GEOPHYSICAL TRANSACTIONS 1990
Vol. 35. No. 4. pp. 243–252

# THE USE OF SYMMETRY IN *f–k* MIGRATION

Einar MAELAND*

In order to speed up the classical *f–k* migration of zero-offset marine reflection data, a symmetrical data-set is artificially constructed, both in space and time. An efficient algorithm utilizes the discrete cosine transform, so only real variables are required. Since this yields a twofold decrease in computation time and storage requirements, no extra computer storage or working space other than the original data space is required. Moreover, since the discrete cosine transform effectively double the record length, computational artifacts caused by the discrete Fourier transform will be reduced.

**Keywords: seismic data processing, *f–k* migration, discrete cosine transform, algorithm, real variables, computational artifacts**

## 1. Introduction

The construction of an efficient algorithm for migration of zero-offset data is an important objective in seismic data processing. Migration by the classical *f–k* algorithm as given by STOLT [1978], is much faster than any other method, e. g., the phase-shift method [GAZDAG 1978], or the Kirchhoff summation [SCHNEIDER 1978]. In *f–k* migration, the spectrum is transformed from the frequency axis to the (vertical) wavenumber axis. To perform this (non-linear) mapping, some method of interpolation is required. Without any other information, both the real and the imaginary part of the *f–k* spectrum have to be interpolated. With respect to the amplitude and phase spectra, the interpolation errors of the phase spectrum (phase-errors) can be more troublesome than amplitude errors. An objective of the present study is to reduce the amount of this interpolation work.

Migration of zero-offset data is based upon the exploding reflector concept [LOEWENTHAL et al. 1976]. Based on this assumption, the data is equal to zero for time $t < 0$. If a function $f(t)$ is causal, i. e., $f(t) = 0$ for $t < 0$, the real and imaginary parts of the Fourier transform form a Hilbert transform pair [PAPOULIS 1977]. If, in addition, $f(t)$ is real, the real and imaginary parts of the Fourier transform are related to the cosine and sine transform of $f(t)$. If the real part of the Fourier transform is given, the imaginary part can in principle be found, and this latter part is redundant. Although zero-offset data are not "causal" in the horizontal space coordinates, it is possible (artificially) to construct symmetrical data without losing any information from the original data. As a consequence of this construction, the Fourier transform in wavenumber space will be real and even. Hence, it is possible to work entirely with a real spectrum, and many

*Seismological Observatory, University of Bergen, Allegt. 41, N–5007 Bergen, Norway
Manuscript received (revised version): 27 March, 1990

problems of computer storage and artifacts caused by interpolation of a complex spectrum can be avoided.

CARTER and FRAZER [1982] proposed a rapid method for $f$–$k$ migration of zero-offset data. They used the fact that the Fourier transform of any real function is hermitian (conjugate even), which means that the real part of the Fourier transform is symmetric, while the imaginary part is anti-symmetric. The consequences are that certain parts of the $f$–$k$ spectrum (negative frequencies) need not be stored in the memory of the host computer. Moreover, they also used a familiar trick such that in order to compute the Fourier transform of two real functions, $f(t)$ and $g(t)$, say, it is possible to compute the Fourier transform of the complex function $h(t) = f(t) + ig(t)$. The Fourier transform of $f(t)$ and $g(t)$ are then the hermitian (conjugate even) and the anti-hermitian (conjugate odd) part, respectively, of the Fourier transform of $h(t)$. However, if the input data are real and either even or odd, COOLEY et al. [1970] have shown that an even faster method exists to compute the discrete Fourier transform. Thus, the construction of symmetric zero-offset data in order to speed up the classical $f$–$k$ migration warrants a closer study.

## 2. The $f$–$k$ algorithm

In order not to overburden the present analysis with detail, only the 2-dimensional case will be studied. If the data (pressure) as a function of space ($x$) and time ($t$) are given by $P(x, t)$, imagine that a symmetric function $P(x, t)$ is constructed according to

