Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 543540, 8 pages
http://dx.doi.org/10.1155/2015/543540

*Research Article*

# Hybrid Kinematic-Dynamic Approach to Seismic Wave-Equation Modeling, Imaging, and Tomography

**Alexandr S. Serdyukov**[1] **and Anton A. Duchkov**[1,2]

[1] *Institute of Petroleum Geology and Geophysics SB RAS, Pr. Ac. Koptyuga 3, Novosibirsk 630090, Russia*
[2] *Novosibirsk State University, Street Pirogova 2, Novosibirsk 630090, Russia*

Correspondence should be addressed to Alexandr S. Serdyukov; aleksanderserdyukov@yandex.ru

Received 15 April 2015; Revised 1 July 2015; Accepted 6 July 2015

Academic Editor: Yuming Qin

Estimation of the structure response to seismic motion is an important part of structural analysis related to mitigation of seismic risk caused by earthquakes. Many methods of computing structure response require knowledge of mechanical properties of the ground which could be derived from near-surface seismic studies. In this paper we address computationally efficient implementation of the wave-equation tomography. This method allows inverting first-arrival seismic waveforms for updating seismic velocity model which can be further used for estimating mechanical properties. We present computationally efficient hybrid kinematic-dynamic method for finite-difference (FD) modeling of the first-arrival seismic waveforms. At every time step the FD computations are performed only in a moving narrowband following the first-arrival wavefront. In terms of computations we get two advantages from this approach: computation speedup and memory savings when storing computed first-arrival waveforms (it is not necessary to make calculations or store the complete numerical grid). Proposed approach appears to be specifically useful for constructing the so-called sensitivity kernels widely used for tomographic velocity update from seismic data. We then apply the proposed approach for efficient implementation of the wave-equation tomography of the first-arrival seismic waveforms.

## 1. Introduction

Mechanical properties of the ground are crucial information for safe construction. In particular, seismic analysis is an important stage of structural analysis in areas with high earthquake risk. This includes estimation of a structure response to seismic motion. Specific topics here include performance-based seismic design [1], dynamic stress concentration in tunnels and underground structures during earthquakes, slope stability analysis, and design of power transmission tower-lines. Many methods of computing structure response require knowledge of mechanical properties of the ground which could be derived from near-surface seismic studies. It is known that subsurface geologic irregularities may have serious effect on actual site response to ground motion which can be modeled for complicated subsurface structures [2]. Thus it is important to construct a detailed seismic model from near-surface seismic studies.

In this paper we address computationally efficient implementation of the wave-equation tomography for constructing seismic velocity models. This method requires multiple numerical computation of seismic wavefield propagation (specifically first-arrival waveforms) which is a computationally challenging problem, especially in 3D.

Numerical methods of seismic wavefield modeling (finite-difference (FD), finite-element, etc.) are widely used in seismic data imaging and inversion. They can be used for computing wavefield propagation in complex subsurface structures but usually are computationally expensive. For this reason a lot of work is done for speeding up these methods.

As mentioned by Boore [3] there is no need to compute the wavefield in some region of the computational grid before the first-arrival front passes this region (it is uniformly zero in this case). This approach known as expanding computational domain was reintroduced by Karrenbach [4]. Furthermore, if only first-arrival waveforms are of interest, then numerical computations are required only for the grid points in a narrow region following the first-arrival wavefront. Mentioned idea was introduced by [5] and further developed by Kvasnička and Zahradnřandék [6] and Hansen and Jacobsen [7].

The first-arrival front can be computed by numerically solving the eikonal equation on a regular grid using finite-difference method [8, 9] or the fast marching (FM) method [10]. Note that the cost of computing first-arrival traveltimes is negligible compared to numerically solving the wave equation.

Similar approach to addressing first arrivals is the so-called SWEET algorithm for computing first-arrival traveltimes and amplitudes using the damped wave solution in the Laplace domain as suggested in [11]. This algorithm can be implemented for large velocity models by reducing the number of grid points per wavelength with an out-of-core multifrontal algorithm [12]. The advantage of this algorithm is that it computes most energetic wave arrivals. However, it is different from the previously mentioned methods because the Laplace damping is distorting the first-arrival waveforms. Thus it provides stable estimate of finite-frequency traveltimes and amplitudes to be used in Kirchhoff type imaging rather than waveforms which can be used in wave-equation migration/inversion type algorithms.

Efficient method of computing the first-arrival waveforms can be beneficial in the reverse-time migration for constructing seismic images [13, 14] and the wave-equation tomography (WT) for velocity model building of Luo and Schuster [15] (finite-frequency generalization of the conventional ray-based traveltime tomography). Both methods consider each shot gather independently and imply computation of two wavefields: forward computation of the so-called source field and adjoint (time-reversed) computation of the so-called receiver field. Construction of seismic images or velocity update kernels requires time integration of a product of these two wavefields. Thus there is a need to store the entire history of forward field in the computer memory which is usually unfeasible. Alternatively, one can store the history of the forward field at the boundaries and then recalculate it backward in time simultaneously with the adjoint computation [16]. This can reduce memory requirements for storing wavefields but require extra computations to be made. Computation of the source wavefield only in a narrowband around first arrivals seems to be optimal for mentioned applications as it requires considerably less memory.

