Direct Fourier migration for vertical velocity variations

Gary F. Margrave

ABSTRACT

The Stolt f-k migration algorithm is a direct (i.e. non-recursive) Fourier-domain technique based on a change of variables (or equivalently, a mapping) that converts the input spectrum to the output spectrum. The algorithm is simple and efficient but limited to constant velocity. A v(z) f-k migration method, capable of very high accuracy for vertical velocity variations, can be formulated as a nonstationary combination filter that avoids the change of variables. The result is a direct Fourier-domain process that, for each wavenumber, applies a filter matrix to a vector of input frequency samples to create a vector of output frequency samples. The filter matrix is analytically specified in the mixed domain of input frequency and migrated time. It is moved to the domain of input frequency and output frequency by a fast Fourier transform. For constant velocity, the v(z) f-k algorithm recreates the Stolt method but without the usual artifacts related to complex-valued frequency domain interpolation. Though considerably slower than the Stolt method, vertical velocity variations, through an rms velocity (straight ray) assumption, are handled with no additional cost. Greater accuracy at slight additional expense is obtained by extending the method to a WKBJ phase shift integral. This has the same accuracy as recursive phase shift and can be made to handle turning waves in the same way.

Nonstationary filter theory allows the algorithm to be easily reformulated in other domains. The full Fourier domain method offers interesting conceptual parallels to Stolt's algorithm. However, unless a more efficient method of calculating the Fourier filter matrix can be found, the mixed-domain method will be faster. The mixed-domain nonstationary filter moves the input data from the Fourier domain to the migrated time domain as it migrates. It is faster because the migration filter is known analytically in the mixed domain.

INTRODUCTION

There are two major forms of migration based on Fourier transforms and many derivatives of these. Stolt (1978) presented a migration method, called f-k migration, that is "exact" for constant velocity. Stolt's method is called *direct* in that it constructs the migrated image directly from the unmigrated without intermediate products. Stolt also suggested an approximate extension to variable velocity but this has not proven popular because it is of low accuracy. Gazdag (1978) presented the other form, called phase-shift, and that is recursive rather than direct. The recursion is in the form of a loop over sampled depth. At each step the wavefield from the depth just above is downward extrapolated by phase shift. For constant velocity, phase shift is easily shown to be mathematically identical to f-k migration, though the latter is much quicker to compute. For v(z), phase shift is superior to f-k migration and is known to converge to a famous approximate wave equation solution called the WKBJ solution (Aki and Richards, 1980).

Contents

For lateral velocity variations, Gazdag and Squazzero (1984) gave the PSPI (phase-shift plus interpolation) extension of the phase shift method. Later, Margrave and Ferguson (1997) showed that PSPI can be written as a nonstationary filter and suggested an alternate generalization of phase shift called NSPS (nonstationary phase shift).

It is reasonable to ask why try to extend the Fourier methods to variable velocity when the Kirchhoff and finite difference methods handle this so naturally. There is an advantage of simplicity for the Fourier methods because they do not require explicit amplitude corrections for geometrical spreading or focussing. Instead, this is accomplished implicitly through the mechanics of the forward and inverse Fourier transforms. Conceptually, a Fourier transform performs a plane-wave decomposition and, since plane-waves propagate without spreading, they are easily extrapolated or mapped to new positions without amplitude adjustments. Essentially, this is an example of a physical effect that is more easily treated after a plane-wave decomposition that before. There are other such effects, for example anisotropic wave propagation.

So, while the spatial variation of velocity is somewhat difficult to handle with Fourier methods, there are other physical effects that are more easily handled. This has been a driving force behind the development of new mathematics that is able to deal simultaneously with spatial and Fourier domain effects. Examples are the wavelet transform (e.g. Ogden, 1997), the theory of pseudodifferential operators (e.g. St. Raymond, 1991), and the theory of nonstationary filters (e.g. Margrave, 1998).

This work shows how Stolt’s theory can be generalized to v(z) using the language of nonstationary filters. (This could equally well have been done through pseudodifferential operator theory with identical results.) The result is a direct Fourier migration, that is a true generalization of Stolt’s method, and that is a WKBJ theory for v(z). The theory is closely related (but distinct) to that used by Hale et al. (1992) for the migration of turning waves. Recently, Etgen (1998) has presented an alternative v(z) f-k migration that is quite different from the theory presented here.

This paper begins with a mathematical review of Stolt’s method that serves to establish notation and refresh the reader on the algorithm’s details. Then it is shown that the spectral mapping of f-k migration can be replaced by a nonstationary filter operation. For constant velocity, this nonstationary filter accomplishes the Stolt mapping without the usual interpolation artifacts. Then two different generalizations of the constant velocity nonstationary filter are explored: an rms assumption and a first-order WKBJ method. The v(z) f-k method is formulated as an operation on a vector of frequencies for fixed wavenumber. Such a frequency vector is migrated by multiplying it by a matrix called the migration filter. The v(z) f-k method is shown to generalize f-k migration and also to unify it with v(z) phase-shift.

A series of example impulse responses are shown for the v(z) algorithm, Stolt’s f-k migration, Gazdag’s phase shift, and a Kirchhoff method. Migration filters are also shown for several cases. Finally, computational cost considerations are presented. Stolt’s method is by far the fastest, but restricted to constant velocity. The mixed-domain form of v(z) f-k is shown to have the same computation effort as recursive phase shift. Finally, the full-Fourier v(z) f-k method is shown to be the slowest but could become much faster if further research is successful.

REVIEW OF CONSTANT VELOCITY FOURIER MIGRATION

This theory is presented for 2-D zero-offset geometry for simplicity of exposition. Let $\Psi(x,t)$ represent a zero-offset seismic wavefield recording. The exploding reflector model allows $\Psi(x,t)$ to be considered as the boundary value of a wavefield, $\Psi(x,z,t)$, measured at the z=0 recording plane. Thus $\Psi(x,t)$ is more properly written as $\Psi(x,0,t)$ where $\Psi(x,z,t)$ is a solution to the 2-D scalar wave equation

$$\frac{\partial^2}{\partial x^2}\Psi + \frac{\partial^2}{\partial z^2}\Psi = \frac{1}{v^2}\frac{\partial^2}{\partial t^2}\Psi \tag{1}$$

The migration problem proposes that $\Psi(x,z,0)$ be found given Ψ is a solution to equation (1) and that $\Psi(x,0,t)$ prescribes the wavefield at z=0. Since equation (1) is a second order hyperbolic PDE (partial differential equation), its unambiguous solution requires both Ψ and $\partial\Psi/\partial z$ be prescribed on the boundary. Thus the migration problem is posed with inadequate boundary conditions and requires additional assumptions for a solution. Stolt (1978) first showed how such a solution can be obtained under the assumptions that $\Psi(x,z,t)$ contain only upward traveling waves. Since Stolt developed his solution using Fourier techniques, he also made the assumption that v is a constant.

The development of Stolt's solution requires the Fourier transform of equation (1). For this purpose, $\Psi(x,z,t)$ can be represented as the inverse Fourier transform of its (ω,k_x) spectrum as

$$\Psi(x,z,t) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \varphi(k_x,z,\omega)\exp\left(i\left(\omega t - k_x x\right)\right)dk_x d\omega \tag{2}$$

where

$$\varphi(k_x,z,\omega) = \iint_{-\infty}^{\infty} \Psi(x,z,t)\exp\left(-i\left(\omega t - k_x x\right)\right)dxdt \quad . \tag{3}$$

Substitution of equation (2) into equation (1) leads to

$$\iint_{-\infty}^{\infty}\left[\frac{\partial^2}{\partial z^2}\varphi(k_x,z,\omega) - \left(\frac{\omega^2}{v^2} - k_x^2\right)\varphi(k_x,z,\omega)\right]\exp\left(i\left(\omega t - k_x x\right)\right)dk_x d\omega = 0 \quad . \tag{4}$$

The satisfaction of equation (4) for all k_x and ω can only be done if the term in brackets vanishes identically. That is

$$\frac{\partial^2}{\partial z^2}\varphi(k_x,z,\omega) = k_z^2\varphi(k_x,z,\omega) \tag{5}$$

where

Contents

$$k_z^2 = \frac{\omega^2}{v^2} - k_x^2 \ . \tag{6}$$

The general solution to equation (5) can be verified by direct substitution to be

$$\varphi\left(k_x,z,\omega\right) = A\left(k_x,\omega\right)\exp\left(ik_z z\right) + B\left(k_x,\omega\right)\exp\left(-ik_z z\right) \tag{7}$$

where A and B are the (k_x,ω) spectra of upward and downward traveling waves respectively (z is assumed to increase downward). As mentioned previously, the determination of both A and B requires knowledge of both Ψ and $\partial\Psi/\partial z$ at z=0. In the present context, B is assumed zero and $A(k_x,\omega)$ is set equal to $\varphi(k_x,0,\omega)$. Thus the spectrum of the *exploding reflector wavefield* can be constructed as

$$\varphi\left(k_x,z,\omega\right) = \varphi\left(k_x,0,\omega\right)\exp\left(ik_z z\right) \tag{8}$$

while the time domain result is the inverse Fourier transform of equation (8) and is

$$\Psi\left(x,z,t\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \varphi\left(k_x,0,\omega\right)\exp\left(ik_z z\right)\exp\left(i\left(\omega t - k_x x\right)\right)dk_x d\omega \tag{9}$$

In the exploding reflector formalism, the migrated section is $\Psi(x,z,0)$ which is obtained from equation (9) as

$$\Psi\left(x,z,0\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \varphi\left(k_x,0,\omega\right)\exp\left(ik_z z\right)\exp\left(-ik_x x\right)dk_x d\omega \ . \tag{10}$$

