

Fast Phase-shift time-stepping for RTM

Ben D. Wards, Gary F. Margrave, and Michael P. Lamoureux*

ABSTRACT

Reverse time migration (RTM) is a depth migration algorithm that can image overturned reflectors. Unfortunately, due to the numerical performance of finite difference operators, oversampling of the timestep and grid spacing make the algorithm slow relative to other migration algorithms. Pseudospectral methods calculate the spatial derivatives in the wavenumber domain but use a finite-difference approximation in the time domain and thus also suffer from numerical dispersion. The phase-shift time stepping (PSTS) equation propagates a solution to the two-way variable-velocity acoustic wave equation by calculating the spatial derivatives of the wavefield in the Fourier domain but does not use a finite-difference approximation for the time derivative. Instead, the PSTS equation adapts the exact solution for constant velocity medium to variable velocity medium by a locally homogeneous approximation. While usually faster than finite differencing, PSTS still imposes a considerable computational burden. A number of numerical approximations of the PSTS equations are derived. Firstly, we propose a method of timestepping in a linear velocity gradient. Secondly, a method that takes a number of time steps in the Fourier domain before rewindowing in the space domain. Thirdly, a method that propagates the derivative of the wavefield and the wavefield forward in time which has no limitation on the size of the timestep because of stability or aliasing.

INTRODUCTION

Reverse-time migration (Baysal et al., 1983; McMechan, 1983) is a depth migration algorithm. It can image reflectors using overturned waves and multiples. However, as a result of the sampling requirements, processing seismic surveys will either require harsh filtering to remove higher frequency data, thus increasing the timestep and grid spacing, or they will require long run times even with a cluster of computers. The fine sampling requirements occur because finite-difference operators propagate high frequencies with an incorrect dispersion relation. An example of the impressive performance, yet low frequency response, is the method of Jones et al. (2007).

Wards et al. (2007), and Wards et al. (2008) used the phase-shift time-stepping equation for reverse time migration. They adapted it to variable velocity by windowing the wavefield with a Gabor window and propagating the resulting wavefield with a constant velocity. This is a special case of an ϵ -separation, to be defined in the next section, of the kernel of the PSTS equation. The numerical complexity is linearly proportional the rank (number of velocity-window pairs) of the ϵ -separation. We consider more general ϵ -separations than piecewise constant. Additionally, we propose an alternative to the PSTS equation that propagates the wavefield and its derivative forward in time and in so doing does not suffer from an aliasing condition that limits the size of the timestep for the PSTS equation.

*POTSI, Dept. of Mathematics and Statistics, University of Calgary

AN ϵ -SEPARATION OF THE KERNEL OF THE PSTS EQUATION

To solve the variable velocity acoustic wave equation, the exact solution for a constant velocity acoustic wave equation can be adapted by the locally homogeneous approximation (Wards et al., 2008). This means that a solution is locally propagated with a constant velocity. The resulting equation, called the PSTS equation, is

$$U(\Delta t, \vec{x}) = -U(-\Delta t, \vec{x}) + 2\mathcal{F}^{-1} \left[\cos \left(2\pi v(\vec{x}) |\vec{k}| \Delta t \right) \mathcal{F} [U(0, \vec{x})] \right], \quad (1)$$

where $U(t, x, z)$ is the amplitude of the wave at the point (t, x, z) , x is the lateral coordinate, z is the depth coordinate, Δt is the size of the timestep taken in the t coordinate, $v(x, z)$ is the speed of propagation of the wave, $\vec{k} = (k_x, k_z)$ are the wavenumbers which are conjugate to the coordinates $\vec{x} = (x, z)$, and \mathcal{F} , and \mathcal{F}^{-1} are the forward, and inverse Fourier transform over the coordinates \vec{x} , and \vec{k} , respectively. The time $t = 0$ in equation (1) can be taken to be arbitrary so that equation (1) can be used recursively to timestep.

Equation (1) is too numerically complex to be used directly for wavefield propagation as it is necessary to evaluate a 2-dimensional Fourier integral at each point (x, z) in the output domain. In order for an efficient numerical implementation of the PSTS equation, it is advantageous to derive an ϵ -separation of the cosine kernel, $\cos(2\pi v(\vec{x}) |\vec{k}| \Delta t)$. This is to take advantage of the numerical speed of the FFT. We wish to determine functions $a_n(\vec{k})$ and $b_n(\vec{x})$, $n = 1, \dots, N$, so that

$$\int_{\mathbb{R}^2 \times \mathbb{R}^2} \left| \cos(2\pi v(\vec{x}) |\vec{k}|) - \sum_{n=1}^N a_n(\vec{k}) b_n(\vec{x}) \right| dx dz dk_x dk_z < \epsilon. \quad (2)$$

In particular, it is necessary that, N , the rank of the approximation be as small as possible, in order to minimize computational effort. In this case, the ϵ -separation is said to be optimal. Separating the \vec{k} and \vec{x} variables allows the use of the FFT for the numerical evaluation of equation (1),

$$\begin{aligned} U(\Delta t, \vec{x}) &= -U(-\Delta t, \vec{x}) + \sum_{n=1}^N \mathcal{F}^{-1} \left[a_n(\vec{k}) b_n(\vec{x}) \mathcal{F} [U(0, \vec{x})] \right] \\ &= -U(-\Delta t, \vec{x}) + \sum_{n=1}^N b_n(\vec{x}) \mathcal{F}^{-1} \left[a_n(\vec{k}) \mathcal{F} [U(0, \vec{x})] \right]. \end{aligned} \quad (3)$$

In Wards et al. (2008) equation (1) is numerically evaluated by propagating windowed copies of the wavefield with a constant velocity. When the velocity field is approximated by

$$\int_{\mathbb{R}^2 \times \mathbb{R}^2} \left| \sum_{n=1}^N \Omega_n(x, z) v_n - v(x, z) \right| dx dz dk_x dk_z < \delta, \quad (4)$$

