OPTIONS FOR SOURCE WAVEFIELD RECONSTRUCTION IN PRESTACK REVERSE-TIME MIGRATION

BAO NGUYEN and GEORGE A. MCMECHAN

Center for Lithospheric Studies, The University of Texas at Dallas, 800 W. Campbell Road, Richardson, TX 75080-3021, U.S.A.

(Received December 4, 2008; revised version accepted May 22, 2009)

ABSTRACT

Nguyen, B. and McMechan, G.A., 2009. Options for source wavefield reconstruction in prestack reverse-time migration. *Journal of Seismic Exploration*, 18: 305-314.

One of the main steps in prestack reverse-time migration is recovery of the source wavefield at each time step during the reverse-time extrapolation of the receiver wavefield. This recovery can be accomplished in a number of ways, each of which has it's advantages, disadvantages, and computational and data storage requirements. Two types of source wavefield recovery exist. The most common (and the most expensive) approach to source wavefield recovery is computing and saving the wavefield snapshots and accessing them in reverse-time order; the most economical approaches involve reconstruction of the source wavefield, at each time step, from boundary or initial conditions (or a combination of these). The advantages of reconstruction are most obvious in 3D, where the disk storage requirement is reduced by at least two orders of magnitude compared to saving all the time snapshots.

KEY WORDS: reverse-time migration, computational efficiency, image condition.

INTRODUCTION

Prestack reverse-time migration (RTM) operates on common-source gathers and involves three steps: extrapolation of the source wavefield (to construct the information needed for application of the image condition), reverse-time extrapolation of the receiver wavefield, and application of the imaging condition (at each time step in the receiver wavefield extrapolation).

0963-0651/09/$5.00

In the reverse-time extrapolation of the receiver wavefield, time slices of the wavefield act as time-dependent boundary conditions at the receiver points (Sun and McMechan, 1986). At each time step, the intersection/overlap between the source and receiver wavefields defines the imaging condition at that time. The source wavefield needs to be accessed in planes (for 2D) or volumes (for 3D) in reverse-time order. Thus, it is necessary to compute all the time steps in the forward direction before they can be recovered/accessed in the reverse-time direction. More complete descriptions of prestack RTM can be found in Chang and McMechan (1986), Loewenthal and Hu (1991) and Gray et al. (2001).

Various source wavefield recovery methods have been developed for RTM (Chang and McMechan, 1986; Sun and McMechan, 1986; Loewenthal and Hu, 1991; Xu et al., 1995; Zhang et al., 2007). Each implementation has its unique advantages and inherent disadvantages. The main tradeoff is between the quality and information content of the resulting image and the computational efficiency. Although the ideas of computing image conditions via excitation time or by saving the full wavefield over the entire grid have been known for decades, there are also alternative options that are less well known (and in some cases, not previously published), that give significant reductions in the computational resources required.

Yoon et al. (2004) summarizes the challenges in RTM and details a representative hardware specification and the parameters for 2D and 3D RTM. They found that for small 2D problems, frequency (ω) domain RTM may reduce the overall execution time (CPU time), but for larger 2D, and all 3D problems, efficiency becomes a major concern for implementations in the ω-domain as well as in the time domain. Those authors conclude that ω-domain RTM cannot be easily applied to large scale imaging problems as it is impractical in 3D (as of 2004). Those conclusions apply only to their, and related, implementations.

The subject of this paper is a comparison of five methods for source wavefield recovery in the time domain, where RTM has the benefit of being easily parallelized (Fricke, 1988; Yoon et al., 2004). Source wavefield reconstruction reduces the disk space requirement by orders of magnitude compared to the usual procedure of storing and recovering all the receiver wavefield snapshots.

OPTIONS FOR SOURCE WAVEFIELD RECOVERY

A brief summary of five algorithms for wavefield recovery is provided below. Tables 1 and 2 show the ideal usage of computational resources and the CPU operation count for each of the options described, for 2D and 3D implementations, respectively.

Table 1. 2D computational resource requirements. N_x and N_z are the numbers of grid points in the X-, and Z-directions and N_t is the total number of time steps. Abbreviations are used for the various implementation options; AGP is saving the entire source wavefield at all grid points, ATS is saving at all time steps, AGE is saving at all grid edges except the one where the receivers are located, and 2TS is saving at two time steps. The factors of two in the CPU Op Count column are a consequence of both forward- and reverse-time extrapolations being required.

