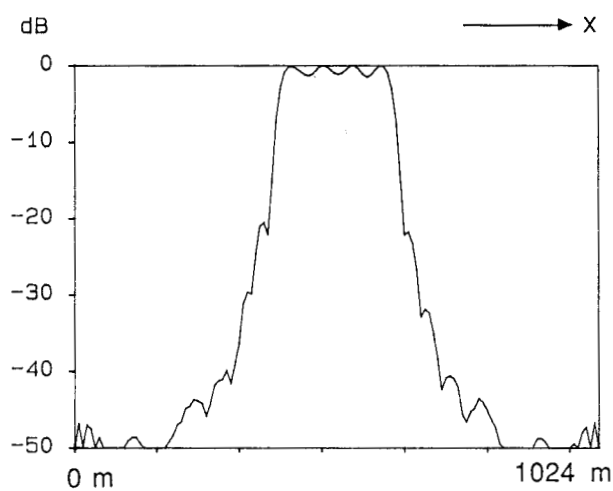


FIG. 13d. Amplitude cross-section of Fig. 13b.

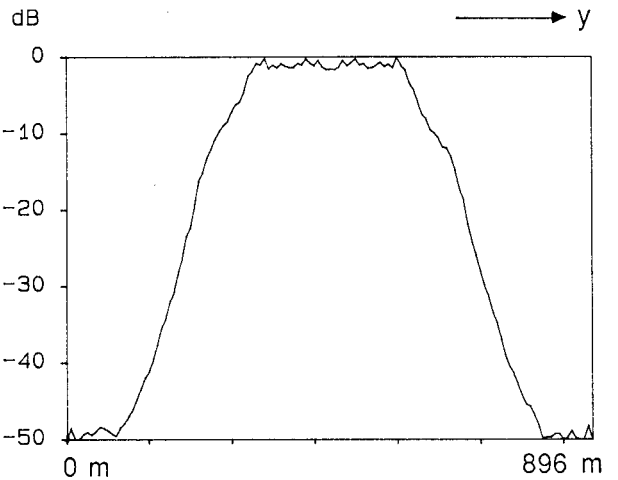


FIG. 13e. Amplitude cross-section of Fig. 13c.

These examples confirm the well-known result that in case of strong lateral velocity variations '2 times 2D migration' does no longer yield acceptable results. In this case full 3D depth migration is required (Fig. 11).

## CONCLUSIONS

In today's practice of seismic processing, 3D post-stack migration is usually carried out as a sequence of 2D migrations, the so-called two-pass migration. However, it has long been recognized that two-pass migration gives satisfactory results only in the case of media with small velocity variations. In more complex situations with significant velocity variations, especially in the lateral directions, a full 3D depth migration should be used.

We presented a method for full 3D wavefield extrapolation that we used as the basis for a zero-offset depth migration scheme. The wavefield extrapolation is performed in the space–frequency domain with optimized operators that have been pre-calculated and stored in a table. The operators are optimum in the sense that they have the smallest possible size given a user-specified accuracy and dip range. This means that steeply dipping events can be accurately treated in an efficient way. Because the extrapolation is performed recursively in the space domain, both vertical and lateral velocity variations can be handled. As shown in the examples, the migrated events are positioned well and diffraction energy is focused correctly. As expected, this is not the case if the same zero-offset data are processed with a two-pass migration algorithm. Once more, this confirms that full 3D techniques are required in case of significant velocity variations.

A disadvantage of 3D processing is the huge computational effort that is usually required. Therefore, we paid special attention to efficiency aspects. The use of optimized operators that have been stored in a table contributes considerably to a high

efficiency. Furthermore, the symmetry properties of the operators are exploited which reduces the computational costs. Because the wavefield extrapolation can be formulated in terms of vector operations, the method is very well suited for a vector computer. Also a parallel implemenation could be realized in a natural way, because the frequency components are treated independently. In our algorithm, working per frequency component has the advantage that the requirements concerning the computer memory remain moderate.

Together these properties make full 3D zero-offset depth migration feasible on mini super computers.

## ACKNOWLEDGEMENTS

The investigations were supported by the sponsors of the TRITON consortium project at the Laboratory of Seismics and Acoustics of the Delft University of Technology, The Netherlands.

## REFERENCES

BERKHOUT, A.J. 1984. *Seismic Migration: Imaging of Acoustic Energy by Wavefield Extrapolation. B. Practical Aspects*, Elsevier Science Publishers.

BERKHOUT, A.J. 1985. *Seismic Migration: Imaging of Acoustic Energy by Wavefield Extrapolation. A. Theoretical Aspects.* Elsevier Science Publishers.

BERKHOUT, A.J. and VAN WULFFTEN PALTHE, D.W. 1979. Migration in terms of spatial deconvolution. *Geophysical Prospecting* **27,** 261–291.

CHANG, W. and MCMECHAN, G.A. 1989. 3D acoustic reverse-time migration. *Geophysical Prospecting* **37,** 243–256.

GILL, P.E., MURRAY, W. and WRIGHT, M.H. 1981. *Practical Optimization.* Academic Press.

HOLBERG, O. 1988. Towards optimum one-way wave propagation. *Geophysical Prospecting* **36,** 99–114.

KELLY, K.R., ALFORD, R.M. and WHITMORE, N.D. 1982. Modelling – the forward method. *Concepts and Techniques in Oil and Gas Exploration*, Jain, K.C. and deFigueiredo, R.J.P., eds., 91–114. Society of Exploration Geophysicists.

RAOULT, J. and DEZARD, Y. 1985. Depth migration of seismic data by local extrapolation operators. 55th SEG meeting, Washington D.C., Expanded Abstracts, 316–318.

VAN DER MADE, P.M. 1988. Determination of macro subsurface models by generalized inversion. Doctoral thesis, Delft University of Technology.

YILMAZ, O., CHAMBERS, R., NICHOLS, D. and ABMA, R. 1987. Fundamentals of 3D migration. *Leading Edge* **6,** 22–30; **7,** 26–33.