Analyzing the time saving of microseismic finite differences modeling when using an expanding box algorithm

Germán I. Brunini* (UNLP, CONICET), Juan I. Sabbione (UNLP, CONICET) and Danilo R. Velis (UNLP, CONICET)

This paper was prepared for presentation during the 17th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 16-19 August 2021.

Abstract

One of the mayor drawbacks of waveform propagation using finite differences (FD) is the intense computational effort required to solve the wave equation. The computational cost is directly proportional to the number of grid points of the discretized medium, and may become prohibitive for large scale models and fine grid meshes. This motivated the development of strategies to reduce the computational cost of forward modeling, such as the so-called expanding box (EB) methods. An EB method relies on solving the involved equations within a growing area that encloses the perturbed grid points only, which is delimited by the source and the expected most-advanced wavefront. As the wavefield propagates, this area expands with time until all the grid points are perturbed encompassing the whole space domain. The method avoids unnecessary calculations where the wavefield has not reached yet. In this context, we evaluate the impact of using an EB algorithm when modeling microseismic data, where frequencies are high and the required sampling interval is very small. We consider a FD elastodynamic wave equation modeling and typical microseismic borehole monitoring scenarios. We analyze and evaluate the expected execution time saving of the EB case in comparison with the conventional FD method by means of an homogeneous 2D model and typical microseismic source-receiver geometries, taking into account both P and SV-wave phase arrivals. Results using Julia programming language show that execution time saving can be as high as 30%.

Introduction

Forward seismic modeling is a powerful tool for many seismic applications, as it allows studying how media properties affect the waveform propagation. In microseismic studies, waveform modeling has been widely used for various mining and hydrocarbon exploration applications. For instance, Usher et al. (2013) use a finite difference (FD) method to examine the influence of both the velocity model and the source frequency on waveforms and event locations. Roussel et al. (2011) propose a poroelastic 3D model to simulate the interaction between fractures in horizontal well-bores in order to obtain optimum fracture spacing values and investigate the fracturing sequence. Folesky et al. (2015) present a microseismic rupture propagation imaging method based on a FD modeling of a synthetically generated seismogram. Also, Meek et al. (2015) model seismic waves with different source mechanisms with a 3D elastic FD algorithm to determine the sensitivity of microseismic attributes, underlying geology parameters, and array configuration. Both Pike (2014) and Rodríguez-Pradilla and Eaton (2018) show extensive and detailed microseismic studies and interpretations based on FD modeled data. In another application, Vavryčuk and Kühn (2012) make use of a 3D FD viscoelastic code to perform a two-step time-frequency moment tensor inversion in a real mining environment.

Using an explicit FD technique to simulate a wavefield propagating through the underlying media commonly requires solving the elastodynamic wave equation in the space-time domain. The model is usually regarded as a box in 2D, or as a cube in 3D, which must enclose the source and the geophones locations. In time, the signal is generally calculated from the source origin time and propagated until the perturbation reaches the whole sensor array. Typical source-receiver distances can vary from tens to hundreds of meters. The total time of the simulations, on the other hand, depends on the acquisition geometry and velocity model. Usually, because in microseismic studies frequencies are higher than those of conventional seismic processes, proper microseismic modeling requires a fine sampling in both the time and space domains. For these reasons, the intense computational effort required to solve the elastodynamic wave equations becomes one of the major drawbacks. In other words, microseismic forward modeling involves very fine grids and small sampling intervals that lead to very high computational costs, which in some projects may became prohibitive, specially for 3D scenarios.

In order to reduce the computational times, we can seek to adjust the model to the smallest possible size that still encloses the perturbed part of the medium. Unfortunately, this technique may lead to an undesirable effect, which is caused by the wavefront reaching the mesh boundary and reflecting back into the modeled space, eventually masking the signal of interest. The simplest way to avoid this interference is enlarging the mesh size, which delays the border effect to larger times, but of course at the expense of an increased computational cost. To tackle with this issue, many computational techniques have been developed in order to mitigate these artificial reflected waves without the need of increasing the grid size. The most common approaches are the so-called absorbing boundary conditions (ABS) (Clayton and Engquist, 1977; Stacey, 1988) and, more recently, the perfectly matched layers (PML) (Berenger, 1994; Komatitsch and Martin, 2007). Regardless of the choice of method for dealing with the borders, in many applications is still needed to decrease the computational cost that is intrinsically associated with the space domain size.