Implementation	CPU Op Count	Disk Storage	I/O
Excitation Time	$N_xN_zN_t$	$2N_xN_z + 2N_t$	2 × Disk Space
AGP at ATS	$N_xN_zN_t$	$N_xN_zN_t$	2 × Disk Space
AGE at ATS	$2N_xN_zN_t$	$N_xN_t + 2N_zN_t$	2 × Disk Space
AGP at 2TS	$2N_xN_zN_t$	$2N_xN_z$ (RAM)	0
Hybrid Method	$2N_xN_zN_t$	$2N_zN_t + 2N_xN_z$ (RAM)	2 × Disk Space

Table 2. 3D computational resource requirements. N_x, N_y and N_z are the numbers of grid points in the X-, Y- and Z-directions and N_t is the total number of time steps. Abbreviations are used for the various implementation options; AGP is saving the entire source wavefield at all grid points, ATS is saving at all time steps, AGE is saving at all grid edges except the one where the receivers are located, and 2TS is saving at two time steps. The factors of two in the CPU Op Count column are a consequence of both forward- and reverse-time extrapolations being required.

Implementation	CPU Op Count	Disk Storage	I/O
Excitation Time	$N_xN_yN_zN_t$	$3N_xN_yN_z + 2N_t$	2 × Disk Space
AGP at ATS	$N_xN_yN_zN_t$	$N_xN_yN_zN_t$	2 × Disk Space
AGE at ATS	$2N_xN_yN_zN_t$	$2N_xN_zN_t + 2N_yN_zN_t$ + $N_xN_yN_t$	2 × Disk Space
AGP at 2TS	$2N_xN_yN_zN_t$	$2N_xN_yN_z$ (RAM)	0
Hybrid Method	$2N_xN_yN_zN_t$	$2N_yN_zN_t + 2N_xN_zN_t$ + $2N_xN_yN_z$ (RAM)	2 × Disk Space

Exitation time

The excitation time at any point in an image grid is the one-way traveltime from the source to that point, and corresponds to its image time. This method for recovery involves forward modeling of the propagating wavefield by either ray-tracing or a wave solution such as finite-differencing (Chang and McMechan, 1986; Loewenthal and Hu, 1991), and detecting when the wavefront arrives at each grid point (with either the minimum time or the maximum amplitude).

When based on ray-tracing, no amplitude information is saved. The data that are stored result in a file that is very small and very fast to read, as it only contains the locations of the points that are imaged at each time step (Fig. 1b). Finite-differencing has the benefit of producing accurate incident amplitudes (which are, in this context, not used), and requires solving a two-way wave equation, which is expensive compared to ray tracing. The main disadvantage of using the excitation time to define the source wavefield is that it precludes recovery of accurate amplitudes in the migration. Application of the excitation time image condition involves reading the locations of the points that are imaged at each time step during extrapolation of the receiver wavefield (e.g., Chattopadhyay and McMechan, 2008).

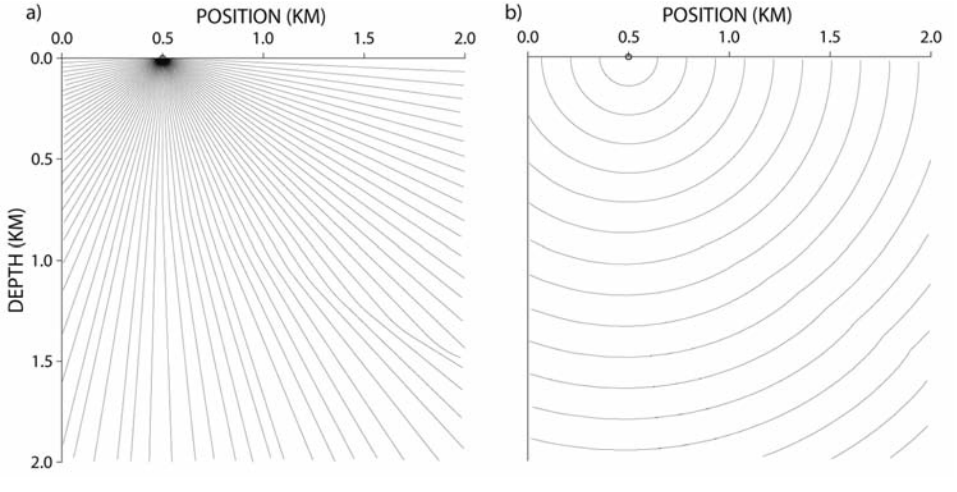

Fig. 1. (a) Rays from a primary source point. (b) Constant time loci (wavefronts) consisting of groups of points that satisfy the imaging condition at their respective times.