Equation (10) gives a migrated depth section. A migrated time section is obtained by letting z=vt and defining $\eta = vk_z$. This results in

$$\Psi\left(x,\tau,0\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \varphi\left(k_x,0,\omega\right)\exp\left(i\eta\tau\right)\exp\left(-ik_x x\right)dk_x d\omega \ . \tag{11}$$

Though equation (11) is complete as it stands, its numerical implementation is hindered by the fact that the ω integration is not a Fourier transform. Stolt addressed this by a change of integration variable

$$\omega\left(\eta,k_x\right) = \sqrt{\eta^2 + v^2k_x^2} \ \text{ or } \ \eta\left(\omega,k_x\right) = \sqrt{\omega^2 - v^2k_x^2} \tag{12}$$

where the positive square root is chosen in either equivalent expression. The formal change of variables in equation (11) yields

$$\Psi\left(x,\tau,0\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \theta\left(k_x,\eta\right)\exp\left(i\eta\tau\right)\exp\left(-ik_x x\right)dk_x d\eta \tag{13}$$

where

$$\theta\left(k_x,\eta\right) = \frac{\eta}{\sqrt{\eta^2 + v^2 k_x^2}}\varphi\left(k_x,0,\sqrt{\eta^2 + v^2 k_x^2}\right) \ . \tag{14}$$

Equations (13) and (14) express the complete f-k (i.e. ω-k_x) algorithm developed by Stolt for constant velocity, zero-offset, migration. In summary, the input data (usually a cmp stack) is Fourier transformed over x and t to obtain $\varphi(k_x,0,\omega)$, $\theta(k_x,\eta)$ is constructed from equation (14), and the result is inverse Fourier transformed as in equation (13). The construction of $\theta(k_x,\eta)$ is commonly called a mapping process and is the creation of the spectrum of $\Psi(x,z,0)$ directly from the spectrum of $\Psi(x,0,t)$. This process is illustrated in Figures 1 and 2. (In Figure 2, as in all similar figures in this paper, solid black represents the largest absolute value number in the data matrix, pure white is the negative of the number which is solid black, and neutral gray is zero.)

REFORMULATION AS A NONSTATIONARY FILTER

An alternative to the Stolt mapping is to consider equation (11) as a nonstationary filter. This can be done quite simply as

$$\Psi\left(x,\tau,0\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \varphi\left(k_x,0,\omega\right)\exp\left(i\tau(\eta - \omega)\right)\exp\left(i(\omega\tau - k_x x)\right)dk_x d\omega \tag{15}$$

or

$$\Psi\left(x,\tau,0\right) = \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \varphi\left(k_x,0,\omega\right)\alpha\left(\omega,\tau,k_x\right)\exp\left(i(\omega\tau - k_x x)\right)dk_x d\omega \tag{16}$$

where

$$\alpha\left(\omega,\tau,k_x\right) = \exp\left(i\tau(\eta - \omega)\right) = \exp\left(i\tau\left(\sqrt{\omega^2 - v^2 k_x^2} - \omega\right)\right) = \exp\left(i\omega\tau\left(\sqrt{1 - p^2 v^2} - 1\right)\right) \tag{17}$$

is the nonstationary filter transfer function. The related function $m(\omega,\tau,k_x)=\alpha(\omega,\tau,k_x)\exp(+i\omega\tau)=\exp(i\tau[\omega^2-v^2k_x^2]^{1/2})$ (see equation 31) will be called *the migration filter*. Equation (17) presents several equivalent expressions for $\alpha(\omega,\tau,k_x)$ [or, implicitly, $m(\omega,\tau,k_x)$] and the last uses the ray parameter $p = k_x/\omega$. Equation (16) achieves a constant velocity migration by applying the migration filter simultaneously with the inverse Fourier transform from ω to τ. Note that the filter is nonstationary only in ω and remains stationary in k_x. (A theory that encompassed lateral velocity variations would cause further nonstationarity in k_x.)

Though equation (16) gives a complete migration, it does so while transforming the data from frequency to time. A more direct parallel with Stolt's method requires moving the method entirely into the Fourier domain. Since the k_x integral in equation (16) is an ordinary inverse Fourier transform, it can be dropped for now to concentrate on the nonstationary ω integral. Thus, consider

Contents

$$\Psi\left(k_x,\tau,0\right)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\varphi\left(k_x,0,\omega\right)\alpha\left(\omega,\tau,k_x\right)\exp\left(i\omega\tau\right)d\omega\ . \tag{18}$$

According nonstationary filter theory (Margrave, 1998), equation (18) is a nonstationary combination filter expressed in the mixed domain. It can be moved entirely into the Fourier domain by the forward Fourier transform of the τ dependence of $\alpha(\omega,\tau,k_x)$. The result is an operation on the spectrum of the unmigrated data to yield the spectrum of the migrated data, and is written

$$\theta\left(k_x,\eta\right)=\int_{-\infty}^{\infty}\varphi\left(k_x,0,\omega\right)A\left(\omega,\eta-\omega,k_x\right)d\omega \tag{19}$$

where

$$A\left(\omega_1,\omega_2,k_x\right)=\int_{-\infty}^{\infty}\alpha\left(\omega_1,\tau,k_x\right)\exp\left(-i\omega_2\tau\right)d\tau\ . \tag{20}$$

In equation (20) ω_1 and ω_2 are nominal frequencies are to be substituted with ω and $\eta-\omega$ respectively in equation (19). Performing that substitution leads to

$$M\left(\omega,\eta,k_x\right)\equiv A\left(\omega,\eta-\omega,k_x\right)=\int_{-\infty}^{\infty}\exp\left(i\omega\tau\sqrt{1-v^2k_x^2\omega^{-2}}-i\eta\tau\right)d\tau \tag{21}$$

where $M(\omega,\eta)$ is the integration kernel, or *migration filter in the Fourier domain,* required in equation (19). M is the Fourier transform of the migration filter $m(\omega,\tau,k_x)=\exp(i\tau[\omega^2-v^2k_x^{\ 2}]^{1/2})$ defined previously. Thus

$$\theta\left(k_x,\eta\right)=\int_{-\infty}^{\infty}\varphi\left(k_x,0,\omega\right)M\left(\omega,\eta,k_x\right)d\omega \tag{22}$$

According to nonstationary filter theory, $\theta(k_x,\eta)$ as given by equation (22), or equivalently equation (19), must be identical to that given by equation (14). Once equation (22) is evaluated, the migrated section is obtained by equation (13) as before. The discrete equivalent of equation 22 is a matrix-vector multiplication as shown in Figure 4. The advantage of the nonstationary formulation over the Stolt mapping is that the former can be easily extended to vertically variable velocity as will be shown in the next section.

The remainder of this section will show precisely how equation (22) is equivalent to equation (14) for constant velocity. First, note that $M(\omega,\eta,k_x)$, as given by equation (21), is a Dirac delta function

$$M\left(\omega,\eta,k_x\right)=\delta\left(\sqrt{\omega^2-v^2k_x^2}-\eta\right) \tag{23}$$

Insertion of equation (23) into (22) gives

$$\theta\left(k_x,\eta\right)=\int_{-\infty}^{\infty}\varphi\left(k_x,0,\omega\right)\delta\left(\sqrt{\omega^2-v^2k_x^2}-\eta\right)d\omega\ . \tag{24}$$

The integral in equation (24) can be evaluated using the property of the Dirac delta function that (Riley et al. 1997, p351)

$$\delta\left(g(\omega)\right)=\sum_{\text{all m}}\frac{1}{\left|g'(\omega_m)\right|}\delta\left(\omega-\omega_m\right);\qquad g(\omega_m)=0 \tag{25}$$

In equation (25) $g'(\omega)$ is the derivative of $g(\omega)$, ω_m denotes a zero of $g(\omega)$ and the delta function requires a sum over all such zeros. In this case, $g(\omega)$ has two zeros which are determined as follows

$$g(\omega_m)=\sqrt{\omega_m^2-v^2k_x^2}-\eta=0 \tag{26}$$

from which it results

$$\omega_m=\pm\sqrt{\eta^2+v^2k_x^2} \tag{27}$$

and furthermore

$$g'\left(\omega_m\right)=\frac{\omega_m}{\sqrt{\omega_m^2-v^2k_x^2}}=\frac{\pm\sqrt{\eta^2+v^2k_x^2}}{\eta}\ . \tag{28}$$

For the consideration of upward traveling waves only, we choose the + sign in equation (28) which means that the sum in equation (25) has only one term. This is then substituted into equation (24) to give:

$$\theta\left(k_x,\eta\right)=\int_{-\infty}^{\infty}\varphi\left(k_x,0,\omega\right)\frac{\eta}{\sqrt{\eta^2+v^2k_x^2}}\delta\left(\omega-\sqrt{\eta^2+v^2k_x^2}\right)d\omega\ . \tag{29}$$

The delta function collapses this integral immediately to

$$\theta\left(k_x,\eta\right)=\frac{\eta}{\sqrt{\eta^2+v^2k_x^2}}\varphi\left(k_x,0,\sqrt{\eta^2+v^2k_x^2}\right) \tag{30}$$

which is equation (14). This demonstrates that equation (22) does exactly recreate the Stolt mapping and is therefore a complete reformulation of constant velocity f-k migration.

EXTENSION TO V(Z)

Two different extensions of the nonstationary filter formulation to vertical velocity variations will be explored. Though the term v(z) appears in section and paper titles, the actual implementation here is $v(\tau)$. Of course, in the absence of lateral velocity variations, the distinction is purely one of convenience since any migrated time section can be converted to a migrated depth section (and vice-versa) by simple 1-D trace “stretching”. The two extensions are conceptually analogous to either straight raytracing with v_{rms} or curved raytracing using Snell’s law. Not surprisingly, the former has computational advantages while the latter is more accurate.