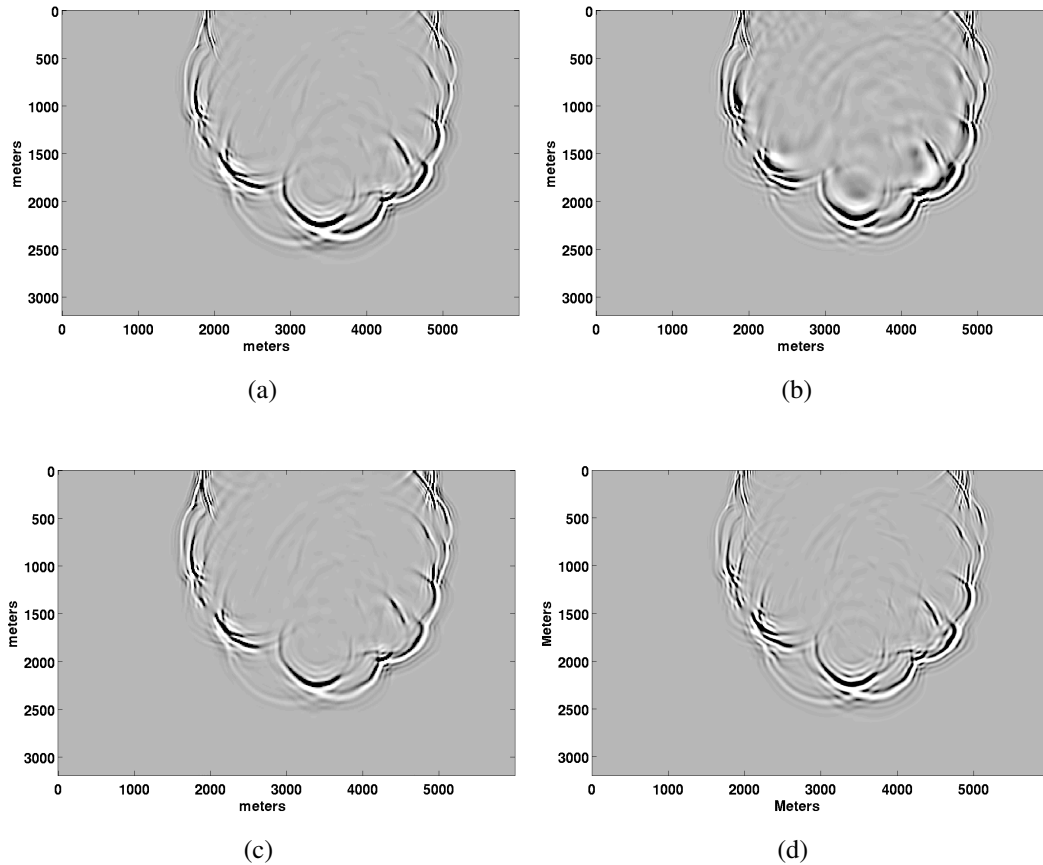


FIG. 1. Snapshot of the forward modelled shot 120 of the Marmousi dataset for various values of p . The grid spacing is $\Delta x = 12.5$ and the size of the time step is $\Delta t = 1.5ms$ and Gaussian with a halfwidth of $3.5\Delta x$ is convolved with the windowing functions to suppress reverberations. (a) Windowing after propagation ($p = 0$). (b) Windowing equally before and after propagation ($p = 0.5$). (c) Windowing before propagation ($p = 1$). (d) Using finite difference with $\Delta x = 5m$ and $\Delta t = 0.3ms$.

the resulting time stepping equation for a particular mean error δ is

$$U(\Delta t, \vec{x}) = -U(-\Delta t, x, z) + \sum_{n=1}^N 2\mathcal{F}^{-1}[\cos(2\pi\omega_n\Delta t)\mathcal{F}[\Omega_n(x, z)U(0, x, z)]], \quad (5)$$

where

$$\omega_n(k_x, k_z) = v_n\sqrt{k_x^2 + k_z^2}, \quad (6)$$

$\Omega_n(x, z)$ forms a partition of unity with

$$\sum_{n=1}^N \Omega_n(x, z) = 1, \quad 0 \leq \Omega_n(x, z) \leq 1, \quad (7)$$

and v_n is the velocity used for propagation in the n th window. By the Cauchy-Schwarz-Bunyakovsky theorem, equation (4) implies the existence of an ϵ -separation like that of equation (2).

In equation (3) it is advantageous for the functions $b_n(\vec{x})$ to be placed before the first Fourier transform. This can be done whenever $b_n(\vec{x})$ is a simple windowing function. In this case, the timestepper is

$$U(\Delta t, \vec{x}) = -U(-\Delta t, \vec{x}) + \sum_{n=1}^N 2\mathcal{F}^{-1} \left[a_n(\vec{k})\mathcal{F} [b_n(\vec{x})U(0, \vec{x})] \right]. \quad (8)$$

An issue of interest is whether to window before and/or after propagation. Therefore, an alternative approximation to equation (1), when $g_n = (\Omega_n)^p$, $\gamma_n = (\Omega_n)^{p-1}$, is

$$U(\Delta t, \vec{x}) = -U(-\Delta t, x, z) + \sum_{n=1}^N \gamma_n \mathcal{F}^{-1} \cos(2\pi\omega_n\Delta t)\mathcal{F}[g_n U(0, x, z)]. \quad (9)$$

Figure 1 compares different values of p to a finite difference solution. All 4 images are very similar. The case $p = 1/2$ takes approximately twice as long to compute as the cases $p = 0, 1$.

A Taylor series can be used to approximate the cosine operator in the variable \vec{x} about the point \vec{x}_0 . This is done to construct more general ϵ -separations than using constant velocity windows. For some velocity functions this neighborhood can be extended to the entire domain of computation. The power series expansion about the point $\vec{x} = \vec{x}_0$ for the function $\cos(2\pi v(\vec{x})|\vec{k}|\Delta t)$ is

$$\begin{aligned} \cos(2\pi v(\vec{x})|\vec{k}|\Delta t) &= \cos(2\pi v(\vec{x}_0)|\vec{k}|\Delta t) \\ &- \sin(2\pi v(\vec{x}_0)|\vec{k}|\Delta t)(\vec{x} - \vec{x}_0) \cdot \nabla v(\vec{x}_0)2\pi|\vec{k}|\Delta t \\ &- \cos(2\pi v(\vec{x}_0)|\vec{k}|\Delta t)[(\vec{x} - \vec{x}_0) \cdot \nabla v(\vec{x}_0)2\pi|\vec{k}|\Delta t]^2 \\ &- \sin(2\pi v(\vec{x}_0)|\vec{k}|\Delta t) \sum_{l=1}^2 \sum_{k=1}^2 (\vec{x} - \vec{x}_0)_k (\vec{x} - \vec{x}_0)_l \frac{\partial^2 v(\vec{x}_0)}{\partial x_i \partial x_j} (2\pi|\vec{k}|\Delta t)^2 \\ &+ H.O.T. \end{aligned} \quad (10)$$

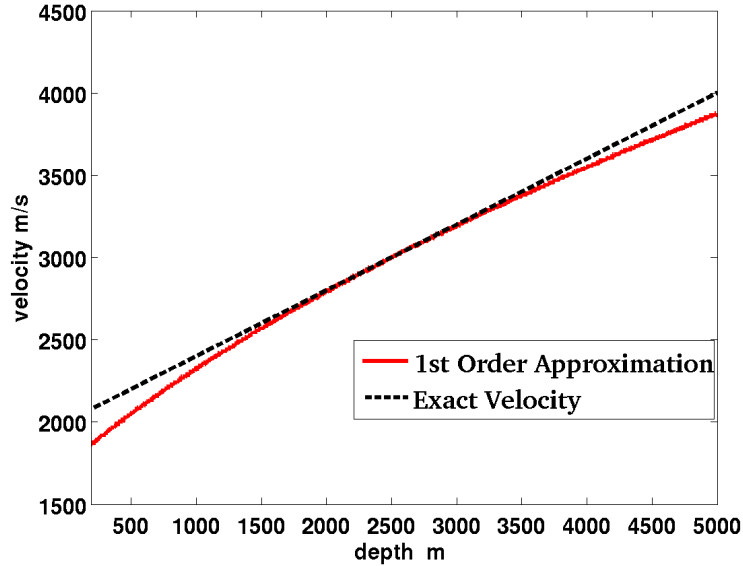


FIG. 2. The lower red line is the velocity of propagation using the 1st order approximation from equation (11). The higher dashed black line is the velocity to be approximated $v(\vec{x}) = v_0 + \vec{a} \cdot (\vec{x} - \vec{x}_0)$.

where H.O.T. denotes higher order terms.

When $v(\vec{x}) = v_0 + \vec{a} \cdot (\vec{x} - \vec{x}_0)$, equation (10) reduces to

$$\begin{aligned}
 \cos(2\pi v(\vec{x})|\vec{k}|\Delta t) &= \cos(2\pi v(\vec{x}_0)|\vec{k}|\Delta t) \\
 &- \sin(2\pi v(\vec{x}_0)|\vec{k}|\Delta t)(\vec{x} - \vec{x}_0) \cdot \vec{a} 2\pi|\vec{k}|\Delta t \\
 &- \cos(2\pi v(\vec{x}_0)|\vec{k}|\Delta t)[(\vec{x} - \vec{x}_0) \cdot \vec{a} 2\pi|\vec{k}|\Delta t]^2 \\
 &+ H.O.T.
 \end{aligned} \tag{11}$$

The 2nd order and 1st order Taylor series approximations, equation (11), are compared by observing the velocity at which they propagate a plane wave perpendicular to the velocity gradient. Figure 2 is the 1st order approximation and Figure 3 is the 2nd order approximation. Both provide acceptable solutions. The 2nd order solution required 10 percent more computation time. Given that both approximations systematically over estimate or under estimate the velocity, a polynomial expansion that is uniform about over an interval would provide a better approximation. Figures 2 and 3 suggest using a window of length 2000m.