$$P(-x, t) = P(x, -t) = P(-x, -t) = P(x, t). \tag{1}$$

Although this may at first sight seem to necessitate a much larger (four times) memory space than the original space, this is not the case. Let $x_n = n\Delta x$ and $t_m = m\Delta t$, where $\Delta x$ and $\Delta t$ are the (constant) sampling intervals along the $x$-axis and $t$-axis, respectively. A simple way to obtain the $f$–$k$ spectrum is to perform a fast Fourier transform (FFT) column-by-column, and put the transformed data back into the memory $P(x_n, \omega_m)$, where $\omega_m$ is the frequency, then a FFT row-by-row, and put the transformed data back into the memory $P(k_n, \omega_m)$ where $k_n$ is the wavenumber. This procedure presupposes a discrete Fourier transform of a working array $f_m = P(x_n, t_m)$, say. The procedure is then repeated in order to compute the discrete Fourier transform in the $x$-direction. Without any (symmetry) conditions of the input data, the spectrum will be complex, so extra memory space is required. However, if a symmetric working array ($f_m$) is constructed, then the Fourier transform becomes real and symmetric, and no extra memory in the host computer, beyond the original space $P(x_n, t_m)$, is required.

The Fourier series representation of any (periodic) real and symmetric function contains only real coefficients, which correspond to the cosine terms of the series. This result can be extended to the discrete Fourier transform. Consider a sequence $f_0(m)$ of length $2M-1$ (the odd-length symmetrical cosine transform), where $f_0(m) = f_m$ when $m \geq 0$, and $f_0(m) = f_{-m}$ when $m < 0$. The discrete Fourier transform of this sequence is

$$F_0(k) = \frac{1}{L} \sum_{1-M}^{M-1} f_0(m)\, exp[-2\pi i km / (2M-1)], \tag{2}$$

for $|k| \leq M-1$, where $L = 2M-1$ is the total length of the sequence. Since $f_0(m)$ is real and symmetric, this relation reduces to

$$F_0(k) = \frac{1}{L} \sum_{m=0}^{M-1} \tilde{f}_m\, cos[2\pi km/(2M-1)], \tag{3}$$

where $\tilde{f}_m$ is defined by $\tilde{f}_0 = f_0$ and $\tilde{f}_m = 2f_m$ for $1 \leq m \leq M-1$.

It is possible to compute the odd cosine transform with the discrete Fourier transform algorithm of odd length since

$$F_0(k) = \frac{1}{L} \,\mathbf{Real} \left\{ \sum_{m=0}^{M-1} \tilde{f}_m \exp[-2\pi i mk/(2M-1)] \right\}. \tag{4}$$

The same result can be obtained if the sequence $\tilde{f}_m$ is extended by $M$ zeros, viz., $\tilde{f}_m = 0$ for $m = M, M+1, \ldots, 2M-1$, and computing the DFT of length $2M-1$. This construction yields a causal sequence $f_m$ (by definition).

In any application of the discrete Fourier transform, it is necessary to make a distinction between $M$ even or $M$ odd. The frequency interval is $\Delta\omega = 2\pi/L\Delta t$, and if $M$ is odd, the Nyquist frequency $(\pi/\Delta t)$ is not attained by any $\omega_m = m\Delta\omega$, $|m| \leq M-1$. On the other hand, if $M$ is even, a sample $f_M$ at $t_M$ must be included, so that $L = 2M$. In this case, however, the Nyquist frequency is attained when $m = \pm M$. In order to treat this case correctly, a "one-half" weight at the very last sample is utilized, i. e., $\frac{1}{2} f_M$ at $m = \pm M$.

The most common way to compute the $2M$-length (real) discrete Fourier transform is either to use a $2M$-length complex FFT, or by using a $M$-length complex FFT plus some additional operations. Cooley et al. [1970] have shown that if the sequence (of length $2M$) is either even or odd, a simple procedure can be used to reduce the actual computation of the discrete Fourier transform to that of an $M/2$-length (complex) FFT with some preprocessing and postprocessing. An implementation of this procedure is given by Rabiner [1979]. This yields a twofold decrease in storage since only half the real input data need be supplied. More details on the efficient computation of the discrete cosine transform are given by Vetterli and Nussbaumer [1984]. Finally, an even more direct method is to construct a DFT that works directly on 2-dimensional sampled data, but this technique will not be discussed.

Computing the real spectrum with the decrease of storage requirements may be summarized as follows:

```
Do while n ≥ 0 until  n=N-1
      f(0) = P[x(n), t(0)]
      do while m≥1 until m=M-1
            f(m) = P[(x(n), t(m)]
            f(2M-1-m) = f(m)
      end do
      f(m) = DFT[f(m)]
      do while m≥0  until m=M-1
            P[(x(n), ω(m)] = f(m)
      end do
End do
```

The procedure is repeated in order to compute the discrete Fourier transform in the $x$-direction.