That the nonstationary filter formulation can be extended to vertical velocity variations should be apparent from the following. The impulse response of constant velocity migration is a wavefront circle (in depth) whose center is placed on the surface at a sample’s x coordinate and whose radius is the sample’s depth. Thus, migration can be visualized as a process of replacing each input sample by a circle whose radius is varying with z. Though not a stationary convolution since the replacement function varies, this is a nonstationary filter operation. Since the formalism established above already accounts for the vertical variation of the wavefront circle due to depth, it should be possible to include vertical variation due to other effects such as velocity.

As mentioned above, Equation (21) can be interpreted as the forward Fourier transform, from τ to η, of the migration filter $m(\omega,\tau,k_x)$ given by

$$m\left(\omega,\tau,k_x\right)=\exp\left(i\omega\tau\sqrt{1-p^2v^2}\right) \tag{31}$$

so that

$$M\left(\omega,\eta,k_x\right)=\int_{-\infty}^{\infty} m\left(\omega,\tau,k_x\right)\exp\left(-i\eta\tau\right)d\tau \tag{32}$$

The v_{rms} extension is accomplished simply by altering $m(\omega,\tau,k_x)$ to

$$m_{rms}\left(\omega,\tau,k_x\right)=\exp\left(i\omega\tau\sqrt{1-p^2v(\tau)^2_{rms}}\right)\ . \tag{33}$$

where $v(\tau)^2_{rms}$ refers to the root-mean-square velocity as a function of vertical time. As this expression shows, the phase of the migration filter is no longer simply linear in τ but has much more complicated τ dependence through the variation of $v_{rms}(\tau)$. Nevertheless the τ dependence can still be simply Fourier transformed (equation 32) and the migrated spectrum then computed through equation (22).

The second alternative extension follows from the WKBJ solution to the scalar wave equation in a stratified medium (Aki and Richards, 1980, p416). Equation (8) can be modified to be an initial downward extrapolation step from 0 to dz using the local velocity at v(dz). If this process is applied recursively to span a finite interval from 0 to z, the phase shifts for each layer add. In the limit of infinitesimal steps, the result is an integral expression for the phase. When this result is followed through the filter development, the result is

$$m_{wkbj}\left(\omega,\tau,k_x\right) = \exp\left(i\omega\int_0^\tau \sqrt{1-p^2v^2(u)}\,du\right) \qquad (34)$$

Here the velocity used is the local wavespeed (instantaneous velocity) and the result is a migration filter with the same accuracy as the phase shift method of Gazdag (1978). It has an advantage over the phase shift method in that it is still applied through equation (22) as a direct method.

The WKBJ solution given by equation (34) is considerably more accurate at higher p values than the rms solution of equation (33). In fact, equation (33) is readily derived from equation (34) by appeal to the mean-value theorem of calculus. This theorem asserts the phase integral in equation (34), which is the area under the curve $(1-p^2v^2(u))^{1/2}$ from 0 to τ, can be approximated by an equivalent rectangle whose height is given by $(1-p^2\underline{v}^2)^{1/2}$ where $\underline{v}$ is some sort of average velocity over the macroscopic interval 0 to τ. Thus, $\underline{v}$ can be found from the condition

$$\sqrt{1-p^2\underline{v}(\tau)^2} = \int_0^\tau \sqrt{1-p^2v^2(u)}\,du \ . \qquad (35)$$

When both square roots are expanded to first order in p^2, $\underline{v}$ can be easily solved for and found to be the rms velocity. Thus, equation (33) is an approximation to equation (34) that is valid for small p^2.

The WKBJ solution implemented here is called a first-order solution and handles the phase (position and geometric spreading) correctly for all angles. Higher order solutions are known which incorporate approximate amplitude terms for transmission losses. The second order WKBJ correction multiplies equation (34) by a factor of $(\omega^2v^{-2}(\tau)-k_x^2)^{-1/2}$ which is just $k_z^{-1/2}$. Though this is a reasonable amplitude correction for low vertical angles it is infinite at 90° and therefore unsuitable for a migration algorithm. A reasonable amplitude correction will have to be a higher order correction than this.

IMPLEMENTATION AND EXAMPLES

Stolt's f-k migration algorithm, the v(z) f-k algorithm of this paper, a recursive Gazdag phase-shift, and a simplified Kirchhoff algorithm were all implemented digitally in Matlab to produce the results shown here. These will be referred to as f-k migration, v(z) f-k migration, phase-shift and Kirchhoff migration respectively. The v(z) f-k algorithm was written expressly for this study, while other three are teaching codes that have been in use at the University of Calgary for several years. All are 2-D post-stack (zero offset) implementations. Matlab was chosen for its programming simplicity, excellent graphics, and ability to count floating point operations and cpu-time.

The f-k migration is done in the conventional way with a forward f-k transform, a spectral mapping and scaling (Figures 1 and 2, equation (14)), and an inverse f-k transform. The spectral mapping "unfolds" the triangular non-evanescent portion of the unmigrated spectrum into the circular migrated spectrum. A particular output sample of the migrated spectrum, at (k_x,η), comes from the unmigrated spectrum at (k_x, ω) where

Contents

ω is given by equation (12). Generally, ω will not fall exactly on the input (k_x, ω) grid and a complex-valued interpolation will be required. This interpolation is non-trivial and is a major source of artifacts in f-k migrations (Claerbout, 1985). This study used an eight-point, complex-valued, optimized "sinc function" interpolation based on Claerbout's suggestions. The major computation loop is over output wavenumber and all other operations, including the interpolation, have been vectorized. The result is a very fast code with controlled, though still present, artifacts.

The phase shift migration begins with an f-k transform of the stack and then enters a loop over output time (or depth) samples. (Both this and the f-k migration have options for zero padding in space and time prior to the forward f-k transform to minimize Fourier wraparound.) At each step in the loop a phase shift of $\Delta z(\omega^2/v^2(z) - k_x{}^2)^{-1/2}$ is applied to the spectrum (which extrapolates it downward) and the result is evaluated at t=0. This last step creates a single depth slice of the output in (k_x,z) space. No spectral interpolation is necessary and the result is accurate for all dips. As mentioned in the previous section, in the limit as Δz approaches zero, this method converges to the WKBJ solution.

The Kirchhoff implementation is a simple "NMO removal and sum" process that omits the "details" of dip-limitation, anti-alias filtering, and time differentiation, but does respond to vertical velocity variations. The interpolation required for NMO removal is simple nearest-neighbor. The major computation loop is over output trace location and a second inner loop is used to NMO remove and sum for each output trace. The resulting code is much slower than f-k migration but satisfactory for comparison with v(z) f-k migration.

The v(z) f-k migration was implemented with a main loop over output wavenumber. (Like f-k migration, a forward or inverse f-k transform begins or completes the algorithm.) For each output wavenumber, a vector of input frequency samples is filtered with a nonstationary filter as in equation (22) to give a vector of output frequency samples. This is done as a matrix-vector multiply (Figure 4) by building a matrix representation of $M(\omega,\eta,k_x)$. This matrix is computed by first building a matrix representing $m(\omega,\tau,k_x)$ and then applying an FFT to transform τ to η (Figure 3). Either of equations (31), (33), or (34) can be used to build $m(\omega,\tau,k_x)$ depending upon what sort of velocity adaptivity is desired. Note that since $m(\omega,\tau,k_x)$ depends on the square of k_x, unique filters need only be constructed for half of the wavenumbers. Two other simple optimizations: filtering only up to a desired maximum frequency and zeroing evanescent spectral components, were also done.

In the examples of the concepts and algorithms presented here, the migration of a single live trace is used to clearly display the impulse response as a function of time. In all examples, the input dataset had 512 traces with 256 samples each with only trace 256 being non-zero. (The f-k and phase shift migrations used a temporal zero pad that doubled the trace length.) Spatial and temporal sampling intervals were 12m and 8ms respectively. In both migrations, only frequencies less than 70% of Nyquist (~44 Hz) were migrated. The single live trace was constructed as a series of regularly spaced unit impulses convolved with an "Ormsby" wavelet whose maximum frequency was about

35 Hz. All migrations were performed on a Power Macintosh 7300/200 using Matlab 5.1 in a 45 mb memory partition.

In Figure 5, constant velocity migrations for a velocity of 2000 m/s from the v(z) f-k algorithm (a) and Stolt's method (c) are compared. Figures 5b and 5d show the f-k_x transforms of these same two results. Though largely similar, the v(z) f-k migration shows no evidence of the artifacts seen near .2 seconds in the f-k migration. The lack of significant artifacts is a major benefit of the v(z) f-k algorithm. However, this is purchased at a considerable increase in CPU time. For this case, the f-k algorithm ran in about 38 seconds while v(z) f-k required 520 seconds. (The time required for the v(z) algorithm is independent of velocity and, as will be seen, any vertical velocity variation can be accommodated with little increase in compute time.) According to Matlab, the f-k migration required about $50\text{x}10^6$ floating point operations while the v(z) f-k migration required $2300\ \text{x}10^6$.