SAMPLING ISSUES

Ideally it would be desired to timestep at the Nyquist sampling rate of the seismic data,

$$\Delta t_{nyq} = \frac{1}{2f_{max}} \tag{12}$$

where f_{max} is the maximum signal frequency. Reverse time migration methods however often require finer sampling. For the PSTS equation, the smallest wavelength must be sampled at least twice,

$$\Delta x < \frac{V_{min}}{2f_{max}}, \tag{13}$$

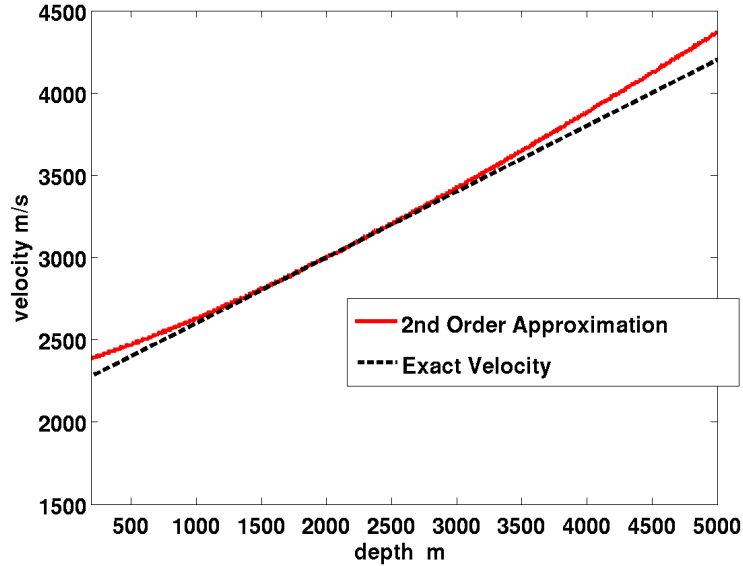


FIG. 3. The higher red line is the velocity of propagation using the 2nd order approximation from equation (11). The lower dashed black line is the velocity to approximated $v(\vec{x}) = v_0 + \vec{a} \cdot (\vec{x} - \vec{x}_0)$.

where V_{min} is the minimum velocity of the model and Δx is the gridspacing in the x and z directions. The timestep must satisfy (Wards et al., 2007)

$$\Delta t < \frac{\Delta x}{\sqrt{2}V_{max}}. \quad (14)$$

For processing the seismic survey with $[V_{min}, V_{max}] = [1500m/s, 5500m/s]$ and $f_{max} = 50Hz$, the sampling requirements are $\Delta x < 15m$, $\Delta t < 0.0015s$, and $t_{nyq} = 0.01s$.

To minimize computation time, a multi-radix FFT is used. The computation domain is padded to the next integer with a large number of prime factors. A pad is necessary to prevent wraparound of the FFT and to enforce a free surface boundary condition.

SALTDOME COMPARISONS

The linear velocity approximation in equation (11) is used to image the overhung flanks of a salt dome for a poststack RTM. The geometry of velocity model is simple $v(x) = 1500 + 0.8z$ and as a result the computational times are not always comparable to more realistic situations. Generally a more complicated velocity model will make the PSTS slower by requiring more Gabor windows.

Figure 4 (a) is the image of the salt dome when one window with a linear velocity is used. There are significant errors in the velocity model which corresponds to incorrect position of the salt dome. However the migration is extremely cheap. Figure 4 (c) is the image using $N = 4$ linear velocity windows. The velocity model is very well approximated. Figure 4 (e) is the image using 12 constant velocity Gabor Windows. There some migration artifacts due some backscattered between windows. Figure 4 (g) is the imaging using 2nd order time and 4th order space finite differencing. For the finite difference RTM the grid spacing is $dx = 5$, the timestep is $dt = 1.1ms$, and the maximum fre-

quency is $f_{max} = 50Hz$. While for all of the PSTS migrations $dx = 15$, $dt = 4ms$, and $f_{max} = 50Hz$. Figure 5 shows the computation times for the migrations in Figure 4.

PSTS WITH THE DERIVATIVE OF THE WAVEFIELD

The PSTS equation calculates the wavefield at $U(t + \Delta t, \vec{x})$ using the wavefields $U(t, \vec{x})$ and $U(t - \Delta t, \vec{x})$. It is also possible to use the derivative of the wavefield $\partial U(t, \vec{x})/\partial t$ instead of the past wavefield $U(t - \Delta t, \vec{x})$. This equation propagates the derivative of the wavefield and the wavefield forward in time. In doing so it does not suffer from a limitation on the timestep but requires twice as much computation and memory as the PSTS equation to compute one timestep. The timestep is limited by the need to window in the space domain which allows propagation with the local velocity and by the need to calculate the imaging condition at each spatial position of the waves. Additionally, prestack RTM requires frequent snapshots of the wavefield for crosscorrelation, and source and receiver fields need to be backpropagated and therefore the Nyquist frequency limits the timestep.

A solution to the constant velocity acoustic wave equation with initial conditions $U(0, x, z) = f_0(x, z)$ and $\partial U(0, x, z)/\partial t = f_1(x, z)$ is

$$U(t, \vec{x}) = \int_{\mathbb{R}^2} \cos(2\pi\omega(k_x, k_z)t) \hat{f}_0(k_x, k_z) \exp(2\pi i \vec{x} \cdot \vec{k}) dx dz + \int_{\mathbb{R}^2} \sin(2\pi\omega(k_x, k_z)t) \frac{\hat{f}_1(k_x, k_z)}{2\pi\omega(k_x, k_z)} \exp(2\pi i \vec{x} \cdot \vec{k}) dx dz, \quad (15)$$

where \hat{f}_0, \hat{f}_1 are the Fourier transform over the spatial coordinates (x, z) of f_0, f_1 , respectively. The time derivative of the wavefield can be calculated at any time, taking the derivative of both sides of equation 15,

$$\frac{\partial U}{\partial t}(t, x, z) = - \int_{\mathbb{R}^2} 2\omega(k_x, k_z) \sin(2\pi\omega(k_x, k_z)t) \hat{f}_0(k_x, k_z) \exp(2\pi i \vec{x} \cdot \vec{k}) dx dz + \int_{\mathbb{R}^2} \cos(2\pi\omega(\vec{k})t) \hat{f}_1(\vec{k}) \exp(2\pi i \vec{x} \cdot \vec{k}) dx dz. \quad (16)$$

Unlike equation (1) which is governed by the stability condition $\max(v(\vec{x}))\Delta x/\Delta t < 1/\sqrt{2}$, equations (15) and (16) are stable for any size time step. Thus it could offer greater computational efficiency.