In this paper we first describe our approach for computing first-arrival waveforms in a narrow computational band following the moving front of the first arrivals. Then we briefly recall the method of the wave-equation tomography and the reverse-time migration which can take advantage of our modeling approach. Finally we present examples showing the speedup that we get in seismic imaging and velocity model update. In this paper we consider only 2D isotropic models to illustrate the concept.

## 2. Hybrid Forward Modeling Technique

We further develop this idea of computing waveforms in a running window and apply it in tomographic velocity model building which can be done in different ways. The most straightforward approach is ray tomography which requires only traveltime picks of the first arrivals.

The key step is regrading the computational arrays in the time increasing order that naturally comes out from the fast marching method [10]. Such strategy allows setting all computational windows before starting FD calculations. Thus we save computer time needed for the window identification and memory pointer shifts outside the window at each FD time step. Also the resulting windowed wavefield can be stored in RAM at every time step using the proposed strategy.

### *2.1. Hybrid Approach to Forward Modeling.*

Let us list main steps of our hybrid modeling approach. For simplicity we consider the 2D wave equation (but it is straightforward to generalize it to the 3D case and elastic wave equation).

The windowed speedup technique proposed by Vidale [5] and illustrated by Kvasnička and Zahradnřandék [6] includes two steps:

(i) Solving the eikonal equation to get traveltime at each grid node.

(ii) Calculating the solution of the wave equation in the shifted window around the first-arrival front that is defined using the traveltime field from the previous step.

The mathematical background of the windowed calculation technique is obvious: the *region of influence* of the wave-equation solution is determined by the *characteristics*.

During the second wave-equation-solution step at each time level all grid points can be updated or held unchanged. The FD calculations take place only for updated points that form the shifted zone around the first-arrival front. The solution in unchanged points is taken from the previous time level. The conventional way (Kvasnička and Zahradnřandék [6], Vidale [5]) to determine the updated zone is to check the condition:

$$t_f - t_{\text{left}} \le n\Delta t \le t_f + t_{\text{right}}, \qquad (1)$$

where $n$ is the number of the time levels, $\Delta t$ is the FD time spacing, $t_f$ is the first-arrival traveltime, and $t_{\text{left}}$, $t_{\text{right}}$ are the constants. These constants can be defined, using the dominant wavelength.

The update condition (1) should be checked for every grid point at every time level during the wave-equation FD solution that may cost significant computer time. We would like to show how this checking can be avoided and propose more efficient new strategy for the windowed calculations that is based on *traveltime increasing array reordering*.

### *2.2. Numerical Grid Upwind Ordering (from Traveltime Solutions).*

As usual, the numerical solvers provide the *viscosity* solution of eikonal equation, which corresponds to the first-arrival traveltimes. At every node $(i, j)$ one has first-arrival traveltime $T_{i,j}$. The resulting array can be sorted in the increasing order, so the result would be a vector (time increasing list):

$$T(k), \quad T(k+1) \ge T(k). \qquad (2)$$

In principle any eikonal FD solver or other computational techniques can be used to obtain traveltime field $T_{i,j}$ for the windowed calculations. We are going to illustrate the approach using the *multistencils fast marching method*, introduced by [17], which is the more recent and robust modification of classic FM.

Here we give a very brief description of original *fast marching method*. The set of all grid points at each step of FM is divided into three sets: *accepted points*, *narrowband,* and *far points*. The main feature of FM algorithm is the special order of traveltimes computation in the grid nodes similar to Dijkstra's algorithm. If point $(i, j)$ is *accepted*, then the traveltime value in it $T_{i,j}$ becomes frozen and does not change any more. Every point that is not *accepted* but has an *accepted* point in its neighborhood should belong to the *narrowband*; the traveltimes in this set are recomputed. All other points are *far*. At every iteration the point from *narrowband* with the minimum value of traveltime becomes *accepted*. For the efficiency *narrowband* is stored in the *heap* structure.

Note that if the *fast marching* solver is used, the rank of the given grid point in the time increasing list (2) is the number of *acceptances* coming out from the FM.

Let us consider the one-to-one mapping:

$$(i, j) \longleftrightarrow k, \tag{3}$$

where $(i, j)$ is the pair of Cartesian indexes and $k$ is the rank of the $T_{i,j}$ in the traveltime increasing list (2); $i = 1, \ldots, N_z$; $j = 1, \ldots, N_x$; $k = 1, \ldots, N_x \times N_z$.

We use mapping (3) to reorder all the arrays, used for the wave-equation FD modeling. For example, let us consider velocity model matrix $V(i, j)$; it can be stored in the form of a vector $V(k)$. Note that mapping (3) should be remembered, as soon as there is a need for getting back to the Cartesian grid. For instance, one can use two integer vectors: $I(k) = i$, $J(k) = j$. Actually almost the same mapping (3) happens within the *fast marching* method for the *narrowband*, stored in the time-sorted heap structure.

*2.3. Computing and Storing Widowed Wavefield.* The proposed time increasing reordering is natural for the windowed speedup calculations. Each time window is set up by its beginning number $k_{\text{begin}}$ and the end number $k_{\text{end}}$ that are defined using list (2). Thus the wave-equation FD calculations at every time level $n$ are done in the cycle from $k = k_{\text{begin}}(n)$ to $k = k_{\text{end}}(n)$ and there is no need for checking the "update" condition (1).