The reason for the reduction in artifacts can be appreciated by examining a typical matrix representing $M(\omega,\eta,k_x)$ as shown in Figure 6. Figure 6a shows the amplitude spectrum ($|M(\omega,\eta,k_x)|$) for k_x = .01 m^{-1} while 4b shows the real and imaginary parts of the same complex matrix. (As previously mentioned, all gray-level figures in this paper use solid black to represent the largest absolute value number in the data matrix, pure white as the negative of the number that is solid black, and neutral gray as zero.) As discussed previously, in the constant velocity, continuous case, $M(\omega,\eta,k_x)$ is a Dirac delta function (equation 23) whose singularity tracks along the curve $\eta = (\omega^2 - v^2k_x{}^2)^{1/2}$, and as Figure 4a shows, the discrete representation of $M(\omega,\eta,k_x)$ does have maxima along this track. The oscillatory behavior of $M(\omega,\eta,k_x)$ away from the ideal singularity curve is precisely the complex interpolator required for the Stolt mapping. Essentially, each row of $M(\omega,\eta,k_x)$ contains an optimal complex-valued interpolator that is the natural consequence of the Fourier transform of $m(\omega,\tau,k_x)$. Thus the v(z) f-k algorithm uses a full-length interpolator optimized for each individual frequency. At values of ω less than the point where the singularity intersects the η=0 axis the filter is evanescent.

As a second example, v(z) f-k, using the WKBJ migration filter and the phase-shift method discussed above are examined for the case of a linear-with-depth instantaneous velocity $v(z) = v_o + cz$ where v_o=1430 m/s and c=.6sec^{-1}. Figure 7 shows the v(z) f-k result in (a), its f-k spectrum in (b), the phase-shift result in (c) and its f-k spectrum in (d). The time-domain impulse responses are largely similar with apparently identical trajectories up to 90° as marked by the vertical li ne. Neither algorithm was explicitly designed for anything beyond 90° so the shape of the phase-shift trajectories past this point is an algorithm quirk. The f-k spectra are also nearly identical. It is interesting to recall that the input to this computation was the same as for Figure 5 which emphasizes that the f-k spectra were computed with something more than a simple mapping technique. The v(z) f-k algorithm had slightly greater CPU time and operations count than the previous example (550 seconds and $2600\ \text{x}10^6$ operations) while the phase shift method took 980 seconds and $1800\ \text{x}10^6$ floating point operations. The operations counts were similar and it seems likely that much of the difference in run times is a function of how well the algorithms vectorize in Matlab.

An example of the Fourier domain migration filter, $M(\omega,\eta,k_x)$, for the WKBJ case is shown in Figure 8. As with Figure 6, the amplitude spectrum is in (a) while the real and imaginary parts are in (b) and (c). It is immediately apparent that this matrix is no longer the discrete equivalent to a Dirac delta function. However, it still has its dominant energy restricted to a well-defined portion of the matrix. Shown on Figure 8a are three trajectories of the form $\eta = (\omega^2 - v^2k_x{}^2)^{1/2}$ where the leftmost uses the minimum instantaneous velocity, the middle uses the mean instantaneous velocity and the right uses the maximum instantaneous velocity. The bulk of the spectral power of the matrix lies between the minimum and maximum velocity curves.

A third and final example is shown in Figure 9 where the v(z) f-k algorithm is shown using the rms migration filter and compared to the Kirchhoff algorithm using the same rms velocities. The instantaneous velocity function was the same as for Figure 7. Figure 9a shows the v(z) f-k migration while the Kirchhoff migration is shown in Figure 9c. As before, the v(z) f-k algorithm made no attempt to handle scattering angles (i.e. *dips*) beyond 90° so that the impulse response curves end at that angle. On the other hand, the Kirchhoff algorithm naturally handles such angles so that the curves extend well past 90°. The rms assumption made here means that these curves are inaccurate at steep angles and the vertical line, which is in the same position as for Figure 7, allows an easy comparison with that figure. The Kirchhoff algorithm is not fully correct for the amplitudes of steeper dips. The compute time for v(z) f-k was the same as the constant velocity case 520 seconds as before while the Kirchhoff algorithm required 1250 seconds. The operations counts were the same 2300 $x10^6$ for v(z) f-k and 1500 $x10^6$ for Kirchhoff. Again it seems likely that differences in vectorizing in Matlab accounts for the longer run time for Kirchhoff.

The f-k spectra of the two rms results are in Figure 9b and 9d. This is a good time to re-emphasize the fundamental differences between these algorithms. Both are *direct* algorithms, meaning that there is no recursive downward continuation involved, but the Kirchhoff computation is direct in the space-time domain while v(z) f-k is direct in the Fourier domain. Thus v(z) f-k computes the spectrum of Figure 9b directly from the input spectrum while Kirchhoff computes the wavefield of Figure 9c directly from the input wavefield. The lower power at steep dips is evident in the Kirchhoff spectrum at higher wavenumbers.

Figure 10 shows a typical $M(\omega,\eta,k_x)$ matrix for the rms migration filter. As before, the amplitude spectrum is in (a) while the real and imaginary parts are in (b) and (c). Comparison with the more accurate WKBJ result in Figure 8 shows broad similarity but the rms filter loses power near the trajectory determined by the maximum instantaneous velocity.

Examples of the three different migration filters have been shown in Figures 6, 8, and 10. As a final comparison of these filters, Figure 11 shows profiles across the amplitude spectrum of each at η=20 Hz. The near-singularity of the constant velocity filter is very evident. Also, the rms filter approximates the WKBJ result well at lower frequencies but makes systematic errors at higher frequencies. For nonevanescent frequencies, there is a one-one correspondence between ω and velocity through the relation $\omega = (\eta^2 + v^2k_x{}^2)^{1/2}$. Numerical experimentation has shown that the power in the WKBJ filter at a particular frequency is related to how significant the corresponding

Contents

velocity is. If a particular constant velocity applies over a large τ range, then the WKBJ filter will be strongly peaked at the corresponding frequency. In this case, the instantaneous velocity model was linear with depth and that means exponential with time.

COMPUTATIONAL COST CONSIDERATIONS

The v(z) f-k algorithm can also be run as a mixed-domain nonstationary filter. This amounts to a direct implementation of equation (18) and produces identical migrations to those in Figures 5, 7, and 9. (Though they did not formulate it this way, Hale et al. (1993) used the mixed-domain method). The mixed-domain filter is actually considerably faster than the full-Fourier domain method (as implemented here) because the cost of moving the migration filter into the Fourier domain is saved. This is because the full-Fourier method was done with a matrix-vector multiply while the mixed-domain method also requires a similar matrix-vector operation. Thus the cost of moving the migration filter between domains is an additional burden for the full-Fourier method. Table 1 summarizes the algorithm testing done here and includes results for the mixed-domain filter.

Algorithm	**floating pt. ops. (x10^6)**	**run time (seconds)**
f-k migration (Stolt)	50	38
v(z) f-k (full-Fourier)	2300(rms) 2600(WKBJ)	520 (rms) 550(WKBJ)
v(z) f-k (mixed-domain)	600(rms) 1000(WKBJ)	460 (rms) 520(WKBJ)
Phase-shift	1800	980
Kirchhoff	1500	2000

Table 1. The floating-point operations count and CPU times for the algorithms discussed in the text as reported by Matlab for a Macintosh 7300/200. The operations counts are probably proportional to realistic estimates of run times in other computing environments. The fact that the full-Fourier v(z) f-k has twice as many operations but takes only marginally more time is probably due to Matlab's efficient FFT algorithm doing most of the work.

Despite its cost, the full-Fourier v(z) f-k method is of interest for several reasons. First, from a theoretical perspective, it is interesting to see how to generalized Stolt's algorithm to v(z). The mixed-domain method, while mathematically equivalent, does not make this connection obvious. Second, the full-Fourier domain method could actually become the faster method if a reasonable analytic approximation for $M(\omega,\eta,k_x)$ could be found. Even without that, considerable speedup would occur if the application of $M(\omega,\eta,k_x)$ were done without a full matrix-vector multiply. That is, could the large, low-amplitude areas of $M(\omega,\eta,k_x)$ seen in Figures 8 and 10 be neglected in the filter application? Even better, could a fast numerical evaluation of $M(\omega,\eta,k_x)$ avoid calculating them in the first place? These questions pose interesting research problems.

Operations count estimates for the four Fourier methods discussed in this paper are summarized in Table 2. This table assumes a 2-D dataset of dimension $N^{1/2}$ by $N^{1/2}$ for a total of N floating point numbers. The fastest algorithm is clearly f-k migration that requires two f-k transforms of NlogN each and the Stolt mapping in between. Assuming all N points must be mapped and that an 8 point interpolator is used, this is estimated at 8N. The largest term for the Stolt algorithm is 2NlogN. Phase-shift requires a forward f-k transform (NlogN) and an inverse k->x transform (.5NlogN) and the phase shifting operation. The latter requires N complex multiplications for each of $N^{1/2}$ extrapolation steps. Thus phase shift ends up with a leading term of $N^{3/2}$. The mixed-domain v(z) f-k has virtually the same cost estimate as phase-shift. The forward and inverse transforms costs are identical and the application of $m(\omega,\tau,k_x)$ is of order $N^{3/2}$. Finally, the full-Fourier method requires the same forward and inverse transforms as f-k migration and the application $M(\omega,\eta,k_x)$ is the same cost as the application of $m(\omega,\tau,k_x)$. The cost of computing $M(\omega,\eta,k_x)$ from $m(\omega,\tau,k_x)$ is $N^{1/2}$ FFT's of length $N^{1/2}$ but this must be done for each of .5 $N^{1/2}$ wavenumbers. The cost of building a table of exponential phase-shift operators, which is common to all but Stilt migration, has been neglected.

The comparison between f-k migration and its v(z) generalization (full-Fourier) is interesting. Both require a 2NlogN cost to go to and from the Fourier domain but the cost of the stilt mapping is of order N while its v(z) generalization is of order $N^{3/2}$logN. This difference is entirely due to the fact that $M(\omega,\eta,k_x)$ is known analytically in the constant velocity case and it is a simple Dirac delta function. Thus it follows that, if an analytic approximation for $M(\omega,\eta,k_x)$ can be found for general v(z), then the full-Fourier method could be made much more efficient.