Collecting the equations above, a time step is accomplished by

$$\left. \begin{aligned} U(\Delta t + t, \vec{x}) &= \mathcal{F}^{-1} \left[\cos(2\pi\omega(\vec{k})\Delta t) \hat{U}(t, \vec{k}) \right] \\ &+ \mathcal{F}^{-1} \left[\frac{\sin(2\pi\omega(\vec{k})\Delta t)}{2\pi\omega(\vec{k})} \frac{\partial \hat{U}}{\partial t}(t, \vec{k}) \right] \\ \frac{\partial U}{\partial t}(\Delta t + t, \vec{x}) &= -\mathcal{F}^{-1} \left[2\pi\omega(\vec{k}) \sin(2\pi\omega(\vec{k})\Delta t) \hat{U}(t, \vec{k}) \right] \\ &+ \mathcal{F}^{-1} \left[\cos(2\pi\omega(\vec{k})\Delta t) \frac{\partial \hat{U}}{\partial t}(t, \vec{k}) \right]. \end{aligned} \right\} \quad (17)$$

Let $V(t) = \frac{\partial U}{\partial t}(t)$ and define the operator

$$\mathcal{K} = \begin{bmatrix} \mathcal{F}^{-1} \cos(2\pi\omega(\vec{k})\Delta t) \mathcal{F} & \mathcal{F}^{-1} \frac{\sin(2\pi\omega(\vec{k})\Delta t)}{2\pi\omega(\vec{k})} \mathcal{F} \\ -\mathcal{F}^{-1} 2\pi\omega(\vec{k}) \sin(2\pi\omega(\vec{k})\Delta t) \mathcal{F} & \mathcal{F}^{-1} \cos(2\pi\omega(\vec{k})\Delta t) \mathcal{F} \end{bmatrix}. \quad (18)$$

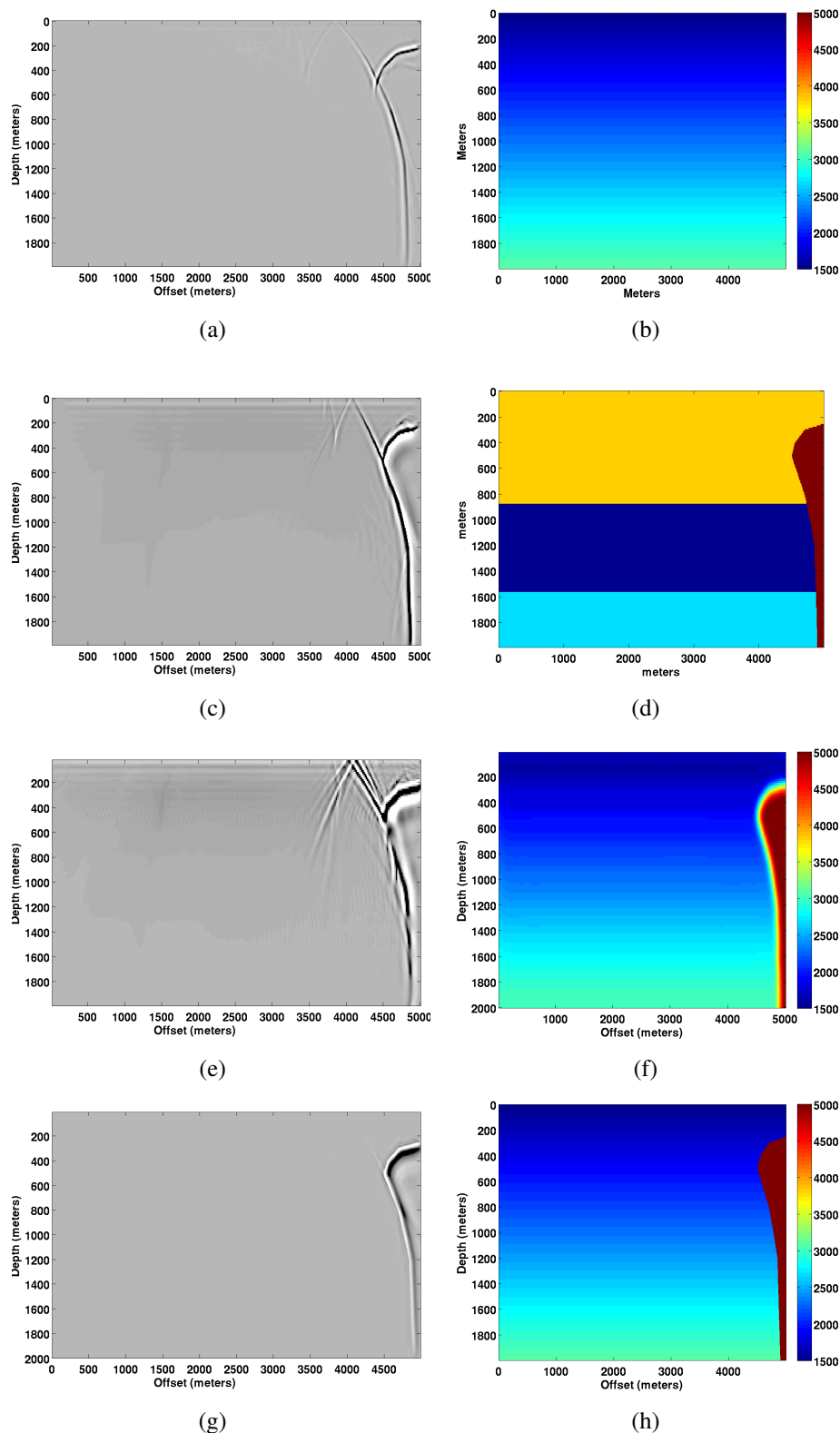


FIG. 4. (a) Poststack image of salt dome using one Gabor window with one linear velocity approximation. (b) The corresponding velocity used in (a). (c) Salt dome image using 4 Gabor windows 3 of which use the linear approximation. (d) Image showing the different regions in the Gabor Decomposition. Colours do not correspond to a velocity. (e) Image using a constant velocity 12 window PSTS approximation. (f) The effective velocity used in (e). (g) Finite-difference RTM. (h) Exact velocity model.

number of Gabor windows	Time (s)	method
1	20s	1 Linear velocity window
4	65s	3 Linear velocity windows
12	62s	12 constant velocity windows
N/A	250s	Finite difference

FIG. 5. Computation time for the images in Figure 4

Then equation (17) can be viewed more succinctly in matrix form as

$$\begin{bmatrix} U(t) \\ V(t) \end{bmatrix} = \mathcal{K} \begin{bmatrix} U(0, \vec{x}) \\ V(0, \vec{x}) \end{bmatrix}, \quad (19)$$

or completely in the the spatial wavenumber domain by letting the matrix

$$K(\Delta t, \vec{k}) = \begin{bmatrix} \cos(2\pi\omega(\vec{k})\Delta t) & \frac{\sin(2\pi\omega(\vec{k})\Delta t)}{2\pi\omega(\vec{k})} \\ -2\pi\omega(\vec{k}) \sin(2\pi\omega(\vec{k})\Delta t) & \cos(2\pi\omega(\vec{k})\Delta t) \end{bmatrix}. \quad (20)$$

Then

$$\begin{bmatrix} \hat{U}(t + \Delta t, \vec{k}) \\ \hat{V}(t + \Delta t, \vec{k}) \end{bmatrix} = K \begin{bmatrix} \hat{U}(0, \vec{k}) \\ \hat{V}(0, \vec{k}) \end{bmatrix}, \quad (21)$$

where the matrix K satisfies the functional relation $K(\Delta t + \Delta s) = K(\Delta t)K(\Delta s)$.

If the operator \mathcal{K}_n corresponds to \mathcal{K} in equation (18) with $\omega = v_n|\vec{k}|$. Then a timestepper for the wave equation is

$$\begin{bmatrix} U(t + \Delta t) \\ V(t + \Delta t) \end{bmatrix} = \sum_{n=1}^N \mathcal{K}_n \begin{bmatrix} \Omega_n(\vec{x})U(t, \vec{x}) \\ \Omega_n(\vec{x})V(t, \vec{x}) \end{bmatrix}. \quad (22)$$

To avoid storing the entire shotfield during RTM, 34 snapshots are stored which are used as checkpoints to recalculate the field at the missing times. By replacing Δt with $-\Delta t$ the operator $\mathcal{K}(-\Delta t)$ in equation (22) can propagate backwards.