## 3. Interpolation

Given the $f$–$k$ spectrum, for each fixed value of the horizontal wavenumber ($k_x$), $f$–$k$ migration is essentially a coordinate transformation from the frequency axis ($\omega$) to the vertical wavenumber axis ($k_z$). In a two-dimensional study, this can be written $\omega / c \rightarrow k_z = \omega / c \cdot \cos(\alpha)$, where $c$ is the velocity and $\alpha$ is the angle between the vertical axis and the direction of the plane waves. To perform this coordinate transformation, a suitable interpolation algorithm has to be used. The algorithm should not only be fast and simple, but also of high resolution. CARTER and FRAZER [1982] used a linear interpolation scheme, but since the $f$–$k$ spectrum inevitably becomes periodic in any application of the discrete Fourier transform, it is more natural to use a periodic interpolation kernel. In this connection it is appropriate to note that for the construction of the odd symmetrical cosine transform, the addition of trailing zeros effects an interpolation of the spectrum. If $\Delta t$ is the constant sampling interval, samples are taken at $t_m = m\Delta t$ where m = 0, 1, 2,..., $M$–1. The Nyquist frequency is then given by $\omega_{\mathrm{Ny}} = \pi / \Delta t$. The discrete Fourier transform of a sequence of length $M$, say, yields a sampling interval in the frequency domain equal to $\Delta\omega = 2\pi / M\Delta t$. By application of the (odd symmetrical) cosine transform, the sampling interval is not changed, so the Nyquist frequency remains the same. However, since the record-length now (artificially) becomes $(2M-1)\Delta t$ the new sampling interval is $\Delta\tilde{\omega} = (2\pi / 2M-1)\Delta t$, or approximately half the original value. This very construction may make any further interpolation superfluous (nearest neighbour interpolation may in some cases be sufficient), but it may be more appropriate to interpolate in terms of cubic splines.

Interpolation by cubic splines is essentially by a low-pass action which incorporates some characteristics of sinc interpolation (Cardinal splines). For a fixed value of horizontal wavenumber, assume that $f_m$ is the data at time $t_m = m\Delta t$, and let the discrete Fourier transform of this sequence be denoted by $F_m$. Interpolation by cubic spline can be written

$$H(\omega) = \sum A_m B(\omega - m\Delta\omega), \tag{5}$$

where $B(\omega)$ is the cubic $B$-spline and $A_m$ are coefficients to be determined from the condition that $H(\omega_m) = F_m$. Among a variety of algorithms available in the literature, the algorithm given by FORD [1975] can be recommended, both for its simplicity and for its efficiency, but strictly speaking, the results are only approximately correct. In the present case it is possible to take advantage of the fact that the interpolation is carried out in the frequency domain. Thus, the results can be obtained with even less efforts, but the actual details are given in the Appendix.

The processing part of $f$–$k$ migration is a mapping from the $(k_x\,\omega)$-domain to the $(k_x, k_z)$-domain. Let $\Omega$ be defined by $\Omega / c = k_z$, where $c$ is the migration velocity. Assume that the $f$–$k$ spectrum is given at $k_n = n\Delta k$ and $\omega_m = m\Delta\omega$, where $\Delta k$ and $\Delta\omega$ are, respectively, the sampling interval in the wavenumber and frequency domain. Then for each $k_n$ and $\omega_m$ the values of the frequency $\Omega(k_n, \omega_m)$ are required. The mapping is governed by the equation

$$(\Omega / c)^2 = (m\Delta\omega / c)^2 - (n\Delta k)^2. \tag{6}$$

This transformation represents, for a fixed $k_n$, a shift of data from frequency $\omega_m$ to a lower frequency $\Omega$ [STOLT 1978]. It is important to achieve $\Omega = j\Delta\omega$, $j = 0, 1, \ldots, M$, hence, interpolation is necessary. However, for any value of $k_n$, some values of the original frequency $(\omega_m)$ may give an imaginary $\Omega$-value. But the $\Omega$ is supposed to be real, i. e., the evanescent part of the wave field is excluded. If an imaginary $\Omega$-value is obtained, the corresponding value of the spectrum is put equal to zero.