In this work, we evaluate the performance a simple expanding box (EB) algorithm that considerably reduces the computational costs in discrete forward modeling by taking advantage of the causality nature of the problem. In what follows, we describe the EB method and show the execution time saving that can be expected in the case of a 2D homogeneous model of various sizes which are typical in microseismic monitoring scenarios. The results show that significant execution times can be saved by using a dynamically adaptive model space at each time step of the FD solution. The method can be readily extended to 3D scenarios and multi-layered heterogeneous media.

Theory

Without loss of generality, we expose the principles of the EB strategy using the 2D first-order hyperbolic formulation of the elastodynamic problem. Readers are referred to Madariaga (1976) or Virieux (1986) for details on this formulation. Typical methods for solving these equations by means of a time-explicit FD technique involve updating, over the entire discretized grid and for each time step, the unknown velocity field, $\mathbf{v}$=(v_x, v_z), and the stress tensors, σ=(σ_{xx},σ_{zz},σ_{xz}). At any given time step, taking into account the causality nature of the problem, it is clear that the medium can only be excited within the region delimited by the wavefront. That is, for any given time step, any point of the discretized medium that lies beyond the current wavefront, remains unperturbed.

Expanding box algorithm

At each time step, it seems reasonable to update the wavefield only within a region (box) that encloses the wavefront that would be generated by a wave propagating with the fastest velocity of the medium. As time increases, the wavefront propagates outwards from the source point and, therefore, the box must expand. This strategy continues until the expanding box (EB) reaches all boundaries of the grid. This EB can be designed in a simple fashion. Even for the most complex media, we can assume that at any time t, a wavefront moving away from a point source is entirely contained within a circle (or sphere in 3D) of radius $r(t) = v_{max}\, t$, where v_{max} is the fastest velocity of the medium. Thus, it suffices to define the EB such that the half-length of its sides equal to $d(t) = r(t) + r_0$, where r_0 is an initial distance set to prevent the wavefront to be in contact with the box limits and to avoid undesired boundary artifacts.

For simplicity, consider an 2D homogeneous medium with velocity v, uniformly discretized into $N_x \times N_z$ grid points with $\Delta x = \Delta z$, and a source located at the grid node (I_S, J_S). The grid points corresponding to the left and right sides of the EB at time t can be determined using:

$$\begin{aligned} I_L(t) &= \max\left\{1, I_s - \left\lfloor \frac{d(t)}{\Delta x} \right\rfloor\right\} \quad \text{and} \\ I_R(t) &= \min\left\{N_x, I_s + \left\lfloor \frac{d(t)}{\Delta x} \right\rfloor\right\}, \end{aligned} \tag{1}$$

where the subscripts L and R refer to the left and right sides, respectively, and $\lfloor\cdot\rfloor$ takes the integer part of its argument. Likewise, similar expressions can be drawn for the sides expanding upwards and downwards, which we write as $I_U(t)$ and $I_D(t)$. It should be noted that the EB is not restricted to be square. One may choose to build a rectangular EB by considering different velocities in the x and z directions, or simply by setting $\Delta x \neq \Delta z$. In the Examples section we will show an example of a square EB containing the P and SV-wavefronts at every time-step for a particular case.

CPU execution time

If we define T_{op} to be the time taken by the CPU to perform all the required operations to update the velocity and stress for a single grid point at one single time-step, the total time for the whole grid at any given time-step can be estimated by means of $T(t) \simeq N_x\, N_z\, T_{\text{op}}$. In the EB method, we can get an estimate of this time by means of

$$\hat{T}(t) = \hat{N}_x(t)\, \hat{N}_z(t)\, T_{\text{op}}, \tag{2}$$

where $\hat{N}_x(t) = I_R(t) - I_L(t) \leq N_x$ and $\hat{N}_z(t) = I_U(t) - I_D(t) \leq N_z$ are the width and height of the EB at time t, respectively. Clearly, $\hat{T}(t) \leq T(t)$, leading to CPU time savings.