Figure 6 are snap shots at 0.8s of shot 120 of the Marmousi data set (Versteeg, 1994). For RTM typically a smoothed velocity field or nonreflecting wave equation is used to eliminate internal reflections. We have chosen 2 different values of the smoother. Also, Figure 6 shows how the wavefields changes when different sizes of the timestep are taken. The computational complexity is inversely proportional to the length of the timestep.

AN EVOLUTIONARY SOLUTION TO THE ACOUSTIC WAVE EQUATION

Rather than solving a partial differential equation (PDE) directly by a dispersive pseudo-spectral method or finite difference method, Tal-Ezer (1986) suggest first rewriting the PDE as a first order system and then approximated the analytical solution. The variable velocity wave acoustic equation is

$$\frac{\partial^2 U}{\partial t^2} = v^2(x, z) \left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial z^2} \right), \quad (23)$$

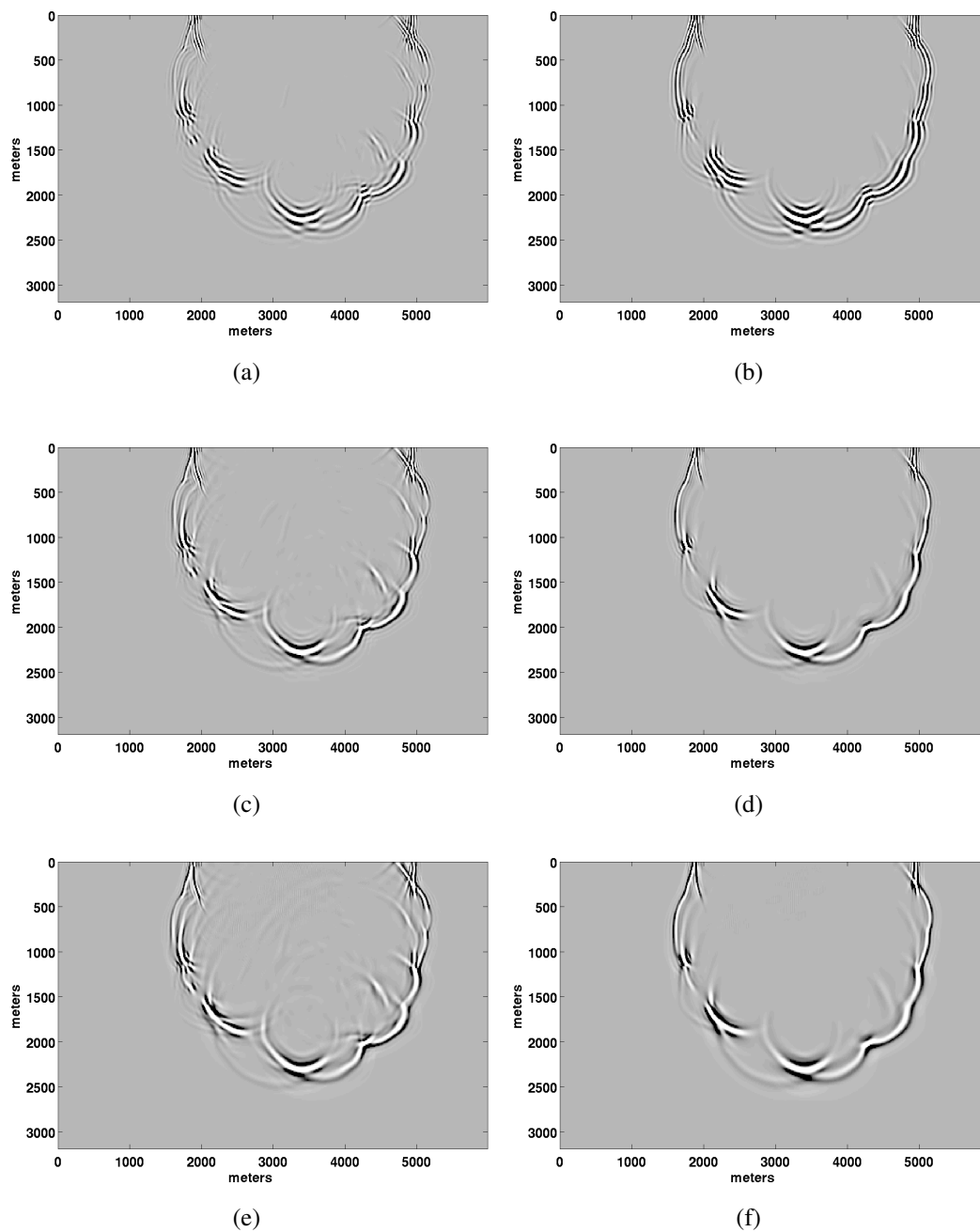


FIG. 6. Forward model shot 120 of Marmousi data set using equation (22). The grid spacing $\Delta x = 12.5$ and the maximum frequency $f_{max} = 50Hz$. The shots were computed for various values of the halfwidth of the smoother and time step Δt . (a) $\Delta t = 1.5ms$ and the halfwidth is $3.5dx$. (b) $\Delta t = 1.5ms$ and the halfwidth is $7dx$. (c) $\Delta t = 4ms$ and the halfwidth is $3.5dx$. (d) $\Delta t = 4ms$ and the halfwidth is $7dx$. (e) $\Delta t = 8ms$ and the halfwidth is $3.5dx$. (f) $\Delta t = 8ms$ and the halfwidth is $7dx$.

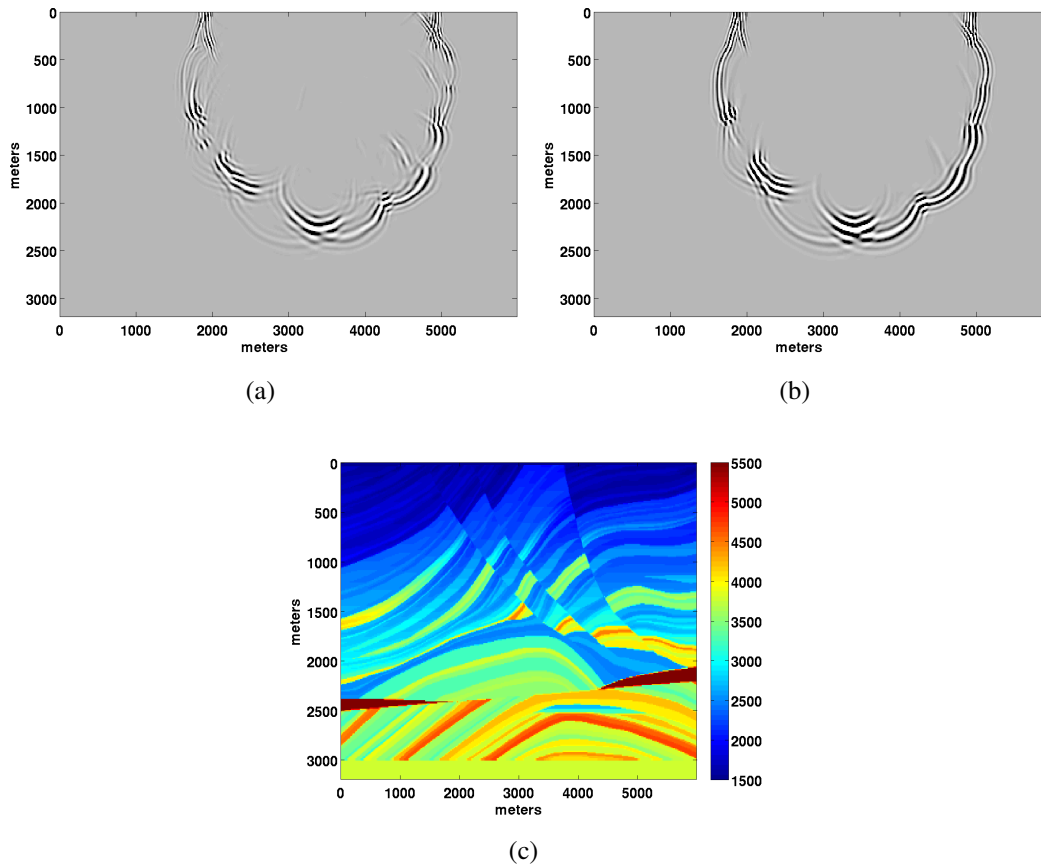


FIG. 7. Forward model shot 120 of Marmoussi data set using equation (5) for comparison in Figure 6. The grid spacing $\Delta x = 12.5$ and the maximum frequency $f_{max} = 50Hz$. The shots were computed for various values of the halfwidth of the smoother and time step Δt . (a) $\Delta t = 1.5ms$ and the halfwidth is $3.5dx$. (b) $\Delta t = 1.5ms$ and the halfwidth is $7dx$. (c) The Marmoussi velocity model. A 16 Gabor window partition is used to approximate the velocity field.