Saving the entire source wavefield over all grid points for every time step

This option stores the complete source wavefield at every grid point at every time step and is thus the most expensive in terms of the required storage, but it is also still the most commonly used in industry today (Zhang et al., 2007). It involves saving the snapshots of the forward propagating wavefield at all grid points of the computational mesh over all time steps (Fig. 2) which, in 3D, is a store of a 4D wavefield (Zhang et al., 2007).

To avoid I/O overhead in 2D, the forward wavefields could be stored in RAM and accessed in reverse-time order during migration, provided enough RAM is available (Yoon et al., 2004). In most machines, the forward wavefields are saved on disk (because of the large accumulated size of the snapshot files), and are read in reverse-time order (one at each time step during the receiver wavefield extrapolation). The most common form of image condition used in this context is cross-correlation of the source and receiver wavefields (Whitmore and Lines, 1986; Chattopadhyay and McMechan, 2008).

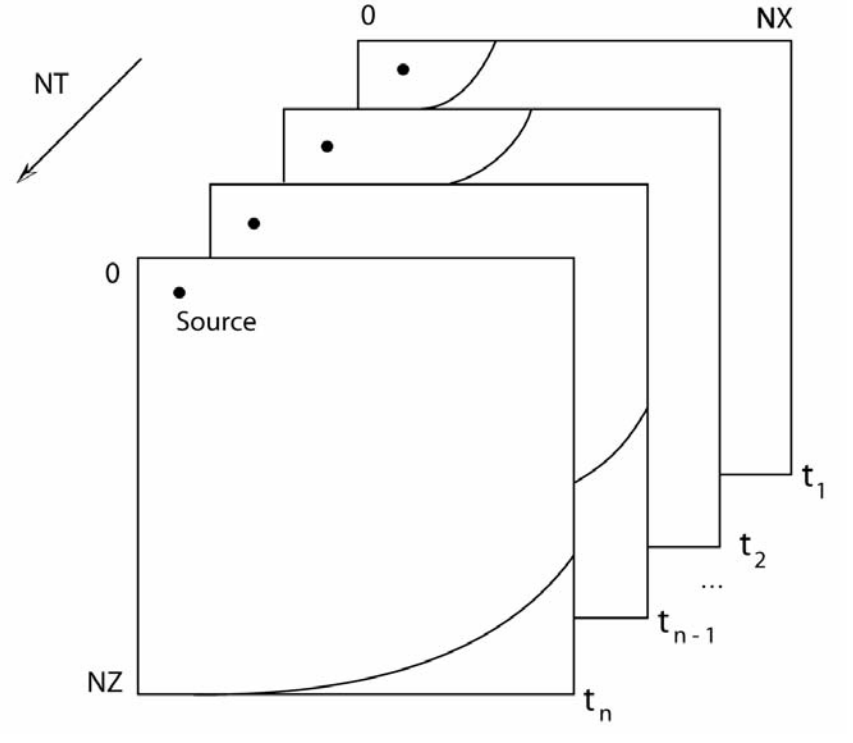

Fig. 2. A source wavefield computed and stored in the forward time direction and then recovered in reverse-time order. In 3D, the 2D slices shown here would become 3D volumes (one for every time step t). NX and NZ are the grid dimensions in the X and Z directions, and NT is the number of time steps.

The I/O involved in writing and subsequent reading of the wavefield snapshots is very slow compared to the computations but can be, at least partly, overlapped in time with the wavefield extrapolation computations in a parallel machine. The limiting factor is the very large size of the saved snapshots, especially in 3D. The size of the snapshot files can also be reduced by reducing the grid dimensions by using high-order finite-difference or pseudo spectral extrapolations.