Equation 2 is a time dependent function that is quadratic in time, since both $\hat{N}_x(t)$ and $\hat{N}_z(t)$ grow linearly with time. The quadratic behavior holds until either the width or the height of the EB (but not both) stops growing (i.e. either $\hat{N}_x(t) = N_x$ or $\hat{N}_z(t) = N_z$). At his point, $\hat{T}(t)$ grows linearly with time. As soon as the EB cannot grow anymore in any one direction (i.e. $\hat{N}_x(t) = N_x$ and $\hat{N}_z(t) = N_z$), $\hat{T}(t)$ becomes constant and equal to $T(t)$.

Therefore, the CPU time saving at any given time-step t can be estimated by means of

$$\Delta T(t) = T(t) - \hat{T}(t). \tag{3}$$

By integrating $T(t)$ and $\hat{T}(t)$ from $t = 0$, we can obtain the corresponding total CPU times at time t. In particular, we are interested in computing the relative CPU time saved when the EB method is used instead of the standard FD scheme that evaluates all grid points at every time-step. This value can be estimated by means of

$$S(t) = \frac{\sum_{t=0}^{t} \Delta T(t)}{\sum_{t=0}^{t} T(t)} \times 100\%. \tag{4}$$

The same idea presented for a 2D scenario can be extended to a rectangular EB for any non-homogeneous media, such as, for example, a 3D anisotropic layered media.

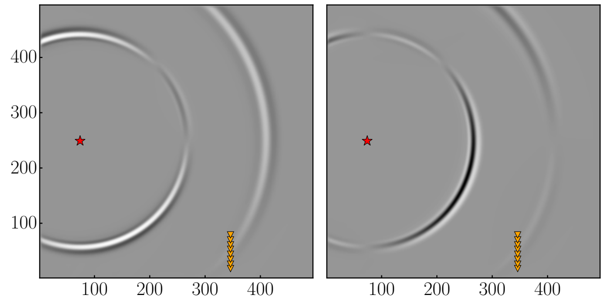

***Figure 1** – Snapshots of the $v_x(x,z)$ (left) and $v_z(x,z)$ (right) velocity fields for a model of size 500×500 showing the P- and SV-wavefronts. The red star denotes the source and the orange triangles denote the receivers.*

Examples

In this section we evaluate the CPU time saving that can be obtained using the EB algorithm by means of a 2D homogeneous isotropic model and a source-receiver geometry that resembles a typical microseismic borehole array monitoring scenario. We set the compressional and shear velocities $c_p = 4000$ m/s and $c_s = c_p/\sqrt{3} \simeq 2300$ m/s, respectively, and a density $\rho = 2700$ kg/m^3. For the source, following Komatitsch and Martin (2007), we use the derivative of a Gaussian pulse with a dominant frequency $f_0 = 100$ Hz and shifted by $t_0 = 1.2/f_0$ from $t = 0$ to guarantee null initial conditions. The point source is a velocity vector oriented 135° in the (x,z) plane and placed at $(I_S, J_S) = (0.15\,N, 0.5\,N)$, where $N = N_x = N_z$ for the sake of simplicity. We use a conservative uniform mesh size with $\Delta x = \Delta z = 0.5\,c_s/(10\,f_0) \sim 1.15$ m. Since we are using a time explicit FD technique, the Courant-Friedrichs-Lewy stability condition (Courant et al., 1928) demands a time step $\Delta t_{cfl} \leq \Delta x/(\sqrt{2}c_p)$. Thus, to be safe, we set $\Delta t = 0.9\,\Delta t_{cfl} \simeq 0.18$ ms.

As for the boundary conditions, we use the convolutional perfectly matched layer (C-PML) on all four sides of the grid, following the recommendations given by Komatitsch and Martin (2007) for a 2D model. We consider the Dirichlet condition on the velocity vector ($\mathbf{v} \equiv 0$) for the external edges of the model. Due to causality and because we assume an initial state of equilibrium with $\mathbf{v} = 0$, the limits of the box ahead of the wavefront remain unperturbed at any time and, so, we also impose Dirichlet condition over the EB edges.

For the analysis, we consider eight different model sizes $N \times N$, with N varying from 500 to 1200, in steps of 100 grid points. These model sizes also allow us to test different source-receiver distances. In all cases, we place a vertical receiver array at $(x_j, z_j) = (0.7\,N\Delta x, 20 + (j-1)\Delta r_z)$, $j = 1 \ldots N_r$, with $N_r = 8$ and $\Delta r_z = 10$ m.