where $v(x, z)$ is the spatially dependent velocity, $U(t, x, z)$ is the wavefield amplitude at the point (t, x, z) , t is the time variable, z is the depth and x is the offset. Transforming equation (23) into a system of first order linear PDEs gives

$$\frac{\partial}{\partial t} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ v^2(x, z) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right) & 0 \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}, \quad (24)$$

subject to the initial condition

$$\begin{bmatrix} U(0, \vec{x}) \\ V(0, \vec{x}) \end{bmatrix} = \begin{bmatrix} f(\vec{x}) \\ g(\vec{x}) \end{bmatrix}. \quad (25)$$

The abstract solution can be written as the exponential of the linear operator

$$\mathcal{L} = \begin{bmatrix} 0 & 1 \\ v^2(x, z) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right) & 0 \end{bmatrix}, \quad (26)$$

so that

$$\begin{bmatrix} U(t, \vec{x}) \\ V(t, \vec{x}) \end{bmatrix} = \exp(\mathcal{L}t) \begin{bmatrix} U(0, \vec{x}) \\ V(0, \vec{x}) \end{bmatrix}, \quad (27)$$

where the exponential operator is

$$\exp(\mathcal{L}t) = \sum_{n=0}^{\infty} \frac{t^n \mathcal{L}^n}{n!}. \quad (28)$$

Then Tal-Ezer (1986) approximates the exponential operator by an expansion in terms of modified Chebychev polynomials and Bessel functions.

It is possible to reformulate (Etgen, 1989) equation (27) in terms of a cosine of the operator \mathcal{L} . Then

$$U(t + \Delta t) = -U(t - \Delta t) + \cos(\mathcal{L}\Delta t)U(t), \quad (29)$$

where

$$\cos(\mathcal{L}\Delta t) = \sum_{n=0}^{\infty} (-1)^n \frac{(\Delta t)^{2n} \mathcal{L}^{2n}}{(2n)!}. \quad (30)$$

It is also possible to approximate the cosine operator with a modified Chebychev polynomials and Bessel functions. For constant velocity equations (29) and (1) are the same.

MULTISTEPPING

For a constant velocity wavefield, it is possible to propagate a wavefield completely in the wavenumber domain without the need at intermediate stages to change back into the space domain with a Fourier transform. Whenever an intermediate snapshot is needed it is

possible to transform back into the space domain with a FFT. In this case equation (1) with $v(\vec{x})$ a constant is

$$\hat{U}(\Delta t, \vec{k}) = -\hat{U}(-\Delta t, \vec{k}) + \cos(2\pi v|\vec{k}|\Delta t)\hat{U}(0, \vec{k}). \quad (31)$$

To use equation (31) for a variable velocity medium each windowed wavefield is propagated a number of times in the wavenumber domain. To prevent aliasing and to insure the velocity field is accurately approximated the number of steps m is chosen to satisfy

$$mdt < \frac{1}{2f_{max}}, \quad (32)$$

where f_{max} is the maximum frequency in the seismic data. If

$$H_n = \begin{bmatrix} 0 & 1 \\ -1 & M_n \end{bmatrix} \quad (33)$$

where $M_n = \cos(2\pi\omega_n\Delta t)$. Then one multi-timestep is accomplished by

$$\begin{bmatrix} U(t + (n-1)\Delta t, \vec{x}) \\ U(t + n\Delta t, \vec{x}) \end{bmatrix} = \mathcal{F}^{-1} \sum_{n=1}^N H_n^m \begin{bmatrix} \mathcal{F}[\Omega_n(\vec{x})U(t, \vec{x})] \\ \mathcal{F}[\Omega_n(\vec{x})U(t - \Delta t, \vec{x})] \end{bmatrix}, \quad (34)$$

where the inverse Fourier transform is applied to each component of the vector that it operators on. This equation takes $2(N+1)$ 2-D FFTs to compute the wavefield at time $m\Delta t + t$ and $(m+1)\Delta t + t$ or if the wavefield at times $t, t + \Delta t, \dots, (m+1)\Delta t + t$ needs to be calculated then it takes $2N + m$ FFTs. For the Marmousi data set we used $m = 6$ for the number of times we propagate in the wavenumber domain and $N = 16$ for the number of windows used to construct the velocity partition. In contrast, one timestep using equation (5) takes $N + 1$ FFTs. Figure 8 shows that using $m = 6$ cuts the calculation time in half.

Alternatively, those Gabor windows for which either the receiver field is being injected into or the shot is being injected can be propagated by windowing and a Fourier transforming at each time step while the other windows are propagated with equation (34). Figure 9 is a poststack migration of a salt dome using equation (34). The image corresponding to the parameter $m = 20$ shows reverberations do to the miss cancellation of energy.

For prestack RTM it is necessary compute the wavefield at a fine spatial sampling to avoid aliasing during the imaging condition. If a fast coarse timestepper can be derived it would be possible recompute at a finer timestep using a fast dispersive finite difference propagator.

SOURCE MODELLING

Let $F(t, \vec{x})$ be the shotfield used to start propagation where the t, \vec{x} variables encode the source location and wavelet. The system wave equation with an inhomogeneous source function is,

$$\frac{\partial}{\partial t} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ v^2(x, z) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right) & 0 \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} + \begin{bmatrix} F(t, \vec{x}) \\ \frac{\partial F}{\partial t}(t, \vec{x}) \end{bmatrix}. \quad (35)$$

m	time (s) no intermediate calculations	accurate and stable
0	128	yes
4	112	yes
6	88	yes
12	72	yes
20	63	no

FIG. 8. Computation time for forward propagating a wavefield for various values of the speedup factor m using equation (34). The speed up factor $m = 0$ corresponds to using the equation (5).