In principle any time-domain wave-equation solver can be modified for windowed calculations, based on the introduced time increasing reordering. The space FD stencil for the node $(i, j)$ includes some neighborhood points, at least $(i \pm 1, j \pm 1)$ (if one is using the second-order central differences). As soon as all the arrays are reordered, one should implement mapping (3) several times to get the ranks $k$ for all the stencil nodes. The more efficient strategy is to introduce additional arrays for the ranks of neighborhood points. For instance, the rank of the "left" node $(i-1, j)$ for the node $(i, j)$ with the rank $k$ can be remembered in the array $k_{\text{left}}(k)$.

Note that one needs to store some additional indexation-mapping arrays for the window and stencils neighborhoods. As an alternative, the first-arrival wavefront propagation should be done simultaneously with wave-equation solution just ahead of the computational window. The FM approach provides the possibility of such "on the fly" traveltime computations. However one should keep mapping (3) (at least backward).

The proposed windowed techniques, based on the time increasing array reordering, provide the opportunity of storing the "history" of the wavefield. At every time level $n = 1, \ldots, N$ the wavefield can be recorded just one by one at the current time level $n$ from $k = k_{\text{begin}}(n)$ to $k = k_{\text{end}}(n)$ in the one large array, containing the entire "history" of the windowed wavefield. After the computations, one can get the access to the wavefield history by reading the wavefield in time level cycle, using the stored widowed limiting ranks $k_{\text{begin}}(n)$ and $k_{\text{end}}(n)$ and using mapping (3) for getting to the Cartesian grid. Note that one can produce the wavefield reading in a reversed time.

*2.4. Fast Marching Method.* Consider the eikonal equation in isotropic medium:

$$(\nabla \tau)^2 = \frac{1}{c^2(\mathbf{x})}, \tag{4}$$

where $\tau(\mathbf{x})$ is the traveltime in the point $\mathbf{x} = (x, z)$ and $c(\mathbf{x})$ is the seismic wave velocity. We will give here a brief description of original *fast marching method* in the two- dimensional case. Let us consider the rectangular Cartesian grid: $x_i = x_0 + ih$, $z_j = z_0 + h * j$, $c_{ij} = c(x_i, z_j)$, and $\tau_{ij} = \tau(x_i, z_j)$. The original FM method [10] is based on the first-order "upwind" FD approximation of eikonal equation (4):

$$\begin{aligned} &\max\left(\max\left(D_{ij}^{-x}\tau, 0\right), -\min\left(D_{ij}^{+x}\tau, 0\right)\right)^2 \\ &\quad + \max\left(\max\left(D_{ij}^{-z}\tau, 0\right), -\min\left(D_{ij}^{+z}\tau, 0\right)\right)^2 \\ &= c_{ij}^{-2}, \end{aligned} \tag{5}$$

where

$$\begin{aligned} D_{ij}^{\pm x}\tau &= \pm\frac{\tau_{i\pm1,j} - \tau_{ij}}{h}, \\ D_{ij}^{\pm x}\tau &= \pm\frac{\tau_{i,j\pm1} - \tau_{ij}}{h}. \end{aligned} \tag{6}$$

The essence of "upwind" approximation (5) is shown in Figure 1. The grid nodes for FD stencil are choosing principle that is based on traveltime increasing "causality"; that is, the direction of "wind" is the direction of wavefront expansion.

The set of all grid points at each step of FM is divided into three sets: *accepted points*, *narrowband,* and *far points*. The main feature of FM algorithm is the special order of traveltimes computation in the grid nodes similar to Dijkstra's algorithm. If point $(i, j)$ is *accepted,* then the traveltime value in it $T_{i,j}$ becomes frozen and does not change any more. Every point that is not *accepted* but has an *accepted* point

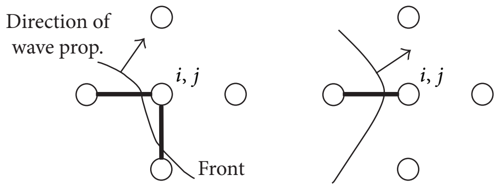


Figure 1: "Upwind" approximation.

in its neighborhood should belong to the *narrowband*; the traveltimes in this set are recomputed. All other points are *far*. At every iteration the point from *narrowband* with the minimum value of traveltime becomes *accepted*. For the efficiency *narrowband* is stored in the *heap* structure.

*2.5. The Virieux Staggered Grid FD Scheme.* In the present paper 2D acoustic wave equation is considered:

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \rho\nabla\cdot\left(\frac{1}{\rho}\nabla p\right) = 0, \tag{7}$$

where $p(\mathbf{x})$ is pressure, $\rho(\mathbf{x})$ is the density, and $c(\mathbf{x})$ is the wave speed.

We use a staggered grid [18] to solve the wave equation (7). In order to implement this scheme the acoustic equation should be rewritten as two first-order equations:

$$\begin{aligned}\frac{\partial p}{\partial t} &= c^2\rho\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z}\right),\\ \frac{\partial v_x}{\partial t} &= \frac{1}{\rho}\frac{\partial p}{\partial x},\\ \frac{\partial v_z}{\partial t} &= \frac{1}{\rho}\frac{\partial p}{\partial z},\end{aligned} \tag{8}$$

where $(v_x, v_z)$ is the particle velocity vector.