## 4. Illustrative examples

A comparison of the proposed algorithm with the conventional FFT-method [STOLT 1978] will be made. The input signal is a zero phase Ricker walvelet, i. e., the second derivative of the function $f(t) = \exp[-2(t/t_0)^2]$, with $t_0 = 0.05$ sec. Moreover, the velocity is $c = 1$ km/sec, while the record lengths are $X = 3$ km and $T = 2$ sec. Three "spikes" are present in the input data set, located at $x_A = 1.5$ km, but at different times $t_A = 0.75$, 1.0 and 1.25 sec, respectively. The migrated output with the conventional FFT-method is displayed in *Fig. 1*. The impulse response is ideally a semi-circle (the exploding reflector) centred at $z = 0$, with a radius equal to $ct_A = 0.75$, 1.0 and 1.25 km, respectively. The most conspicuous artifacts in the conventional FFT-method are the (inverted) semi-circles. Due to the periodicity of the discrete Fourier transform, the computational artifacts are (inverted) semi-circles of radii $r = c\,(T - t_A)$, centred at the bottom ($z = cT$). Other artifacts are also present (circles centred at $x_A \pm cT$), but they are hardly visible due to geometrical spreading. The corresponding results with the proposed algorithm are displayed in *Fig. 2*. This method effectively doubles the record length hence, due to geometrical spreading, the amplitude of the computational artifacts is reduced.

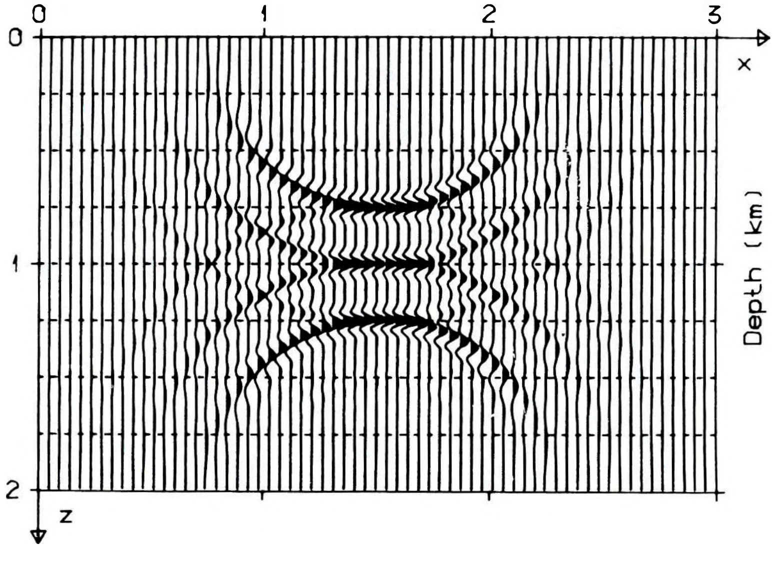


*Fig. 1.* Migrated output (impulse response) with the conventional FFT-method. The (inverted) semi-circles are the computational artifacts, which are strongly in evidence

*1. ábra.* A hagyományos FFT-t alkalmazó migráció impulzus válaszfüggvénye. Az (invertált) félkörök a számítás melléktermékei

*Рис. 1.* Резонансная импульсная функция миграции с традиционным ускоренным преобразованием Фурье.
(Обращенные) полукруги – побочный результат расчетных операций

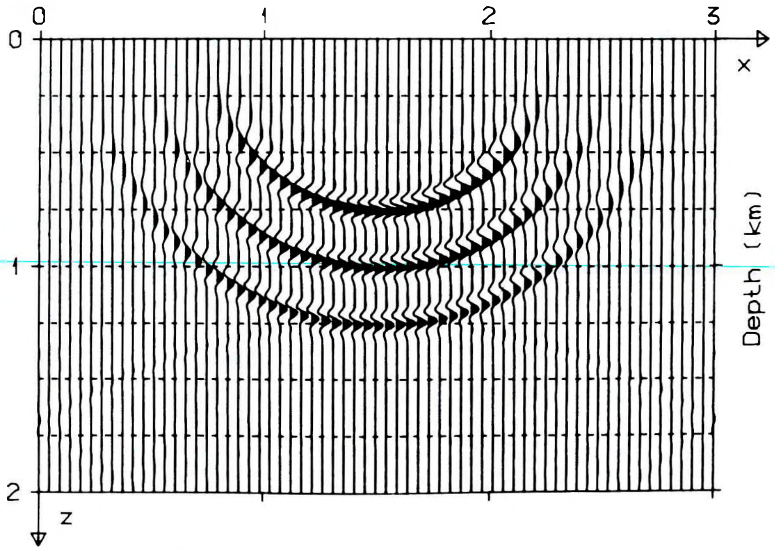


*Fig. 2.* Migrated output (impulse response) with the proposed method. Computational artifacts are still present, but the amplitudes are reduced due to geometrical spreading