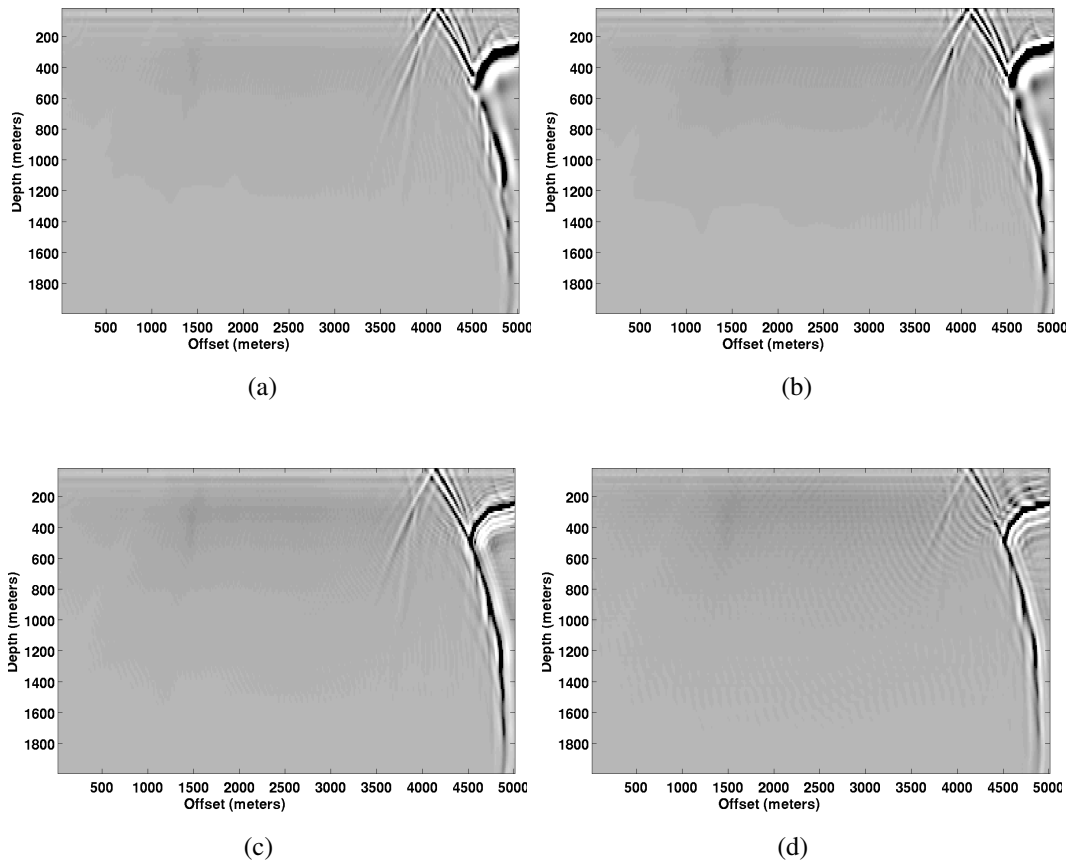


FIG. 9. Images of the salt dome model migrated using equation (34) by taking m equal to 4, 6, 12, and 20 timesteps in (a),(b) , (c), and (d), respectively.

The solution in terms of the exponential of the evolutionary operator $\exp(\Delta t \mathcal{L})$ for a time step Δt is

$$\begin{bmatrix} U(\Delta t + t, \vec{x}) \\ V(\Delta t + t, \vec{x}) \end{bmatrix} = \exp(\mathcal{L}\Delta t) \begin{bmatrix} U(t, \vec{x}) \\ V(t, \vec{x}) \end{bmatrix} + \int_0^{\Delta t} \exp(\mathcal{L}(\Delta t - \tau)) \begin{bmatrix} F(\tau + t, x) \\ G(\tau + t, x) \end{bmatrix} d\tau. \quad (36)$$

When the inhomogeneous forcing functions F and $G = \partial F/\partial t$ are sampled at Δt , equation (36) is approximated by

$$\begin{bmatrix} U(\Delta t + t, \vec{x}) \\ V(\Delta t + t, \vec{x}) \end{bmatrix} = \exp(\mathcal{L}\Delta t) \begin{bmatrix} U(t, \vec{x}) \\ V(t, \vec{x}) \end{bmatrix} + \begin{bmatrix} F(t, x) \\ G(t, x) \end{bmatrix}. \quad (37)$$

For constant velocity, in the spatial Fourier domain the exponential of the operator is defined by equation (20). By adding $U(-\Delta t, \vec{k})$ to $U(\Delta t, \vec{k})$, the inhomogeneous cosine stepper is

$$\hat{U}(t + \Delta t, \vec{k}) = 2 \cos(2\pi\omega\Delta t) \hat{U}(t, \vec{k}) - \hat{U}(t - \Delta t, \vec{k}) + 2\hat{F}(t, \vec{k}). \quad (38)$$

LOCAL FOURIER TRANSFORMS

In the examples shown so far, the wavefield has been Fourier transformed with a global Fourier transform over the entire domain of the wavefield. To maintain a non-reflecting boundary condition on the sides and bottom, a taper is applied to the wavefield every 10th timestep. A free surface boundary condition is used at the surface by zeroing the wavefield at the bottom of the domain after each timestep.

Often the windows $\Omega_i(\vec{x})$ are spatially compact and the Fourier transform can be computed over a subset of the domain of $\Omega_i(\vec{x})$ or a set Fourier transforms with smaller domains. To prevent wrap-around of energy and to maintain boundary conditions it is necessary to ensure sufficient padding of each of the local Fourier transforms. As well, the windows cannot be too small or else the \vec{k} variable will be under sampled. The free surface boundary condition is maintained by zeroing the bottom of any local Fourier transform windows that touches the free surface.

In the case of the global Fourier transform it is possible to use $N+1$ FFTs to accomplish one timestep, where N is the number of windows. N FFTs were used to calculate the spectrum for each windowed wavefield and 1 FFT to go back into the space domain. When using local Fourier transforms each Fourier transform corresponds to a different domain and thus $2N$ FFTs are necessary. Figure 10 (a) is the image using local Fourier transforms, and Figure 10 (b) is the image of the salt dome using global Fourier transforms. Figure 10 (c) is one Gabor window where the value -1 indicates that is is not in the domain. Figure 10 (d) is a global Fourier window. Using local windows cut the computation time in half. For data sets like the Marmousi the windows are not spatially compact and so this method will not reduce computation time.

A MODELLING EXAMPLE

To investigate the ability of the PSTS equation to image multiples, equation (1) is used as a modelling algorithm. For a first approximation RTM smooths the velocity fields or

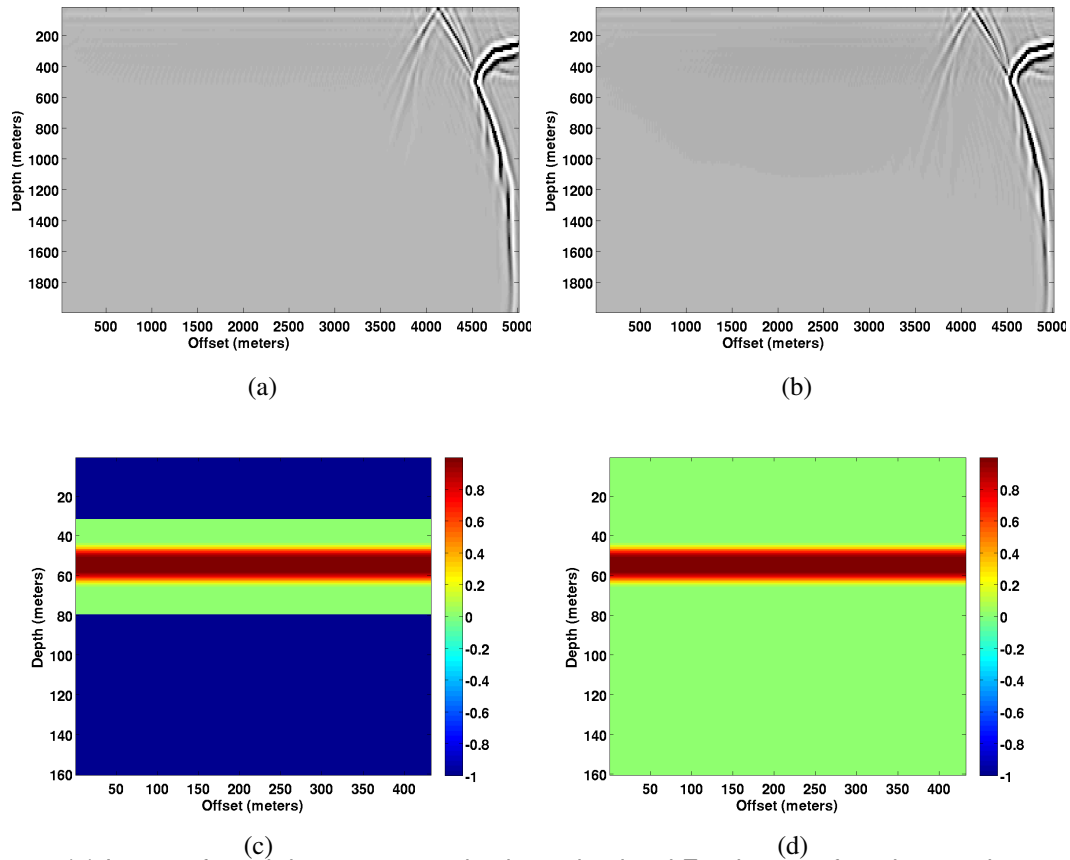


FIG. 10. (a) Image of a salt dome computed using using local Fourier transform in equation (5). (b) Image using Global Fourier transforms. (c) A local Gabor window. The value -1 in the image is not part of the domain of the local Fourier transform. (d) A Global Fourier transform window.

propagates with a nonreflecting wave equation. This is because there is too much backscatter for the imaging condition to handle. There is a certain class of multiples which image part of the structure that primary energy does not adequately illuminate that we may not want to suppress.