Step	f-k	phase-shift	v(z) f-k mixed-domain	v(z) f-k full-Fourier
(x,t)->(f-k)	NlogN	NlogN	NlogN	NlogN
Stolt map	8N	-	-	-
build M	-	-	-	.25$N^{3/2}$logN
apply M	-	-	-	$N^{3/2}$
apply m	-	-	$N^{3/2}$	-
phase shift	-	$N^{3/2}$	-	-
(k,τ)->(x,τ)	-	.5NlogN	.5NlogN	-
(f-k)->(x,τ)	NlogN	-	-	NlogN
Total	2N(logN+4)	$N^{3/2}$+3/2NlogN	$N^{3/2}$+3/2NlogN	$N^{3/2}$(logN/4+1)+ 2NlogN

Contents

Table 2. Operations count estimates for the four Fourier algorithms discussed in the text. A 2-D dataset of dimension $N^{1/2}$ by $N^{1/2}$ is assumed.

Despite its cost, the full-Fourier v(z) f-k method is of interest for several reasons. First, from a theoretical perspective, it is the direct generalization of Stolt's algorithm to v(z). The mixed-domain method, while mathematically equivalent, does not make this connection obvious. Second, as mentioned above, the full-Fourier domain method could actually become the faster method if a reasonable analytic approximation for $M(\omega,\eta,k_x)$ could be found. Even without that, considerable speedup would occur if the application of $M(\omega,\eta,k_x)$ were done without a full matrix-vector multiply. That is, could the large, low-amplitude areas of $M(\omega,\eta,k_x)$ seen in Figures 8 and 10 be neglected in the filter application? Even better, could a fast numerical evaluation of $M(\omega,\eta,k_x)$ avoid calculating them in the first place? These questions pose interesting research problems.

DISCUSSION AND CONCLUSIONS

The theory of f-k migration can be extended to vertical velocity variations as a nonstationary filter. The result is a method that constructs the migrated Fourier spectrum directly from the unmigrated spectrum for arbitrary v(z). Like f-k migration, the v(z) f-k algorithm begins with a forward Fourier (f-k) transform, then applies *the migration filter* to the resulting spectrum, and finishes with an inverse Fourier (f-k) transform. The migration filter application, which generalizes Stolt's f-k spectral mapping, is done as a matrix-vector multiplication for constant horizontal wavenumber. In the constant velocity case, each row of the migration filter is an optimum, complex-valued, interpolation function that eliminates artifacts associated with spectral interpolation.

The migration filter is a nonstationary filter whose form is given analytically in the mixed domain of input frequency and output time. Three different forms were presented for it corresponding to constant velocity, an rms approximation for vertical velocity, and a first order WKBJ approximation. In the full-Fourier domain (input frequency and output frequency) the migration filter is a Dirac delta function for constant velocity; while, for variable velocity, it must be computed from the mixed-domain form with a numerical FFT. Higher order WKBJ forms are possible through modifications to the migration filter.

Variable velocity migration filters have their significant energy restricted to a fairway bounded by hyperbolae determined by the minimum and maximum instantaneous velocities.

Numerical tests show the v(z) f-k method performs as well as recursive phase shift for v(z) and produces a very clean output. Runtimes, for the full-Fourier domain method are comparable to phase-shift. Currently, the mixed-domain filter is faster than the full-Fourier domain though there are possibilities to speed-up the full-Fourier method.

In Li and Margrave (1998, elsewhere in this research report) we present the v(z) f-k method implemented for 2-D prestack geometries. This method handles P-P and P-S

Contents

reflections and has produced very high-resolution images of the Blackfoot target channels. This code has been implemented in Promax and is part of the CREWES 1998 software release.

This new v(z) f-k theory encompasses both f-k migration and phase-shift migration and suggests how more accurate theories might be constructed. There are a number of interesting directions in which to move this theory. Some are listed below:

- Higher order WKBJ corrections can easily be encompassed within the framework presented here. This might lead to more accurate amplitude estimates.

- This method can also be run in “reverse” to create a fast, acoustic, WKBJ modeling program. This might be useful in a variety of settings and could possibly be extended to the elastic case.

- 3D post-stack migrations are no more difficult that 2D prestack. 3D prestack might also be done through the use of constant-offset sections. Etgen (1998) is pursuing this avenue though his algorithm differs from that given here.

- Anisotropy could be included. The TIV case should be quite feasible and, coupled with the P-S imaging of Li and Margrave (1998) could be very beneficial.

- Extensions to lateral velocity variations are possible through a number of mechanisms. The most simple, appropriate for weak gradients, would involve a pre-migration stretch with v(z)/v(x,z) and then an imaging pass with v(z).

- The development of a v(z) extrapolator that responds to local gradients seems also possible. This might actually produce a better extension to lateral velocity variations as it would have a finite aperture unlike the constant velocity phase-shift extrapolator. Such an extrapolator could also be formulated to image a range of depths simultaneously.

ACKNOWLEDGMENTS

I wish to thank the sponsors of the CREWES project for their support of this research.

REFERENCES

Aki, K., and Richards, P.G., 1980, Quantitative Seismology, Volume 1: W.H. Freeman and Co.

Berkhout, A.J., 1985, Seismic Migration – Imaging of acoustic energy by wave field extrapolation: Elsevier, ISBN 0-444-42547-0.

Claerbout, 1985, Imaging the Earth's Interior: Blackwell Scientific Publications.

Etgen, J.T., 1998, V(z) f-k prestack migration of common-offset common-azimuth data volumes, part I: theory: Expanded abstracts 68th Annual SEG meeting, New Orleans, La., 1835-1838.

Gazdag, J., 1978, Wave-equation migration by phase shift: Geophysics, **43**, 1342-1351.

Gazdag, J., and Sguazzero, P., 1984, Migration of Seismic Data, Proceedings of the IEEE, **72**, 1302-1315.

Hale, D., and Artley, C., 1993, Squeezing dip moveout for depth-variable velocity: Geophysics, **58**, 257-264.

Margrave, G.F., 1998, Theory of nonstationary filtering in the Fourier domain with application to time-variant filtering: Geophysics, **63**, 244-259.

Contents

Ogden, R.T., 1997, Essential wavelets for statistical applications and data analysis: Birkhauser, ISBN 0-8176-3864-4.

Riley, K.F., Hobson, M.P., Bence, S.J., 1997, Mathematical Methods for Physics and Engineering: Cambridge University Press, ISBN 0521 55529 9.

Stolt, R.H., 1978, Migration by Fourier Transform, Geophysics: **43**, 23-48.

Schneider, W., 1978, Integral formulation for migration in two and three dimensions: Geophysics, **43**, 49-76.

Yilmaz, O, 1987, Seismic Data Processing: Society of Exploration Geophysicists.

Contents

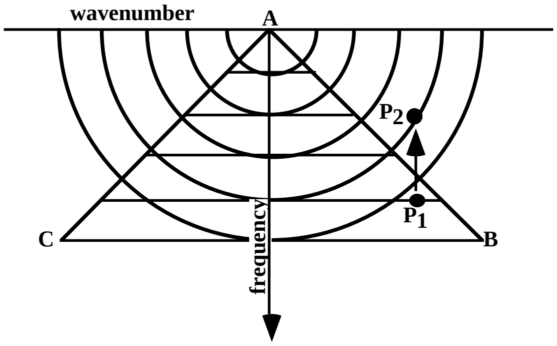

Figure 1. The f-k migration mapping developed by Stolt (1978) and described by equation 12. The f-k spectrum of the unmigrated zero-offset data is contained within the triangle ABC. The edges AC and AB of the triangle are evanescent boundaries where the migrated frequency η transitions from real to complex. The f-k migration mapping constructs the circular region from the triangle. A given point P2 is "mapped" at constant wavenumber from a point P1 whose frequency ω is related to η by equation 12. Since P1 generally will not fall on the input grid, a complex-valued interpolation must be done.

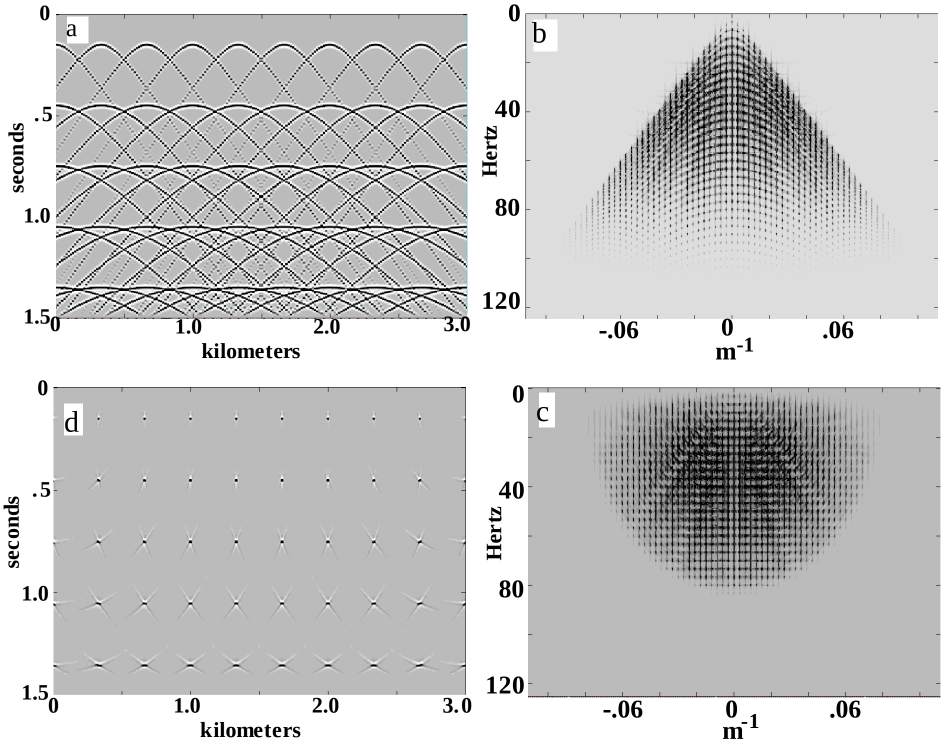

Figure 2. A numerical example of f-k migration by the Stolt algorithm of Figure 1. a) The unmigrated zero-offset section consisting of many point scatterers arrayed on a regular grid. b) The f-k spectrum of a). Note the restriction of energy to a triangular region. c) The f-k spectrum of the migrated data. It is constructed from b) by the Stolt mapping. d) The final migrated time section computed by an inverse f-k transform of c).