The considered FD scheme uses the staggered grid, so that $v_x$ is attached to the points $(i, j)$ $(x_i = x_0 + ih,\ z_j = z_0 + jh)$ of a Cartesian grid, $v_z$ is attached to the "half-integer" nodes $(i + 1/2, j + 1/2)$, and pressure $p$ is attached to the points $(i + 1/2, j)$. Also the velocity and pressure are attached to different time layers. The staggered grid FD stencil for acoustic wave equation is shown in Figure 2. The FD equations are

$$\begin{aligned}\frac{(v_x)^{n+1}_{i,j} - (v_x)^n_{i,j}}{\Delta t} &= \frac{p^{n+1/2}_{i+1/2,j} - p^{n+1/2}_{i-1/2,j}}{\rho_{i,j}h},\\ \frac{(v_z)^{n+1}_{i+1/2,j+1/2} - (v_z)^n_{i+1/2,j+1/2}}{\Delta t} &= \frac{p^{n+1/2}_{i+1/2,j+1} - p^{n+1/2}_{i+1/2,j}}{\rho_{i+1/2,j+1/2}h},\\ \frac{p^{n+1/2}_{i+1/2,j} - p^{n-1/2}_{i+1/2,j}}{\Delta t} &= \left(c^2\rho\right)_{i+1/2,j}\left(\frac{(v_x)^n_{i+1,j} - (v_x)^n_{i,j}}{h}\right.\\ &\quad\left. + \frac{(v_z)^n_{i+1/2,j+1/2} - (v_z)^n_{i+1/2,j-1/2}}{h}\right).\end{aligned} \tag{9}$$

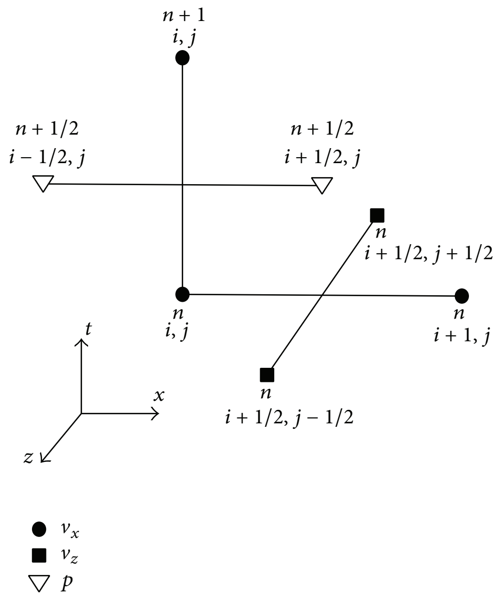


Figure 2: Staggered grid FD stencil.

## 3. Application to Wave-Equation Imaging and Tomography

We show the benefits of the proposed hybrid calculation-and-storing windowed technique for speeding up the reverse-time migration [13, 14] and the wave-equation tomography [15]. For both methods we consider shot gathers independently and for each of them we compute two wavefields: the so-called source field $p_s(\mathbf{x}, t)$ is a result of forward computation and the so-called adjoint field $p_r(\mathbf{x}, t)$ is a result of the reverse-time computation; that is, time-reversed data is plugged in as source signature in the receiver positions. Interaction of these two wavefields is producing seismic image or the velocity model update.

In the reverse-time migration for each shot gather (corresponding to $\mathbf{x}_s$) we get an image as a time integration of a product of these two wavefields:

$$g(\mathbf{x}) = \int p_s(\mathbf{x}, t)\, p_r(\mathbf{x}, t)\, \mathrm{d}t. \tag{10}$$

In the wave-equation tomography for each shot gather (corresponding to $\mathbf{x}_s$) we also construct sensitivity kernels as an interaction of two wavefields:

$$\gamma(\mathbf{x}) = \frac{1}{c^3(\mathbf{x})}\int p_s(\mathbf{x}, t)\, \tilde{p}_r(\mathbf{x}, t)\, \mathrm{d}t. \tag{11}$$

Here the adjoint wavefield $\tilde{p}_r(\mathbf{x}, t)$ is not just time-reversed data but time-reversed pseudo-residual [15]:

$$\delta\tau(\mathbf{x}_r, t) = -\frac{2}{E(\mathbf{x}_r)} p_{\mathrm{obs}}(\mathbf{x}_r, t + \delta\tau)\, \Delta\tau(\mathbf{x}_r), \tag{12}$$

where $p_{\mathrm{obs}}(\mathbf{x}_r, t)$ are observed data at receivers $\mathbf{x}_r$,

$$E(\mathbf{x}_r) = \int p_{\mathrm{obs}}(\mathbf{x}_r, t + \Delta\tau)\, p_s(\mathbf{x}_r, t), \tag{13}$$

and $\Delta\tau$ are traveltime misfits estimated by cross-correlating observed data $p_{\text{obs}}(\mathbf{x}_r, t)$ and synthetic data $p_s(\mathbf{x}_r, t)$ directly from waveforms (maximum of the cross-correlation):

$$\Delta\tau(\mathbf{x}_r) = \arg\min_{\tau} \int p_{\text{obs}}(\mathbf{x}_r, t+\tau)\, p_s(\mathbf{x}_r, t)\, dt. \tag{14}$$

While summing up the results for all shot gathers (for different $\mathbf{x}_s$), we get final migrated image:

$$\widehat{g}(\mathbf{x}) = \sum_s g(\mathbf{x}; \mathbf{x}_s). \tag{15}$$

Similarly, we sum up all shot gather sensitivity kernels (for different $\mathbf{x}_s$) in order to get the velocity model update:

$$\widehat{\gamma}(\mathbf{x}) = \sum_s \gamma(\mathbf{x}; \mathbf{x}_s). \tag{16}$$

Iterative methods can be used to invert for the velocity model $c(\mathbf{x})$ in several steps:

$$c_{k+1}(\mathbf{x}) = c_k(\mathbf{x}) + \alpha_k \widehat{\gamma}_k(\mathbf{x}), \tag{17}$$

where $\widehat{\gamma}(\mathbf{x})_k$ is the sensitivity kernel at $k$th iteration and $\alpha_k$ is updating step length.