Despite the fact that equation (1) can be easily approximated by a Gabor decomposition of the velocity field, the locally homogeneous approximation or the freezing argument is not generally valid for adapting the constant velocity solution to a discontinuous velocity field. Equation (1) can be approximated by approximating the velocity field by a piecewise constant Gabor decomposition. Unlike RTM, the windows are not smoothed to suppress reflections.

Figure 11 are images of synthetic shotrecords used to compare finite difference modelling to PSTS modelling. The PSTS is non-dispersive and had correct kinematics when compared to the finite difference models. However the amplitudes and phases were not correct for PSTS modelling.

CONCLUSION

The PSTS equation is proposed as an alternative to finite differencing and pseudospectral methods to solve the wave equation. A number of numerical schemes are proposed to compute the PSTS. As well a scheme that propagates the wavefield and derivative of the wavefield is used. The computation speed of these methods is determined by the number FFTs used and the size of timestep.

ACKNOWLEDGMENTS

We thank the sponsors of the CREWES project, and those of the POTSI consortium, and especially NSERC, MITACS, PIMS, and Alberta Ingenuity.

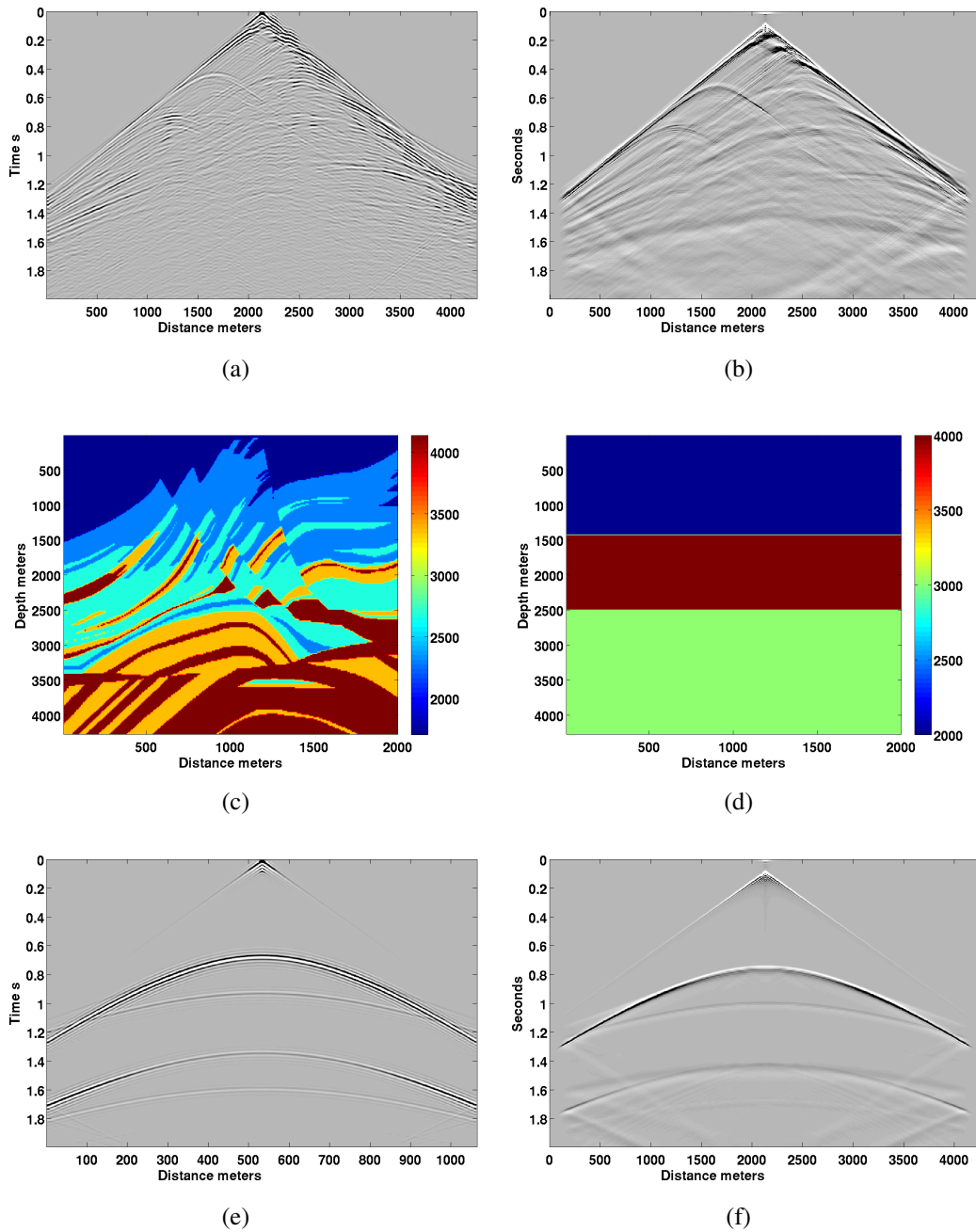


FIG. 11. (a) Shot record for velocity in (c) using finite differences. (b) Shot record using PSTS equation. (c) A simplified Marmousi velocity model with 4 distinct velocities. (d) A simple velocity model. (e) Shot record for the velocity in (d) using finite differences. (f) Shot record for the velocity in (d) using PSTS equation.

REFERENCES

- Baysal, E., Kosloff, D. D., and Sherwood, J. W. C., 1983, Reverse time migration: *Geophysics*, **48**, No. 11, 1514–1524.
- Etgen, J., 1989, Accurate wave equation modeling: SEP-60.
- Jones, I. F., Goodwin, M. C., Berranger, I. D., Zhou, H., and Farmer, P. A., 2007, Application of anisotropic 3d reverse time migration to complex north sea imaging: *SEG Technical Program Expanded Abstracts*, **26**, No. 1, 2140–2144.
- McMechan, G. A., 1983, Migration by extrapolation of time-dependent boundary values: *Geophysical Prospecting*, **31**, No. 3, 413–420.
- Tal-Ezer, H., 1986, Spectral methods in time for hyperbolic equations: *SIAM J. Numer. Anal.*, **23**, No. 1, 11–26.
- Versteeg, R., 1994, The Marmousi experience; velocity model determination on a synthetic complex data set: *The Leading Edge*, **13**, No. 9, 927–936.
- Wards, B. D., Margrave, G. F., and Lamoureux, M. P., 2007, High-fidelity time-stepping for reverse-time migration: *CREWES Research Report*, , No. 48.
- Wards, B. D., Margrave, G. F., and Lamoureux, M. P., 2008, Phase-shift time-stepping for reverse-time migration: the marmousi data experience: *CREWES Research Report*.