Contents

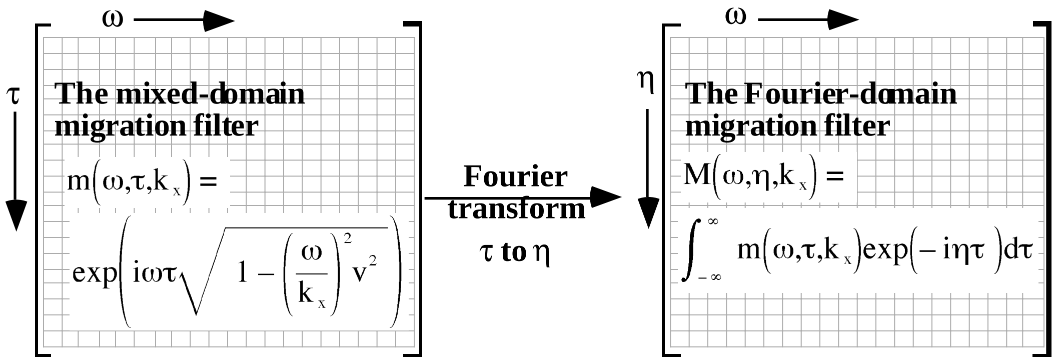
ω
τ
The mixed-domain migration filter
$m(\omega,\tau,k_x) = \exp\left(i\omega\tau\sqrt{1-\left(\frac{\omega}{k_x}\right)^2 v^2}\right)$
Fourier transform
τ to η
ω
η
The Fourier-domain migration filter
$M(\omega,\eta,k_x) = \int_{-\infty}^{\infty} m(\omega,\tau,k_x)\exp(-i\eta\tau)d\tau$

Figure 3. The migration filter, $m(\omega,\tau,k_x)$, in the mixed domain is depicted at left. In matrix form for constant k_x, $m(\omega,\tau,k_x)$ has ω increasing along a row and τ increasing along a column. Three simple analytic forms are discussed form this filter for constant velocity (equation 31), rms approximation (equation 33), and first order WKBJ theory (equation 34). The migration filter can either be applied directly in the mixed domain for moved entirely into the Fourier domain (right) by a Fourier transform along each column from τ to η.

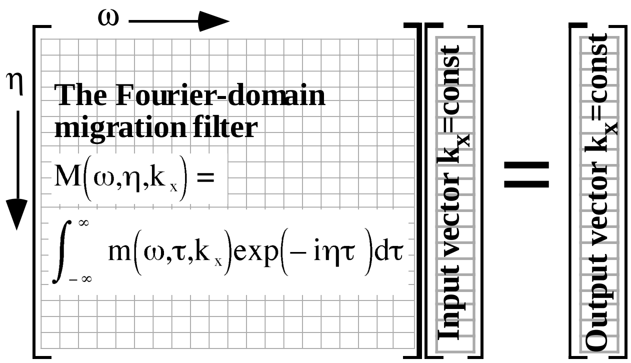
ω
η
The Fourier-domain migration filter
$M(\omega,\eta,k_x) = \int_{-\infty}^{\infty} m(\omega,\tau,k_x)\exp(-i\eta\tau)d\tau$
Input vector k_x=const
=
Output vector k_x=const

Figure 4. The application of the migration filter in the full Fourier domain is shown. The input data is a column vector of ω's at constant k_x and represents a single column of the f-k spectrum in Figure 2b. The vector of migrated frequencies, η, for the same k_x is computed by multiplying the Fourier domain migration filter from the left into the input vector. When velocity is constant, this process accomplishes the Stolt mapping. When velocity varies with v(z), the f-k migration process is generalized.

Contents

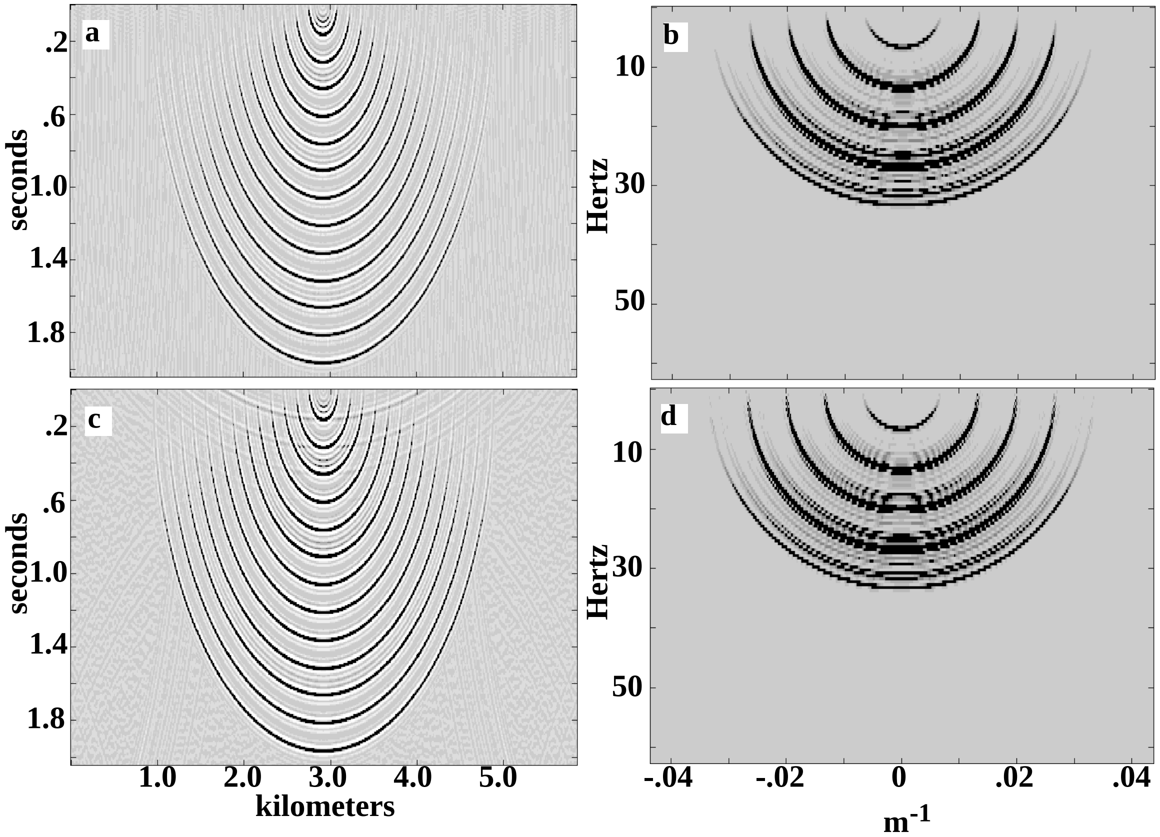

Figure 5. The impulse responses of the v(z) f-k algorithm (a and b) and the Stolt algorithm(c and d) are compared for a constant velocity of 2000 m/s. In the time domain (a and c) the algorithms produce wavefronts with identical trajectories but the Stolt algorithm produces extra artifacts on shallow impulses which are related to the complex-valued Fourier domain interpolation. There is also a apparent difference in amplitude on the steeper dips. This is currently unexplained. In the Fourier domain, the responses are very similar but, in the central wavenumbers, subtle differences appear that are again related to complex interpolation.