The key step of constructing migrated image (10) or velocity model update (11) is to compute the source field $p_s(\mathbf{x}, t)$ and store it for further cross-correlation with the adjoint field $p_r(\mathbf{x}, t)$. For example, precomputed $p_s(\mathbf{x}, t)$ is required for estimating the pseudo-residual used for initiating the adjoint field (cf. (12)).

Thus there is a need to store the entire history of forward source field $p_s(\mathbf{x}, t)$ in the computer memory which is usually unfeasible. Alternatively, one can store the history of the forward field at the boundaries and then recalculate it backward in time simultaneously with the adjoint computation. This can reduce memory requirements for storing wavefields but require one extra waveform computation to be made. Computation of the source wavefield only in a narrowband around the first arrivals seems to be optimal for mentioned applications as it requires considerably less memory. Further we plan to apply the proposed approach only to the source field $p_s(\mathbf{x}, t)$ while computing the adjoint receiver field $p_r(\mathbf{x}, t)$ in a standard mode (note that it does not need to be stored in memory).

## 4. Examples

*4.1. Forward Modeling.* Let us present an example of windowed wavefield computations. The smooth Marmousi velocity model is shown in Figure 3. Receivers are shown by white triangles and source locations are shown by stars. Ricker wavelet with 20 Hz central frequency was used as a source function. Computed pressure data for source 1 are shown in Figure 4: "full" wavefield computed by standard FD scheme in panel (a) and "windowed" computation by the proposed hybrid kinematic-dynamic method in panel (b). Note that the first-arrival waveforms look the same. One can see differences for late arrivals where refracted waves can be seen on "full" data. For source 2 computed "full" and "windowed" data look pretty much the same because there are no refracted waves in this case (for this reason we do not show corresponding gathers).

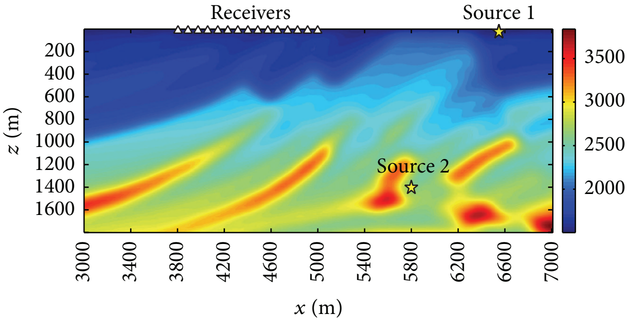


Figure 3: Velocity model (smooth Marmousi) and acquisition geometry.

For this example our hybrid computational approach is 20 times faster than the standard approach (full wavefield computation). Also it takes 17 times less memory to store the source wavefield within narrowband following first arrival (compared to storing full wavefield at each time step). These speedup and memory economy will increase with increasing velocity model size (in terms of dominant wavelength).

*4.2. Wave-Equation Tomography.* Let us consider synthetic example motivated by near-surface applications. For this example we use gradient model with high velocity square anomaly as shown in Figure 5(a). For this true velocity we computed synthetic data using 60 Hz source wavelet (sources and receivers are placed every 5 m at the top of the model). For these data we performed velocity model update using wave-equation tomography (initial model is a model with linear gradient). Source wavefield was computed using our hybrid "windowed" approach. The result of velocity inversion (16) is shown in Figure 5(b). One can see that the velocity anomaly is recovered.

Next example is motivated by applications in seismology. We consider true gradient velocity model with two anomalies as shown in Figure 6(a). Source positions are shown by black stars and receivers were located regularly at the surface. Synthetic data were computed for 10 Hz Ricker wavelet. We performed velocity model update using wave-equation tomography (initial model is a model with linear gradient). Source wavefield was computed using our hybrid "windowed" approach. The result of the velocity inversion is shown in Figure 6(b). One can see that the velocity anomalies can be recovered.

*4.3. Reverse-Time Migration.* Here we consider the reverse-time migration example. Synthetic data was computed for Marmousi model shown in Figure 7(b); 50 shot gathers for 30 Hz Ricker sources were placed every 60 m. Migration velocity model is shown in Figure 7(a).