Saving source wavefield only at grid edges for every time step

Xu et al. (1995) describe a method for reconstructing the source wavefield by saving constant-time wavefield slices (for 2D) or volumes (for 3D) corresponding to the model bottom and sides (Fig. 3) during forward source extrapolation. They then reconstruct the source field interior to the computational grid at each time step by reverse-time extrapolation using the saved data as time-dependent boundary conditions, as is usually done for the

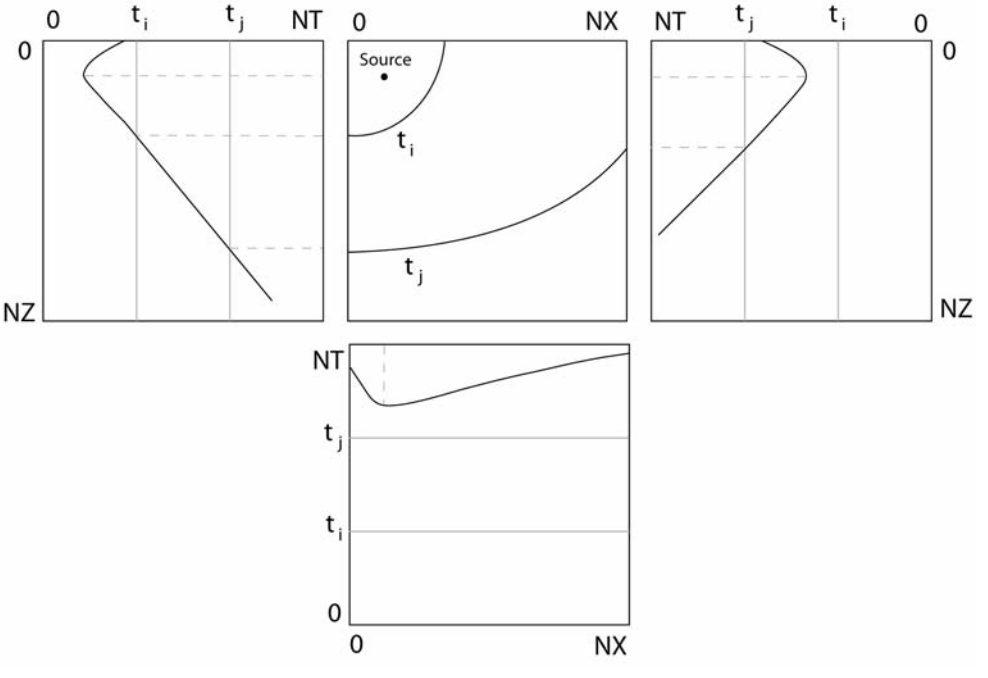

Fig. 3. Saving the wavefield amplitudes only at the grid edges, during extrapolation over all time steps. These time slices serve as time-dependent boundary conditions to reconstruct the source wavefield during reverse-time extrapolation.

receiver field (McMechan, 1983). Although the processing involves three wavefield extrapolations (forward and backward for the source wavefield, and backward for the receiver wavefield), it reduces disk requirements in 3D, by several orders of magnitude, compared to computing, writing, and reading the whole source wavefield at every time step. For reconstruction, only three planes of data (for 2D) or five volumes (for 3D) are stored [compared to NT planes (for 2D) or NT volumes (for 3D), where NT is the number of time steps] when the entire wavefields are stored.

Saving two contiguous time steps over all grid points

This method does not seem to have been previously published, but is documented by de Faria (1986). It requires saving the forward propagating wavefield over all grid points for just two adjacent time steps (Fig. 4). Doing so avoids any disk space or I/O usage by storing these two wavefields directly in RAM. The forward extrapolation is stopped before the source wavefront exits the grid and two adjacent time steps are saved over the entire grid. The time indices of these two wavefields are swapped and then the wavefields are used