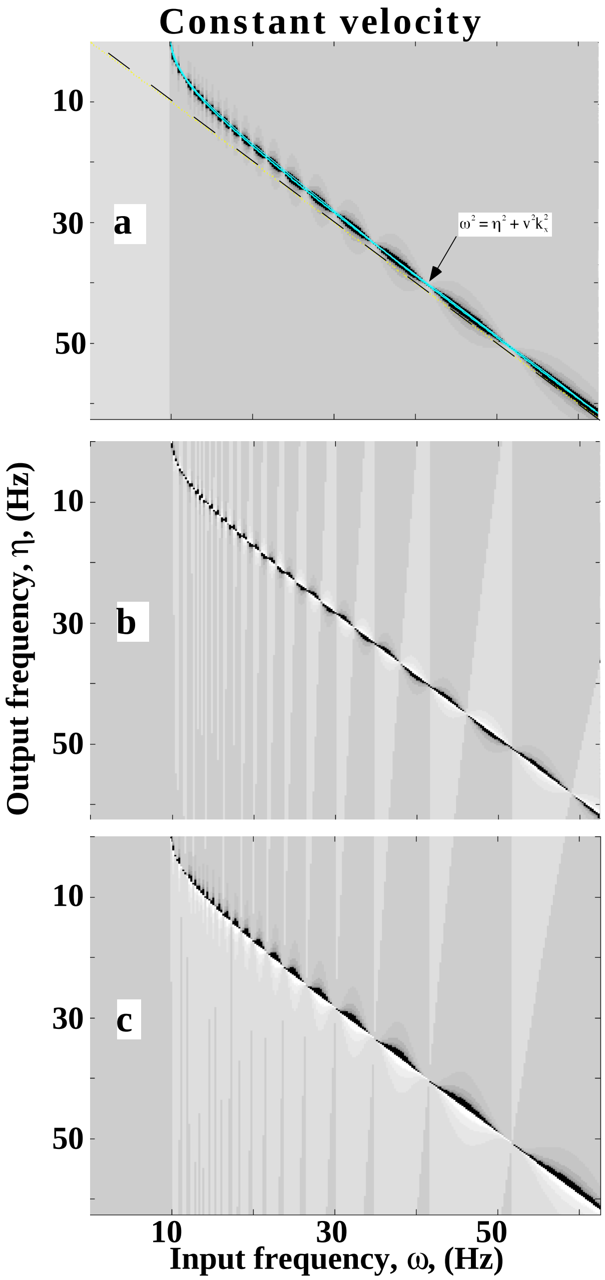

Figure 6. A numerical example of the Fourier domain migration filter, $M(\omega,\eta,k_x)$ is shown for a constant velocity of 2000 m/s and a wavenumber of .01 m^{-1}. a) shows the absolute value of the complex-valued filter while b) shows its real part and c) its imaginary part. The filter contains significant amplitudes only near a hyperbolic trajectory determined by $\omega^2=\eta^2+v^2k_x^2$ which is indicated by a solid line in a). In the limit of continuous sampling and infinite aperture, the constant velocity $M(\omega,\eta,k_x)$ becomes a Dirac delta function representing an infinitely high spike at all points on the hyperbolic trajectory. In the discrete simulation shown here, each row of $M(\omega,\eta,k_x)$ contains an optimal complex-valued interpolation function (roughly a sinc interpolator) which accomplishes the spectral interpolation required by the Stolt mapping (Figure 1) without any artifacts. The application of $M(\omega,\eta,k_x)$ to migrate a vector of wavenumbers is shown in Figure 4. The calculation of $M(\omega,\eta,k_x)$ from $m(\omega,\tau,k_x)$, whose form is known analytically, is shown in Figure 3.

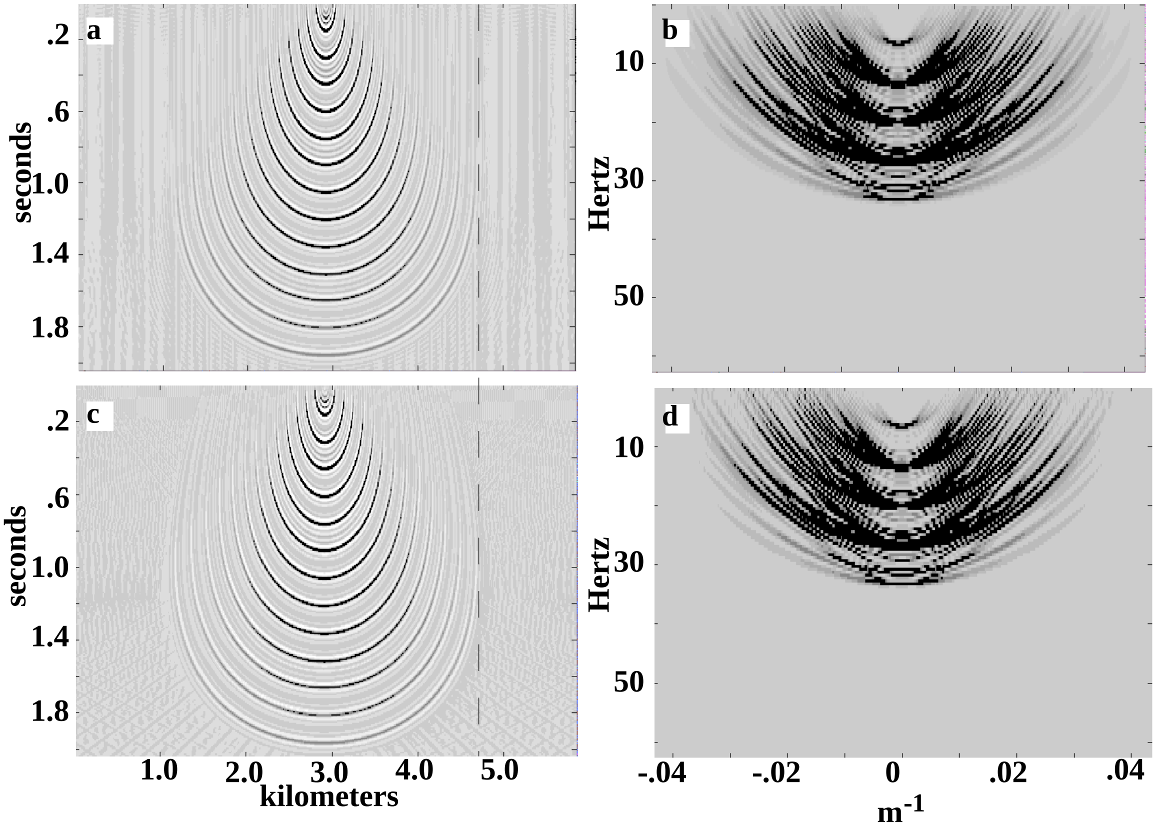

Figure 7. The impulse responses of the v(z) f-k algorithm with the WKBJ filter (a and b) and Gazdag's phase-shift algorithm (c and d) are compared for v(z)= 1430+.6z. The algorithms produce very similar results in both time and Fourier domains. The vertical dashed line marks the farthest lateral extent of the deepest wavefront for comparison with Figure 9.

Contents

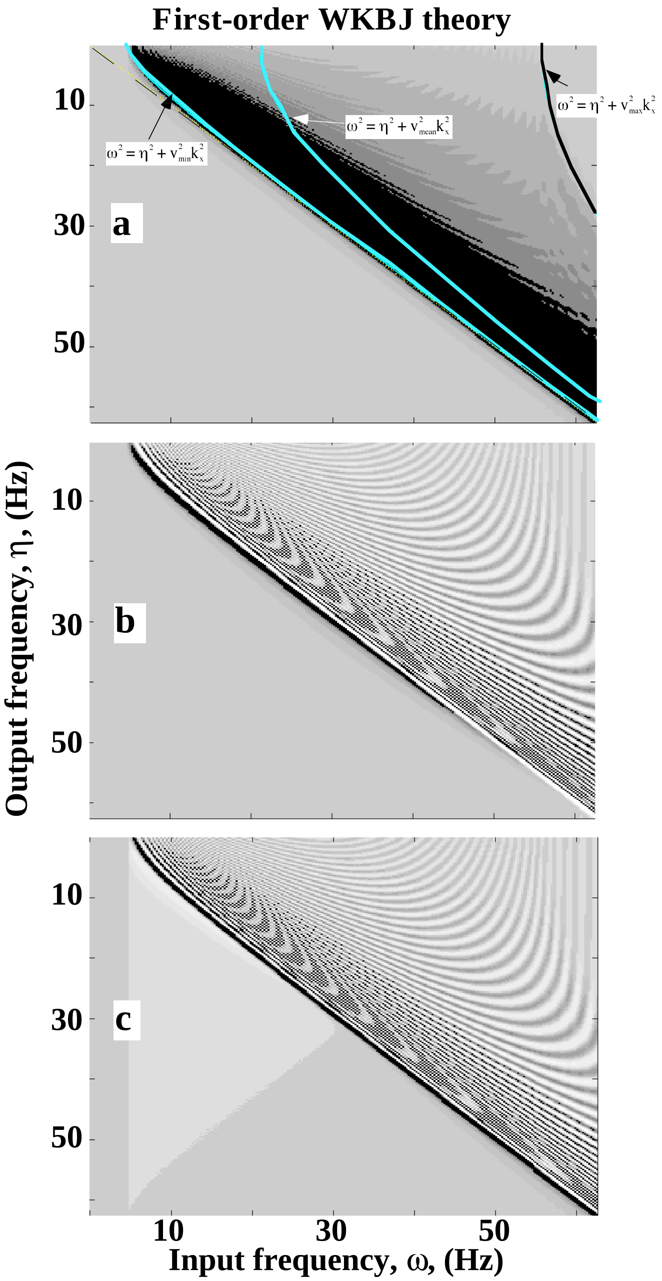

Figure 8. A numerical example of the Fourier domain migration filter, $M(\omega,\eta,k_x)$ is shown for a WKBJ approximation (equation 34) and a wavenumber of .01 m^{-1}. As in Figure 6, a) shows the absolute value of the complex-valued filter while b) shows its real part and c) its imaginary part. The WKBJ approximation is for a linear increase of instantaneous velocity with depth described by v(z)=1430 + .6z m/s. This is a superior way to accommodate vertical velocity variations and can be shown to properly account for all ray bending. Unlike the constant velocity case (Figure 6), $M(\omega,\eta,k_x)$ now has significant energy over much of the upper half of the matrix. The curves drawn on a) are hyperbola determined by $\omega^2=\eta^2+v^2k_x^2$ using the minimum , mean, and maximum instantaneous velocities. The minimum and maximum velocity curves form precise bounds for the numerically significant part of $M(\omega,\eta,k_x)$.