We use our "windowed" hybrid approach to modeling the source field $p_s(\mathbf{x}, t)$; "windowed" modeling is 33 times faster than "full" waveform modeling (1.5 and 50 seconds

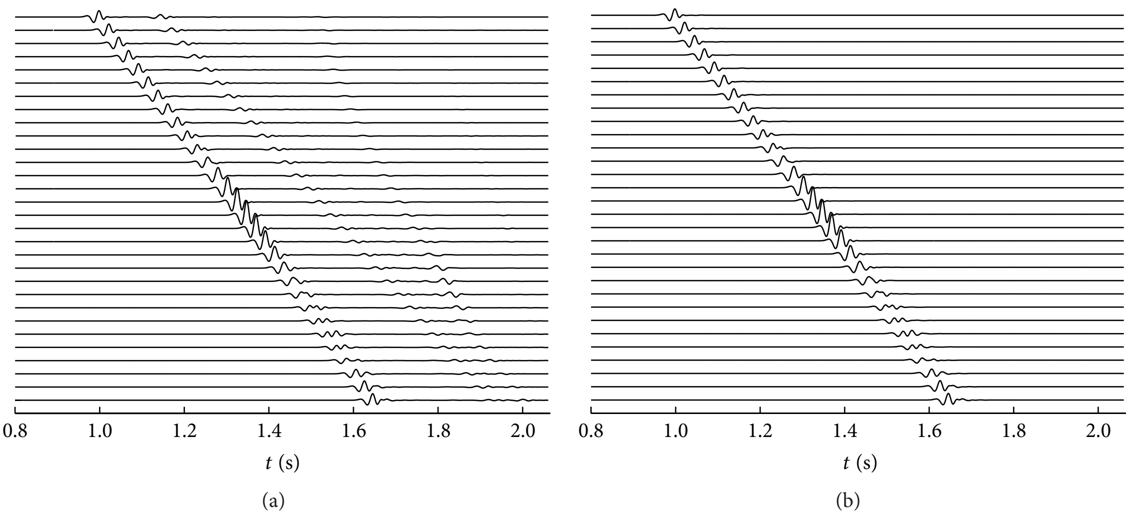


FIGURE 4: Synthetic data for source 1 computed by standard FD scheme (a) and "windowed" approach (b).

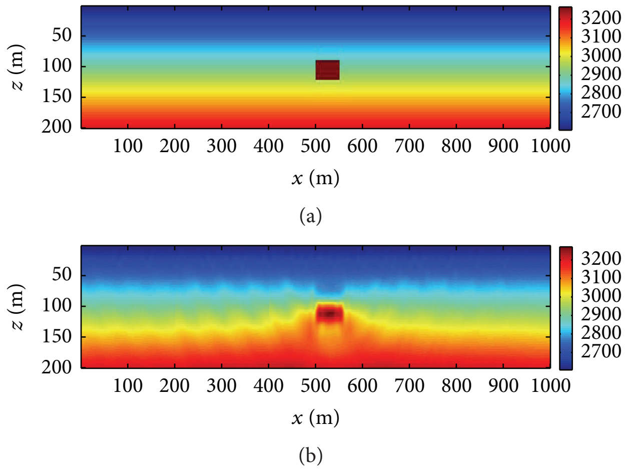


FIGURE 5: Near-surface example: true velocity model (a) and model after tomographic inversion (b).

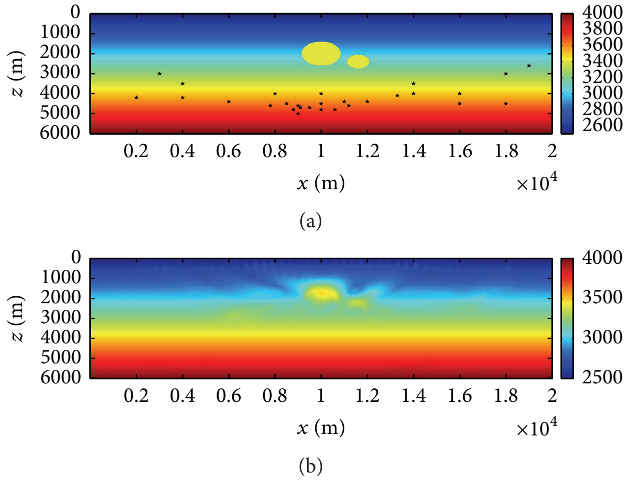


FIGURE 6: Seismologic example: true velocity model (a) and model after tomographic inversion (b).

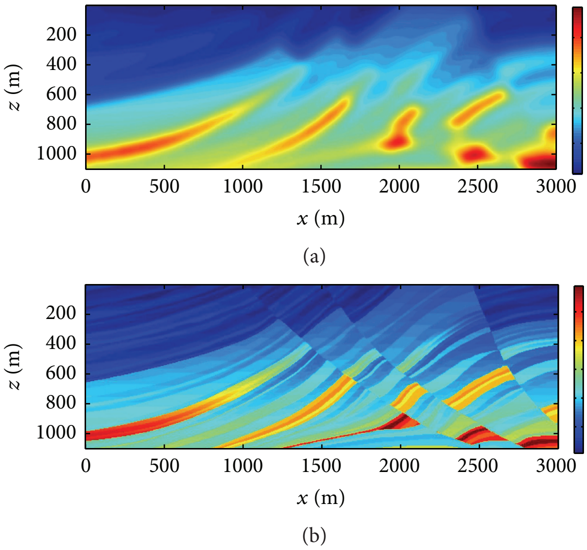
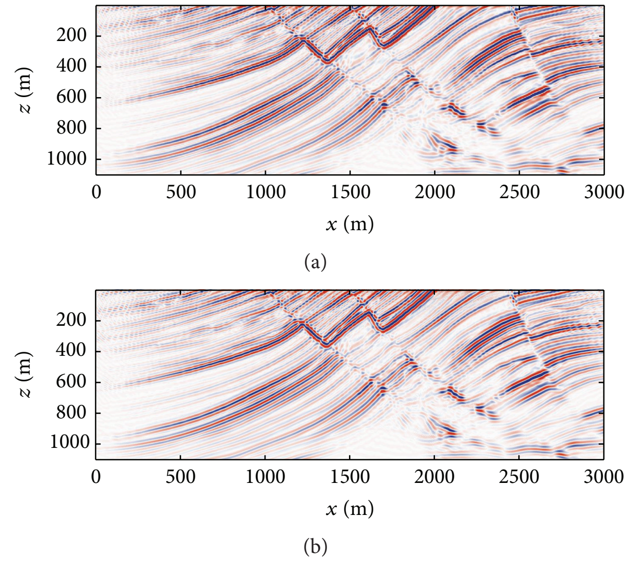