Figure 1 shows snapshots of the v_x and v_z velocity fields at some time step for the model of size 500×500, where the P- and the SV-wavefronts are clearly visible. For the same model, Figure 2 shows the corresponding traces recorded by the geophone array. We repeated the simulations for the eight model sizes considered, using an appropriate number of time-steps so that both the P- and SV-waveforms are fully observed at the furthest receiver in all cases. We guarantee this by estimating the minimum number of time-steps required for the largest model, where the source-receiver distance is about 980 m.

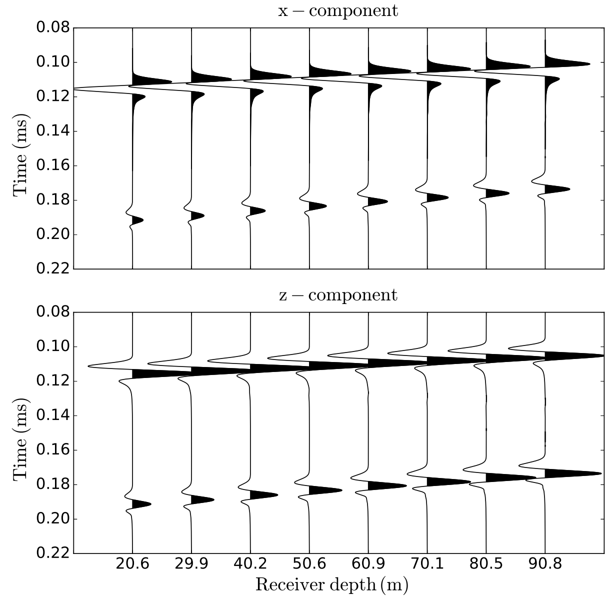

***Figure 2** – Traces recorded by the geophone vertical array for a model of size 500×500, showing the P- and SV-wave phases. Depth increases upwards with $z = 0$ on the grid bottom.*

Figure 3 shows the normalized execution time

$$\hat{T}_{norm}(t) = \frac{\hat{T}(t)}{\max_t\{\hat{T}(t)\}} = \frac{\hat{T}(t)}{T(t)} \qquad (5)$$

for the considered models as a function of iteration (time-steps). We observe four separate intervals where $\hat{T}(t)$ exhibits different behaviors, depending on how the EB grows in relation with the model boundaries (see Figure 4): (1) For $t \in [0, t_1)$ the curve is quadratic, where t_1 is the time required for the left side of the EB to reach the left model boundary (i.e. $I_L(t) = 1$). In this interval, neither $I_R(t)$ and $I_L(t)$, nor $I_U(t)$ and $I_D(t)$ reached the model boundaries; (2) For $t \in [t_1, t_2)$, where t_2 is the time required for the upper and lower sides of the EB to reach the upper and lower model boundaries, the curve is quadratic, too, but less pronounced because the area of the EB grows more slowly. Note that for this particular case, since the source is placed at $J_s = 0.5\,N$, both $I_U(t)$ and $I_D(t)$ hit the model boundaries simultaneously; (3) For $t \in [t_2, t_3)$, only $I_R(t)$ continues to expand until it reaches the right boundary at time t_3 (i.e. $I_r(t) = N$), and so $\hat{T}(t)$ exhibits a linear behavior; (4) As expected, when $I_R(t)$ reaches the right boundary at time t_3 , $\hat{T}(t)$ takes a constant value equal to $T(t)$ for all subsequent time-steps. The described situations represent a simple case that illustrates how the EB grows and how $\hat{T}(t)$ behaves. Naturally, the actual behavior of $\hat{T}(t)$ in a different scenario will depend on the velocity, the geometry/size

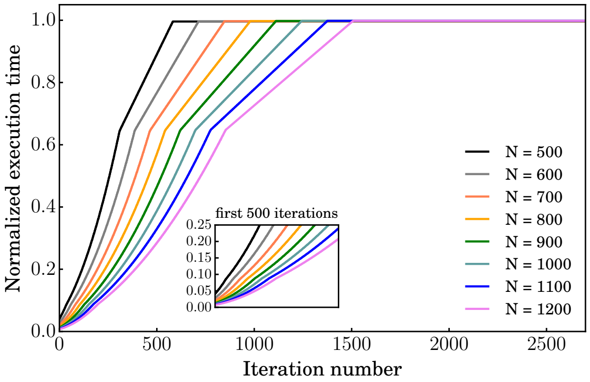

***Figure 3** – Normalized CPU times for various model sizes.*

of the model, and the relative position of the source with respect to the model boundaries.