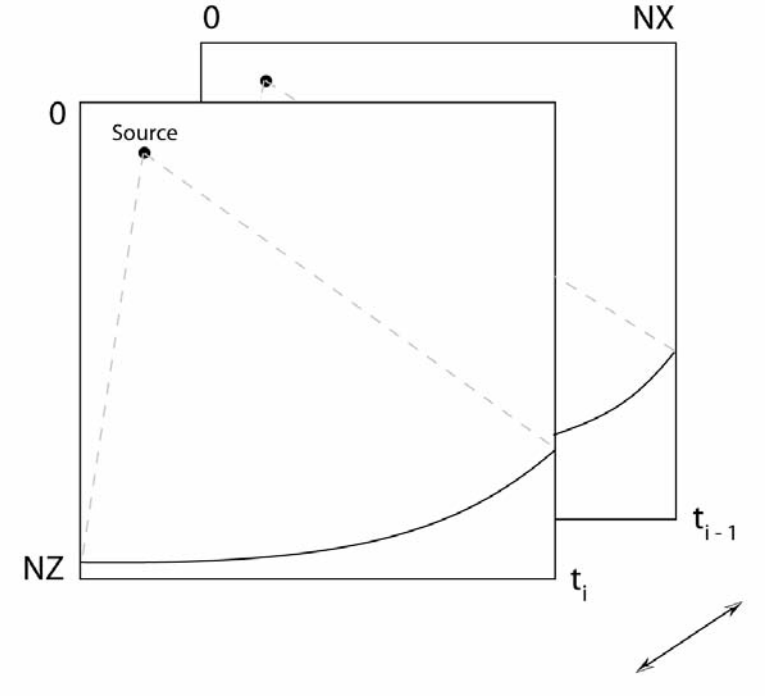

Fig. 4. Wavefronts at two adjacent time steps are saved during the source wavefield propagation, before the source wavefront exits the grid. Interchanging these two snapshots then provides initial conditions for reverse-time extrapolation.

as initial values to propagate backward in time to reconstruct the source wavefield. Two wavefields are the minimum number required, and correspond to using a centered finite-difference star of second-order accuracy for the second derivatives with respect to time.

This method is the least demanding of disk space (it uses none!). The main disadvantage is, that for a single source, the wavefield can be reconstructed only within the pie shaped area in Fig. 4, and hence only that region can be migrated. Other parts of the structure need to be filled in by data corresponding to illumination by other sources.

Hybrid implementation

We create a new hybrid source wavefield reconstruction option by combining the third and fourth methods described above. This algorithm requires saving the forward propagating wavefield along the grid edges, and also stopping the extrapolation at some time t_i before the wavefield exits the grid bottom (Fig. 5). The reconstruction then involves extrapolating a mix of initial and boundary values, as described above.

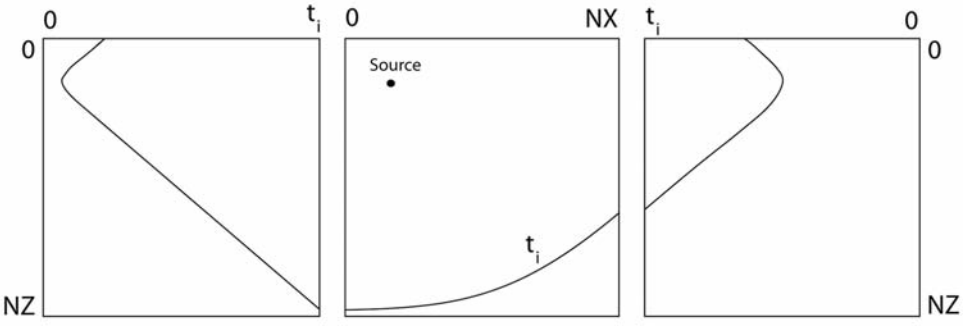

Fig. 5. Stopping the source wavefield extrapolation at some time t_i (center panel) before the wavefront exits the grid (as in Fig. 4), and saving the wavefield values during forward propagation along the right and left edges for all time steps (from 0 to t_i) (as in Fig. 3). These data provide initial (center panel) and boundary conditions (left and right panels) for reverse-time reconstruction of the source wavefield.

COMPUTATIONAL REQUIREMENTS

Our aim in this study is to quantify and compare the requirements of various algorithms for wavefield reconstruction, as a basis for determination of

which algorithms work best under specific conditions and algorithmic and hardware limitations. Tables 1 and 2 list each option, and their associated costs, in terms of computational resources on the basis of the variables N_x, N_y, and N_z (the number of grid points in each direction, and N_t (the total number of time steps).

The amount of CPU used per 'operation' depends on the specific form of the solution used (e.g., the order of the finite-difference computational star, or the type of Fourier transform used in a pseudospectral solution). An 'operation' is the computation associated with solving the wave equation at one grid point. For both 2D and 3D, the number of CPU operations needed for extrapolation (for all times) for each algorithm are essentially the same. The input and output (I/O) are directly related to the disk storage requirements by a factor of two for all of the five implementation options as the saved data are both written and read (only the option of saving two contiguous timesteps has no I/O). For elastic extrapolations, two wavefield components, rather than one, need to be saved and reconstructed in 2D, and three in 3D.