Contents

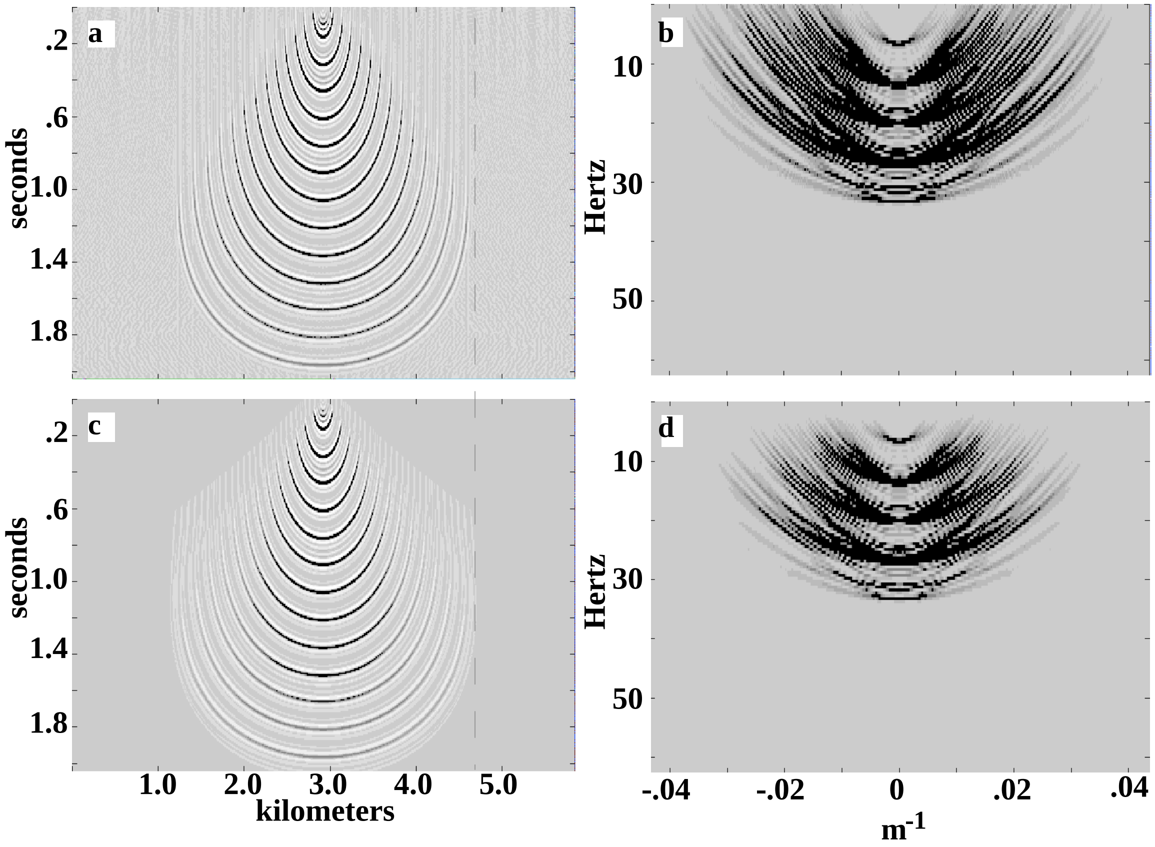

Figure 9. The impulse responses of the v(z) f-k algorithm with the rms filter (a and b) and an rms Kirchhoff algorithm (c and d) are compared for v(z)= 1430+.6z. The methods produce similar wavefront trajectories though the Kirchhoff code is not fully correct for the amplitudes of steeper dips. The vertical dashed line marks the farthest extent of the deepest wavefront of Figure 7. This makes it obvious that the steepest dips are not properly handled by the rms assumption.

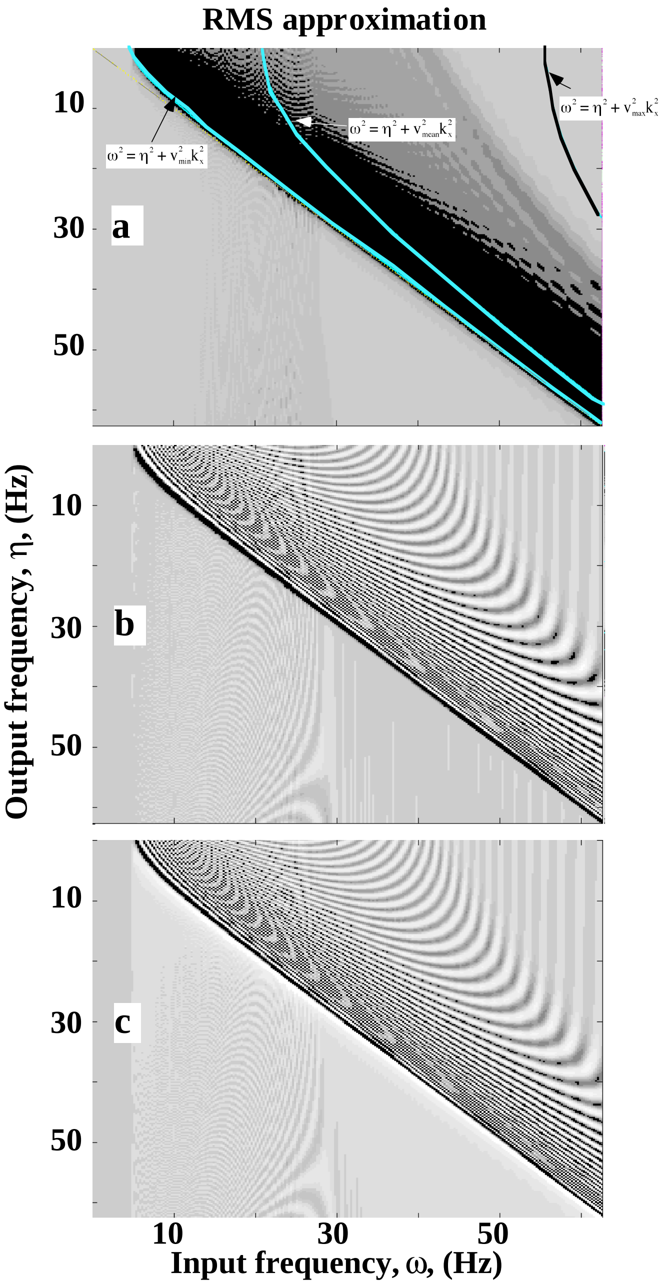

Figure 10. A numerical example of the Fourier domain migration filter, $M(\omega,\eta,k_x)$ is shown for an rms approximation (equation 33) and a wavenumber of .01 m^{-1} . As in Figures 6 and 8, a) shows the absolute value of the complex-valued filter while b) shows its real part and c) its imaginary part. The rms approximation is for a linear increase of instantaneous velocity with depth described by v(z)=1430 + .6z m/s. This is a simple way to accommodate vertical velocity variations by invoking the Dix equation to justify using straight rays at the rms velocity. Like the WKBJ case (Figure 8), $M(\omega,\eta,k_x)$ has significant energy over much of the upper half of the matrix. The curves drawn on a) are hyperbola determined by $\omega^2=\eta^2+v^2k_x^2$ using the minimum , mean, and maximum instantaneous velocities. The minimum and maximum velocity curves form approximate bounds for the numerically significant part of $M(\omega,\eta,k_x)$. The rms filter differs most from the WKBJ filter near the maximum velocity curve (see Figure 11).

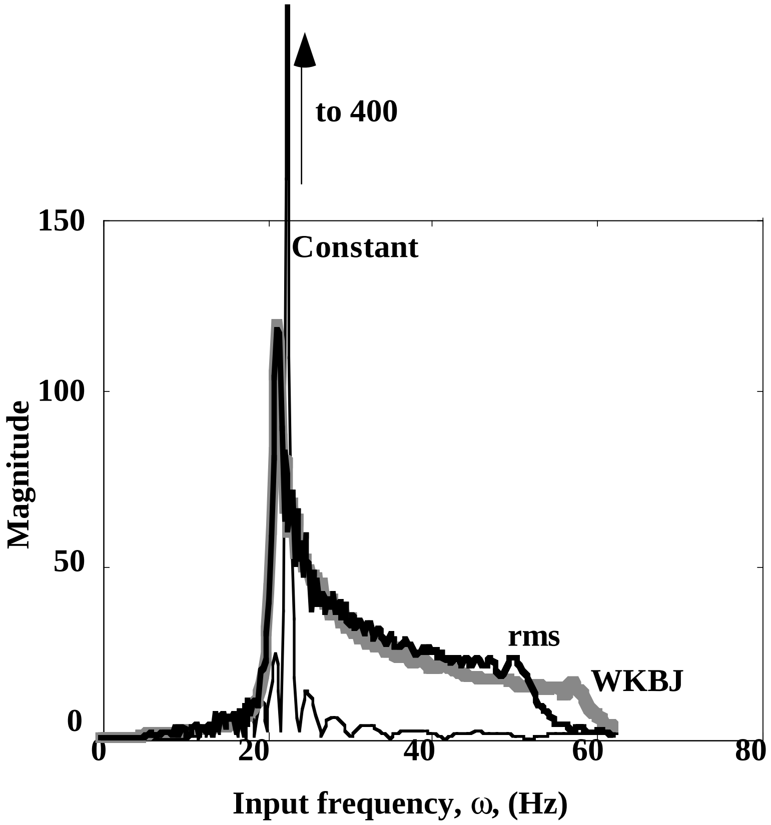

Figure 11. Profiles across Figures 6a, 8a, and 10a are shown at η=20 Hz. The constant velocity response approximates a band-limited Dirac delta function and has been clipped. The WKBJ response is the most accurate for variable velocity and is well approximated by the rms filter for low frequencies. Above 30 Hz, there are significant and systematic differences between the two. The precise form of the WKBJ and RMS filters is determined by the velocity function. For the nonevanescent frequencies, each ω can be related uniquely to a velocity through $\omega^2=\eta^2+v^2k_x^2$. The variable velocity filters show power at a particular ω in direct proportion to how much of the total τ (or z) range is covered by that particular velocity. Here velocity linear with depth was assumed and that can be shown to correspond to a velocity with is exponential with τ. This leads to the sharp peak at the front of the WKBJ response and the decaying tail. Other velocity functions will show different behavior.