Figure 5 shows the relative time saving $S(t)$ for all the considered model sizes. Each simulation was run from $t = 0$ until the slowest wave (i.e. the SV-wave) was fully observed at all the geophones. The smallest model required 1200 iterations while the largest model required about 2500. We observe that the EB algorithm leads to CPU time savings in the range of 20-30% with respect to a standard FD technique. Note that the larger the model, the more efficient the EB becomes for the same number of iterations. As expected, $S(t)$ has an asymptotic behavior towards zero. In light of that, the benefits of the EB method dilutes if the interest of the study is centered on modeling later phase arrivals such as multiples or diffracted waves that may arise in complex heterogeneous models. On the other hand, should the modeling include only P-waves, the CPU savings might be significantly larger than those observed in Figure 5, for the required number of time-steps would be much smaller.

Finally, Figure 6 shows the normalized CPU time $\hat{T}_{norm}(t)$ and the actual execution time for the case with $N = 500$ using a standard PC and Julia programming language as measured by means of the *time()* function. For a better comparison, we also normalized this time by $T(t)$. As expected, both curves are very similar. The small discrepancies observed in Figure 6 can be attributed to internal processes of the machine that are not captured by the *time()* Julia function.

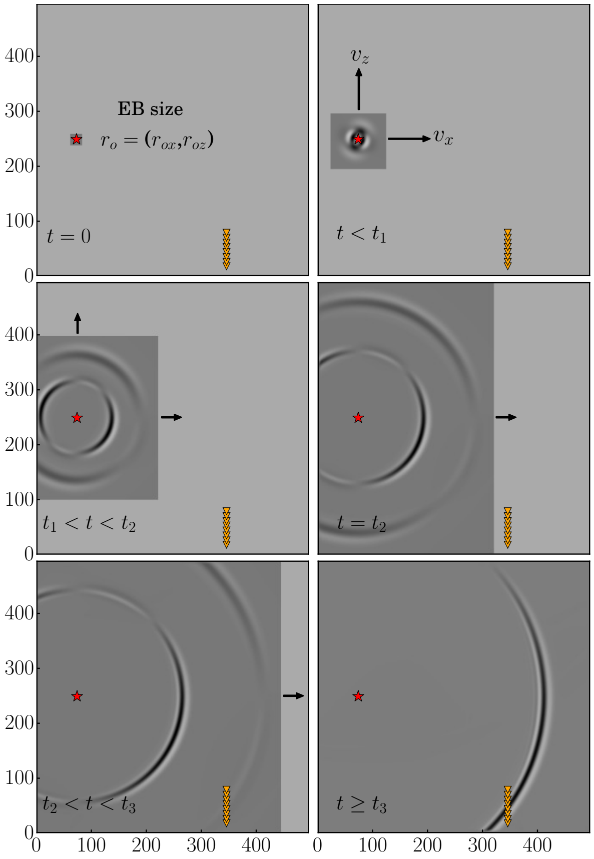

***Figure 4** – Propagation of the EB for a model of size 500×500 and a source placed at the grid point $(75, 250)$ (red star). As the box expands, it reaches the model boundaries at different times (see text for details). At each iteration, the velocity and stress need to be updated only within the dark-gray regions, for the wavefield is expected to be null beyond this area. For illustrative purposes, the orange triangles denote the receivers.*

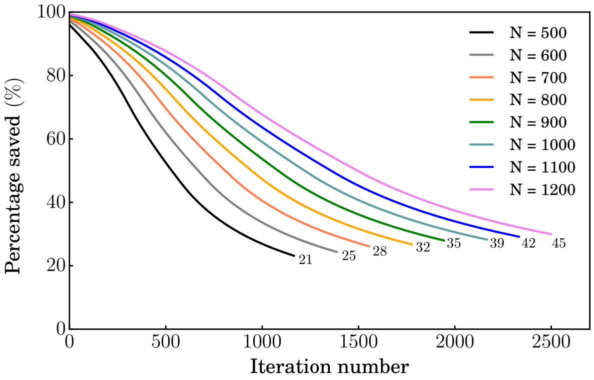

***Figure 5** – $S(t)$ for all the considered model sizes. The number at the end of each curve indicates the times (in ms) until equation 4 was computed.*