Consider a specific example of the storage requirements. For a 2D finite-difference grid of 800 x 500 points and 1500 time steps, saving all the time snapshots as 32 bit floating point numbers requires $800 \times 500 \times 1500 \times 4$ bytes = 2.4 GB per source. Saving the boundary values at the bottom and two sides of the grid requires $[(800 \times 1500) + (2 \times 500 \times 1500)] \times 4$ bytes = 10.8 MB per source. The latter is a factor of 220 smaller than the former. For a 3D grid of $800 \times 800 \times 500$ points and again 1500 time points, saving snapshots for one source requires 1.9 TB; saving grid boundary values at the four edge planes and the bottom requires 3.84 GB; the latter is a factor of 500 smaller. For the special case of $N = N_x = N_y = N_z$, the factor is N/3 in 2D and N/5 in 3D. Certainly various data compressions can be used, but compression and decompression also take CPU resources. Compression reduces the absolute, but not the relative storage needed.

Note that the relative disk storage requirements are independent of the number of time steps as N_t cancels when taking the ratio (see Tables 1 and 2). When using low-order finite-difference extrapolations, the absolute storage can be reduced by saving every n-th, rather than every, time step, but the Nyquist sampling criterion still needs to be satisfied to avoid aliasing.

Considering the spectrum of algorithms, there is a trade-off between CPU resources (time and memory), and slow disk storage; to some extent, one can be substituted for the other. The actual situation is hardware dependent. The main advantage is gained by reconstructing, rather than saving, the source wavefield.

CONCLUSIONS

A quantitative comparison of implementation options provides a better understanding of which methods are more efficient or suitable for different requirements (such as image position only vs. accurate amplitudes). Saving the entire source wavefield at every time step is not justified as it is significantly more taxing than the other options regarding storage requirements, and should be avoided. The main trade-off is whether one should compute and store the entire source wavefield for all time steps, or reconstruct the wavefield from boundary and/or initial values; examination of this trade-off favors reconstruction.

ACKNOWLEDGMENTS

This study was supported by the sponsors of the UT-Dallas Geophysical Consortium, and by the State of Texas Norman Hackerman Advanced Research Program under grant 009741-0021-2007. This paper is Contribution No. 1192 from the Department of Geosciences at the University of Texas at Dallas.

REFERENCES

Chang, W. and McMechan, G.A., 1986. Reverse-time migration of offset vertical seismic profiling data using the excitation-time imaging condition. Geophysics, 51: 67-84.

Chattopadhyay, S. and McMechan, G.A., 2008. Imaging conditions for prestack reverse-time migration. Geophysics, 73: S81-S89.

de Faria, E., 1986. Migração antes do empilhamento utilizando propagação reversa no tempo. M.Sc. Thesis, Universidade Federal da Bahia, Brazil.

Fricke, J.R., 1988. Reverse-time migration in parallel: a tutorial. Geophysics, 53: 1143-1150.

Gray, S.H., Etgen, J., Dellinger, J. and Whitmore, D., 2001. Seismic migration problems and solutions. Geophysics, 66: 1622-1640.

Loewenthal, D. and Hu, L.Z. 1991. Two methods for computing the imaging condition for common-shot prestack migration. Geophysics, 56: 378-381.

McMechan, G.A., 1983. Migration by extrapolation of time-dependent boundary values. Geophys. Prosp., 31: 413-420.

McMechan, G.A., 1989. A review of seismic acoustic imaging by reverse-time migration. Internat. J. Imaging Syst. Technol., 1: 18-21.

Sun, R. and McMechan, G.A., 1986. Pre-stack reverse-time migration for elastic waves with application to synthetic offset vertical seismic profiles. Proc. IEEE, 74: 457-465.

Whitmore, N.D. and Lines, L.R., 1986. Vertical seismic profiling depth migration of a salt dome flank. Geophysics, 51: 1087-1109.

Xu, T., McMechan, G.A. and Sun, R., 1995. 3-D prestack full-wavefield inversion. Geophysics, 60: 1805-1818.

Yoon, K., Marfurt, K.J. and Starr, W., 2004. Challenges in reverse-time migration. Expanded Abstr., 74th Ann. Internat. SEG Mtg., Denver: 1057-1060.

Zhang, Y., Sun, J. and Gray, S., 2007. Reverse-time migration: amplitude and implementation issues. Expanded Abstr., 77th Ann. Internat. SEG Mtg., San Antonio: 2145-2149.