FIGURE 7: Smooth migration model (a) and true velocity model (b).

correspondingly on Intel Core i7-4770 3.4 GHz CPU). It also needs 20 times less memory to store the first-arrival wavefield, that is, 0.8 Gb versus 17 Gb. Reverse-time migrated images for 50 shot gathers (15) are shown in Figure 8: using "full" waveform computation (a) and using our "windowed" computation (b). Both migration results look identical. Note that the receiver wavefield $p_r(\mathbf{x}, t)$ was computed in a "full" waveform mode in both cases.

## 5. Discussion

Examples show that our "windowed" approach allows storing "windowed" source wavefield in memory for any 2D model while fitting "full" source wavefield in memory is not feasible in most cases. Overall speedup of using our "windowed" approach to wave-equation tomography and reverse-time



(a)

(b)

Figure 8: Reverse-time migrated images computing source field by standard full wavefield (a) and our "windowed" approach (b).

migration is close to 3. For standard approach the source field does not fit memory and thus one requires three standard "full" wavefield computations [16] for implementing tomography update or migration. Our "windowed" approach dramatically reduces time required for the source field computation and it can be stored in memory for further use. Thus computational cost of our "windowed" approach is close to one "full" wavefield computation of the receiver field.

We want to mention potential problems in applying our "windowed" modeling approach for contrast models [6]. The eikonal solver provides first arrivals which may correspond to weak head waves while energetic waves useful for imaging may arrive later. Computational window following the first-arrival front may miss these waves. However, the class of velocity models in which our approach works is large enough to make it useful for practical applications.

## 6. Conclusions

We have presented a new hybrid kinematic-dynamic finite-difference (FD) method for computing and storing the first-arrival seismic waveforms. In this "windowed" computational approach at every time step we perform FD computations only in a narrow region following the first-arrival wavefront. This wavefront is precomputed using fast marching eikonal solver. We have proposed a new effective strategy for computations and wavefield storing which is based on time increasing reordering.

We showed several examples of successful use of our "windowed" modeling approach in wave-equation tomography and migration. Proposed approach showed 20–30 times speedup in computing and 17–20 times less memory for storing the source wavefield when compared to standard "full" wavefield computing and storing. It allows overall 3 times speedup for implementing wave-equation tomography or reverse-time migration.

In this paper we consider only 2D isotropic models for illustrating the concept. Extension to 3D models is straightforward. Generalization of the approach to anisotropic models requires efficient anisotropic eikonal solver algorithm availability. It is a topic of our future research.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

Research was partly supported by the Russian Foundation for Basic Research (Grant 14-05-00862). Authors are grateful to the unknown reviewer's comments which have helped to improve the paper.

## References

[1] A. Ghobarah, "Performance-based design in earthquake engineering: state of development," *Engineering Structures*, vol. 23, no. 8, pp. 878–884, 2001.

[2] D. Komatitsch and J.-P. Vilotte, "The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures," *Bulletin of the Seismological Society of America*, vol. 88, no. 2, pp. 368–392, 1998.

[3] D. Boore, "Finite-difference methods for seismic wave propagation in heterogeneous materials," *Methods in Computational Physics*, vol. 2, pp. 1–36, 1972.

[4] M. Karrenbach, "Combining eikonal solutions with full waveform modeling," in *Proceedings of the SEG Annual Meeting*, 2000.

[5] J. Vidale, "Finite-difference calculation of travel times," *Bulletin of the Seismological Society of America*, vol. 78, no. 6, pp. 2062–2076, 1988.

[6] M. Kvasnička and J. Zahradnřandék, "A combined kinematic-dynamic method for fast computations of the first-arriving waveforms," 1996, http://sw3d.mff.cuni.cz/papers.bin/r4mk1.pdf.

[7] T. M. Hansen and B. H. Jacobsen, "Efficient finite difference waveform modeling of selected phases using a moving zone," *Computers & Geosciences*, vol. 28, no. 7, pp. 819–826, 2002.

[8] J. Van Trier and W. W. Symes, "Upwind finite-difference calculation of traveltimes," *Geophysics*, vol. 56, no. 6, pp. 812–821, 1991.

[9] P. Podvin and I. Lecomte, "Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools," *Geophysical Journal International*, vol. 105, no. 1, pp. 271–284, 1991.

[10] J. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.

[11] C. Shin, D.-J. Min, K. J. Marfurt et al., "Traveltime and amplitude calculations using the damped wave solution," *Geophysics*, vol. 67, no. 5, pp. 1637–1647, 2002.

[12] D. Yang, C. Shin, K. J. Marfurt, J. Kim, and S. Ko, “Three-dimensional traveltime and amplitude computation using the supressed wave equation estimation of traveltime (sweet) algorithm,” *Journal of Seismic Exploration*, vol. 12, pp. 75–101, 2003.