Conclusions

The adaptive expanding model space method studied in this work shows considerable execution time saving for FD modeling of a microseismic source propagating under null Dirichlet boundary conditions. The relative improvement on the computation times depends on the model size and number of time iterations considered. For typical microseismic geometries, execution time saving can be up 30% when compared against wave propagation FD modeling that considers the whole space domain. The theoretical execution time savings are in great agreement to those obtained by running a FD modeling Julia code with the same parameters used for the theoretical computations. The expanding model space method should be a routine technique for any microseismic FD modeling application where computational costs are to be considered.

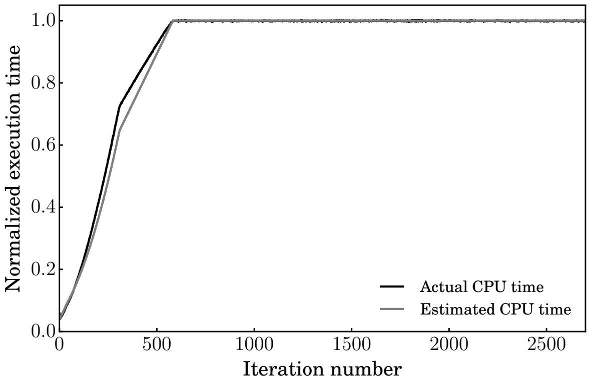

Figure 6 *– Estimated and actual CPU times for* $N = 500$.

Acknowledgements

We thank Universidad Nacional de La Plata, Argentina, for partial funding of this research (Research Grant I+D G163).

References

BERENGER, J.-P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: Journal of computational physics, 114, 185–200.

CLAYTON, R., and B. ENGQUIST, 1977, Absorbing boundary conditions for acoustic and elastic wave equations: Bulletin of the seismological society of America, 67, 1529–1540.

COURANT, R., K. FRIEDRICHS, H. LEWY, 1928, Über die partiellen differenzengleichungen der mathematischen physik: Mathematische annalen, 100, 32–74.

FOLESKY, J., J. KUMMEROW, S. A. SHAPIRO, 2015, Microseismic rupture propagation imaging: Geophysics, 80, WC107–WC115.

KOMATITSCH, D., R. MARTIN, 2007, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation: Geophysics, 72, SM155–SM167.

MADARIAGA, R., 1976, Dynamics of an expanding circular fault: Bulletin of the Seismological Society of America, 66, 639–666.

MEEK, R.A., R. HULL*, A. VON DER HOYA, D. EATON, 2015, 3-d finite difference modeling of microseismic source mechanisms in the wolfcamp shale of the permian basin: Unconventional Resources Technology Conference, San Antonio, Texas, 20-22 July 2015, Society of Exploration Geophysicists, American Association of Petroleum..., 1550–1555.

PIKE, K. A., 2014, Microseismic data processing, modeling and interpretation in the presence of coals: A falher member case study: Master's thesis, Graduate Studies.

RODRÍGUEZ-PRADILLA, G., D. W. EATON, 2018, Finite-difference modelling of microseismicity associated with a hydraulic-fracturing stimulation in a coalbed methane reservoir: First Break, 36, 41–48.

ROUSSEL, N. P., M. M. SHARMA, et al., 2011, Strategies to minimize frac spacing and stimulate natural fractures in horizontal completions: Presented at the SPE annual technical conference and exhibition, Society of Petroleum Engineers.

STACEY, R., 1988, Improved transparent boundary formulations for the elastic-wave equation: Bulletin of the Seismological Society of America, 78, 2089–2097.

USHER, P., D. ANGUS, J. VERDON, 2013, Influence of a velocity model and source frequency on microseismic waveforms: some implications for microseismic locations: Geophysical Prospecting, 61, 334–345.

VAVRYČUK, V., D. KÜHN, 2012, Moment tensor inversion of waveforms: a two-step time-frequency approach: Geophysical Journal International, 190, 1761–1776.

VIRIEUX, J., 1986, P-sv wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, 51, 889–901.