*2. ábra.* A javasolt migrációval nyert impulzus válaszfüggvény. A számítási melléktermékek amplitúdói lényegesen csökkentek a szférikus divergencia következtében

*Рис. 2.* Резонансная импульсная функция, полученная при предлагаемом варианте миграции. Амплитуды побочных результатов расчетных операций существенно снизились вследствие сферической дивергенции

## 5. Conclusion

When migrating zero-offset reflection data for the first time, it is not so important to use a migration technique that is the best possible. Rather, a quick *f–k* migration can be used without running the risk of spending too much time looking for an exact velocity fit. The process of (artificially) constructing symmetric zero-offset data is used as an alternative to the classical *f–k* migration. The advantages are that it is possible to work entirely with real variables, hence, there is no need for extra working space to store the *f–k* spectrum in the host computer. The actual computation of the (real) *f–k* spectrum can be done by utilizing an efficient algorithm such as that of Cooley et al. [1970].

By application of the discrete Fourier transform, the migrated output will be periodic. Hence, computational artifacts will inevitably make their appearance. Since the impulse response is a semi-circle, the artifact will be (inverted) semi-circles. A method to reduce these artifacts is to use a longer record length in time (trailing the data set with zeros). The proposed method utilizes a symmetrical data set, which essentially incorporates some of this technique.

Consequently, it is possible to take advantage of even more symmetry properties than originally proposed by CARTER and FRAZER [1982].

## Appendix

*Periodic cubic B-spline*

The cubic $B$-spline is a polynomial approximation to a function $f(t)$, say, where the samples $f_m = f(m\Delta t)$ are given for $|m| \leq M-1$, where $\Delta t$ is the constant sampling interval. It will be assumed that the input sequence is periodic, i. e., $f_{m+L} = f_m$, where $L = 2M-1$.

Let a function $h(t)$ be constructed according to

$$h(t) = \sum_{1-M}^{M-1} a_m \, B(t-t_m), \tag{A-1}$$

where $B(t)$ is the $B$-spline or Parzen window, while $a_m$ are coefficients to be determined from the imposed condition that $h(t_m) = f_m$. Moreover, in order that $h(t)$ should be periodic, $h(t+L\Delta t) = h(t)$, the coefficients $a_m$ are forced to be periodic too, $a_{m+L} = a_m$, but it is not necessary to put any restriction on function $B(t)$. Returning to the $B$-spline form $B(t)$, this function is non-zero over 3 sample points, with $B(0) = 1$, $B(\pm\Delta t) = 1/4$, while $B(\pm m\Delta t) = 0$ when $m > 1$. The claim that $h(t_m) = f_m$ yields the key equations

$$a_{m-1} + 4\,a_m + a_{m+1} = 4\,f_m . \tag{A-2}$$

According to these equations the coefficients $a_m$ used to weight the spline functions are related to $f_m$ by a banded (tridiagonal) circulant matrix. The inversion of this matrix can be accomplished by using Fourier matrix techniques, i. e., the diagonalization property of circulant matrices. The inverse matrix is also circulant, but not banded. However, when $M$ is large, the asymptotic values of the coefficients in this (inverse) matrix are effectively independent of $M$. According to FORD [1975], the periodic nature of the problem allows it to be expanded (with no loss of accuracy) as if $M \to \infty$.

The construction of the coefficients $a_m$ warrants a closer study. It may be appropriate to study the discrete Fourier transform of equation (A–2), and the result is

$$A_k = 2F_k / [2 + \cos(2\pi k / L)], \tag{A-3}$$

where $A_k$ and $F_k$ are the discrete Fourier transforms of $a_m$ and $f_m$, respectively. The coefficients $a_m$ are then given by the inverse discrete Fourier transform

$$a_m = \sum_{1-M}^{M-1} A_k \cdot \exp(2\pi i k m / L). \quad \text{(A–4)}$$

If $f_m$ is an impulse at $m = 0$, then $F_k = 1$ for all values of the index $k$ (the impulse response). Let the corresponding discrete Fourier coefficients be denoted by $D_k$

$$D_k = 2 / [2 + \cos(2\pi k / L)], \quad \text{(A–5)}$$

which are real and even, i. e., $D_{-k} = D_{+k}$. In this way interpolation can be accomplished by the discrete Fourier transform and its inverse, respectively. The coefficients $a_m$ can be obtained by first taking the discrete Fourier transform $F_k = \mathrm{DFT}\{f_m\}$ multiplying by $D_k$, and finally taking an inverse DFT to obtain $a_m = \mathrm{IDFT}\{F_k \cdot D_k\}$

The processing part of *f–k* migration is essentially an interpolation on the frequency axis. The "simplest" way is to take a DFT in space to obtain $P(k_n, t_k)$, then a DFT in time to obtain $P(k_n, \omega_m)$. At this very step an interpolation as indicated by equation (A–1) must be performed. However, to obtain the coefficients $a_m$, imagine that $P(k_n, t_k)$ is multiplied by the filter coefficients $D_k$, followed by a DFT. Then the next step is to perform the convolution, which is nothing but an evaluation of a polynomial at the desired frequency values. The final step is to transform back to time and space coordinates to achieve the migrated output. This procedure may be considered as an alternative method to the algorithm given by FORD [1975].

## REFERENCES


CARTER J. A. and FRAZER L. N. 1982: Rapid *f–k* migration of zero-offset marine reflection data. Journal of Geophysical Research **87**, (B11), pp. 9365 – 9373

COOLEY J. W., LEWIS P. A. and WELCH P. D. 1970: The fast Fourier transform algorithm: Programming considerations in the calculating of sine, cosine and Laplace transforms. Journal of Sound and Vibrations **12**, pp. 315 – 337

FORD W. S. 1975: Periodic cubic spline interpolation with equidistant nodes. The Computer Journal **18**, pp. 183 – 184

GAZDAG J. 1978: Wave equation migration with the phase-shift method. Geophysics **43**, 7, pp. 1342 – 1351

LOEWENTHAL D., LU L., ROBERSON R. and SHERWOOD J. 1976: The wave equation applied to migration. Geophysical Prospecting **24**, 2, pp. 380 – 399

PAPOULIS A. 1977: Signal Analysis, McGraw-Hill Book Co.

RABINER L. R. 1979: On the Use of Symmetry in FFT Computations. IEEE Trans. Acoust., Speech, Signal Proc. **27**, pp. 233 – 239

SCHNEIDER W. A. 1978: Integral formulation for migration in two and three dimensions. Geophysics **43**, 1, pp. 49 – 76

STOLT R. H. 1978: Migration by Fourier Transform. Geophysics **43**, 1, pp. 23 – 48

VETTERLI M. and NUSSBAUMER H. J. 1984: Simple FFT and DCT algorithms with reduced number of operations. Signal Processing **6**, pp. 267 – 278

## AZ *f–k* MIGRÁCIÓ SZIMMETRIÁJÁNAK HASZNOSÍTÁSA

Einar MAELAND

Dinamikusan korrigált, tengeri reflexiós szeizmikus anyag hagyományos *f–k* migrációjának felgyorsítására térben és időben szimmetrikus adatrendszert hoznak létre. Egy hatékony algoritmust közölnek, amely a cosinus transzformációt használja, ezért csak valós változókra van szükség. Ez csökkentést jelent mind gépidőben, mind tárolási kapacitásban, így nincs szükség csak az eredeti adatok által foglalt tárolókapacitásra. Sőt, mivel a cosinus transzformáció megkétszerezi a rekordhosszt, a diszkrét Fourier-transzformáció okozta művi jelek amplitúdója nagymértékben csökken.

## УТИЛИЗАЦИЯ СИММЕТРИИ *f–k* МИГРАЦИИ

Эйнар МЕЙЛАНД

Для убыстрения традицинной *f–k* миграции материалов морской сейморазведки МОВ с динамической поправкой создается система данных, симметричная в пространстве и во времени. Приводится еффективный алгоритм, в котором используется косинусное преобразование, поэтому он нуждается лишь в реальных переменных. Это приводит к сокращению как машинного времени, так и емкости памяти, так что отпадает необходимость в памяти для первичных данных. Более того, посколъку при косинусном преобразовании удваивается длина записей, значительно уменьшается амплитуда искусственных сигналов, появляющихся при дискретом преобразовании Фурье.