[13] E. Baysal, D. D. Kosloff, and J. W. C. Sherwood, “Reverse time migration,” *Geophysics*, vol. 48, no. 11, pp. 1514–1524, 1983.

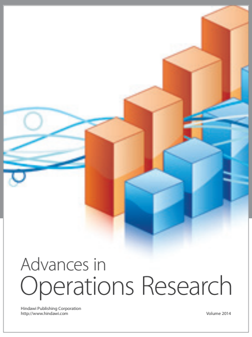
[14] G. A. McMechan, “Migration by extrapolation of time-dependent boundary values,” *Geophysical Prospecting*, vol. 31, no. 3, pp. 413–420, 1983.

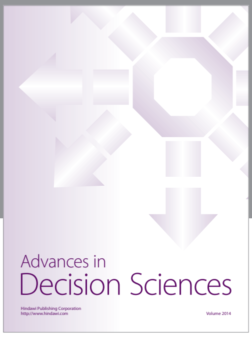
[15] Y. Luo and G. T. Schuster, “Wave-equation traveltime inversion,” *Geophysics*, vol. 56, no. 5, pp. 645–653, 1991.

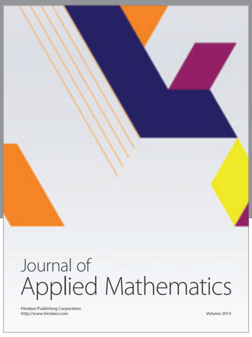
[16] O. Gauthier, J. Virieux, and A. Tarantola, “Two-dimensional nonlinear inversion of seismic waveforms: numerical results,” *Geophysics*, vol. 51, no. 7, pp. 1387–1403, 1986.

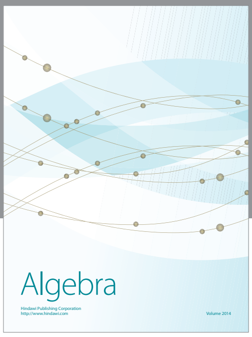
[17] M. S. Hassouna and A. A. Farag, “Multistencils fast marching methods: a highly accurate solution to the Eikonal equation on Cartesian domains,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1563–1574, 2007.

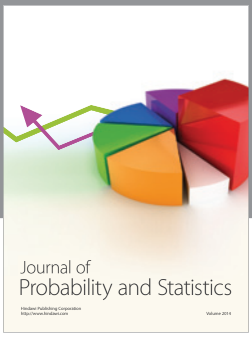
[18] J. Virieux, “P-SV wave propagation in heterogeneous media: velocity-stress finite-difference method,” *Geophysics*, vol. 51, no. 4, pp. 889–901, 1986.


Advances in
Operations Research

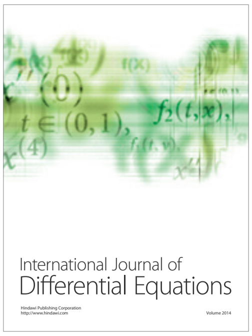

Advances in
Decision Sciences


Journal of
Applied Mathematics

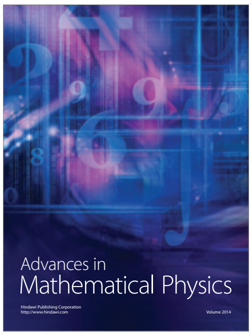

Algebra

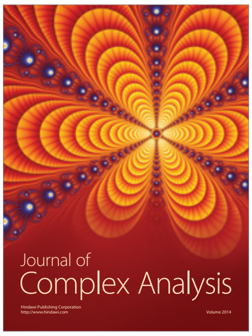

Journal of
Probability and Statistics

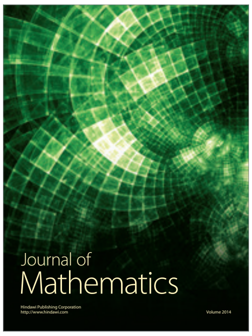
The Scientific
World Journal

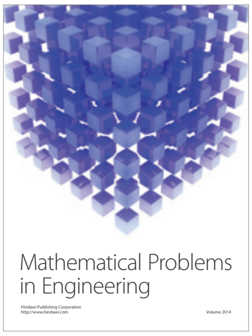

International Journal of
Differential Equations

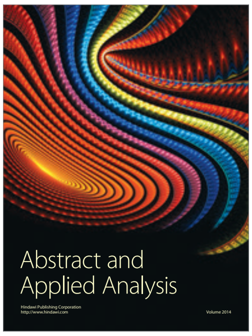
International Journal of
Combinatorics

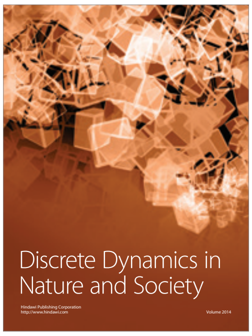

Advances in
Mathematical Physics

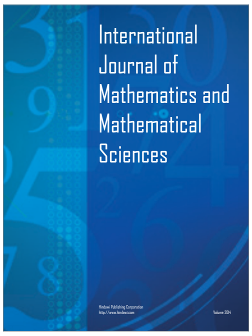

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

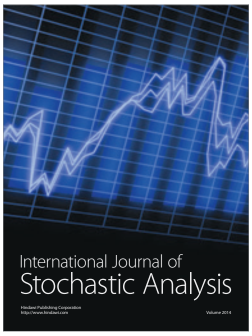

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization