CSIRO PUBLISHING

Exploration Geophysics
https://doi.org/10.1071/EG17128

A high expansion implicit finite-element prestack reverse time migration method

Limin Liu[1,2,3,4] Yun Wang[1] Yong Wang[3] Jing Chen[3] Yanqiu Liu[3]

[1]State Key Laboratory of Ore Deposit Geochemistry, Institute of Geochemistry, Chinese Academy of Sciences, Guiyang 550081, China.
[2]University of Chinese Academy of Sciences, Beijing 100049, China.
[3]Geophysical Research Institute, Jiangsu Oilfield Company, SINOPEC, Nanjing 210046, China.
[4]Corresponding author. Email: liulm.jsyt@sinopec.com

Abstract. Building on the concepts of cohesion degree and local relaxation, we propose an integrated hierarchical equilibrium parallel finite-element reverse time migration (HEP-FE-RTM) algorithm, which is a fine-grained central processing unit (CPU) parallel computation method in two-level host-sub-processors mode. A single master process is responsible for data reading and controlling the progress of the calculation, while each subordinate process deals with a part of the depth domain space. This algorithm is able to achieve single source forward-modelling and inversion calculation using more than 2000 CPUs. On the premise of controlling iteration times for convergence, sub-module/processors only communicate with their adjacent counterparts and the host processor, so the level of data exchange is proportional to cohesion degree. This HEP-FE-RTM algorithm has the distinct advantage that parallel efficiency does not decrease as the number of processors increases. In two-level host-sub-processors mode, more than 2000 processors are used and one billion unknowns are solved. By combining the finite-element implicit dynamic Newmark integral scheme, this approach achieves a prestack reverse time migration (RTM) with high expansion. Making full use of the characteristics of high accuracy and strong boundary adaptability of the finite-element method, through the optimisation of finite-element solving, the HEP-FE-RTM algorithm improved the efficiency of parallel computing and achieved RTM implementation using finite element. Model tests show that this method has a significant effect on both imaging efficiency and accuracy.

Key words: cohesion degree, finite element, local relaxation, multi-level parallel, reverse time migration.

Received 7 December 2016, accepted 26 November 2017, published online 4 January 2018

Introduction

The reverse time migration (RTM) method was proposed by Hemon (1978) and Baysal et al. (1983). Compared to the Kirchhoff migration, this approach is able to simultaneously solve multi-path and dip-angle problems to attain a variety of complex underground structural images because it incorporates a two-way wave equation. The RTM method and the full waveform inversion algorithm are very similar. Mora (1989) demonstrated that the inversion problem can be regarded as the result of a series of migrations and consequently updated this model, while Du and Qin (2009) applied the elastic wave multi-component prestack RTM method to a horizontal isotropic medium. In addition to this work, a variety of reverse time imaging methods have subsequently been developed in order to achieve higher imaging quality and reduce computation time. For example, Hu and McMechan (1986) used a hybrid approach that incorporated both the ray-tracing method to calculate first break traveltimes during source-wavefield forward-modelling and the finite-difference method (FDM), used when receiver wavefield extrapolation is performed, to obtain the excitation traveltime imaging condition of prestack RTM imaging. Whitmore and Lines (1986) later proposed the use of cross-correlation imaging conditions integrating source and receiver wavefields. Finally, as a result of high-performance hardware development and parallel-computing technology, the finite-difference parallel RTM method based on central processing units (CPUs) and graphic processing units (GPUs) has emerged. With this in mind, Liu et al. (2010) proposed a high-order finite-difference seismic prestack RTM algorithm based on a GPU parallel cluster. GPU parallel computation constitutes fine-grained parallelism where multiple GPUs process the same shot data, while CPU parallel computation is coarse-grained parallelism used for parallel computation of shots.

As for boundary condition, the most effective method is the perfectly matched boundary method, which changes the propagating wavefield into a decaying wavefield (Berenger, 1994). A second common technique is to introduce a damping region at the edge of the computation domain (Cerjan et al., 1985). This is often combined with techniques to kill plane waves that are perpendicular to the computational boundary. All of these techniques have proven effective for modelling seismic data, but the equations used for the boundary and internal domains are very different, which is not conducive to parallel implementation and load balancing. Furthermore, the RTM imaging condition requires simultaneous use of source forward-modelling and recorded data reverse time extrapolation wavefields. If the two wavefields are to converge at the same time points, one propagation process must be stored, which requires large storage resources. This requirement is hard to meet in actual operation.

To reduce the storage requirement, Symes (2007) discussed checkpointing methods where the source wavefield is stored at various intervals during forward propagation. When propagating the receiver wavefield these checkpoints are read from disk and re-propagated into a buffer to be correlated with the receiver propagation. There are several undesirable features to this approach. First, the source wavefield must be re-propagated. Second, buffering of the re-propagated source wavefield introduces a large memory requirement. Finally, reading the snapshots from disk does make RTM input/output (IO) bound. To further solve the storage and IO requirements problems, Clapp (2009) proposed an alternative approach. He replaced the conventional damped region with an increasingly random velocity region. Rather than eliminate reflections, he distorted them to minimise coherent correlations with the receiver wavefield. By propagating the source wavefield to max time and then reversing the propagation, the need for checkpointing is eliminated. Although random boundary conditions will affect the seismic wave forward-modelling results, these have little impact on the imaging results, meaning boundary reflection images cannot be realised. In terms of parallel computing, the greatest benefit of this method is that there is no additional computational burden and this process does not affect load balancing.

Currently, wavefield forward-modelling algorithms are divided into the FDM and the finite-element method (FEM). The basic idea of the former is the use of a linear combination of the physical values at nodes to approximate the derivative term of the partial differential equation. To do this, a staggered-grid finite-difference scheme is used, which has higher computational efficiency. By increasing the approximation order, the accuracy is enhanced at the expense of a decrease in calculation efficiency. Although this approach can also be easily extended to the use of GPUs to accelerate computation, it is extremely difficult for this method to deal with irregular regions.

The FEM is an important approach for the calculation mechanics of solids, as it can deal with any complex region and generate high-precision numerical simulation results. However, this approach requires high computational power. This is because the basic idea of FEM is to discretely divide the solution domain into finite elements that are non-overlapping and inter-connected and then choose the interpolation function for each element in order to approximate the true solution of a problem using the linear combination of physical-node value and interpolation function. Each element interpolation function is assumed to constitute the overall function of the solution domain; thus, the solution for the entire domain is composed of the approximate solutions of all elements.

Recent studies have shown that while the implicit FEM can be expected to achieve accurate imaging for irregular topographies, complex structures and complex velocity-distribution conditions (Zhang and Ning, 2002), the required computational memory will be much larger, as the key to this process is the solution of linear equations following FE discretisation. Currently, in terms of solving parallel linear equations, the classic triangular decomposition method, conjugate gradient method with preconditioning and the generalised minimum residual (GMRES) method have all been extended to parallel platforms, and a series of highly efficient solvers have been developed. In terms of parallel efficiency and potential for development, the direct-solution method, represented by the multi-frontal massively parallel sparse approach, exhibits a lower solution scale due to inherent limitations, while the iterative solution method, represented by a conjugate gradient based on Krylov subspace and the GMRES method, can reduce the amount of communication to a certain extent. Nevertheless, some preprocessing measures need to be taken if this approach is applied, including preliminary triangular decomposition using direct solution to improve the robustness of the algorithm. This method, therefore, also has several limitations (He et al., 2010). In summary, although current algorithms and solvers have optimised and improved the parallel efficiency of solving sparse linear equations at different technical levels, most can so far only be applied with a computer cluster comprising up to a few dozen (i.e. maximum 200) processors, which severely limits further expansion of calculation scale and constrains FEM processing of seismic data.

Building on the idea of cohesion degree and local relaxation, we propose here an integrated CPU fine-grained hierarchical equilibrium parallel FE-RTM (HEP-FE-RTM) algorithm, which enables calculation of the same shot data using more than 2000 CPUs. On the premise of controlling the iteration times for convergence, sub-module/processors only communicate with adjacent sub-modules/processors and the host processor, so data exchange level is proportional to the degree of cohesion; thus, this algorithm has the advantage that parallel efficiency does not decrease with an increase in the number of processors. Indeed, in two-level host-sub-processor mode, more than 2000 processors can be used and one billion unknowns solved. When integrated with the highly efficient sparse linear equation solving algorithm, preprocessing of the parallel FEM and implicit finite element (IFE) dynamic Newmark integral method, an IFE prestack RTM with high expansion can be achieved. The high accuracy and strong boundary adaptability of this FEM can then be applied to RTM. Finally, model tests show that this method has a significant effect on the efficiency and accuracy of imaging.

Theoretical background

In this paper, problems of computational efficiency and the storage capacity of the FE-RTM are solved using the parallel IFE method and random boundary conditions. We propose the use of an integrated CPU fine-grained HEP-FE-RTM, which is able to calculate the same shot data using more than 2000 CPUs.

Two-way wave equation FEM

The acoustic wave equation of second order partial differential stress is simple in the case of homogeneous medium, as follows:

$$\frac{1}{\rho}\left(\frac{1}{v^2}\frac{\partial^2}{\partial t^2}-\nabla^2\right)u(\mathbf{x},t)=s(\mathbf{x}_s,t). \tag{1}$$

In this expression, ρ is the density, $v = v\ (x, y, z)$ is the velocity field of the medium (the quadratic term only appears in the equation, scalar, irrespective of direction and refers in particular to wave propagation velocity), $u = u\ (x, y, z, t)$ is the scalar stress distribution, ∇^2 are the Laplace operators and $s\ (\mathbf{x} - \mathbf{x}_s,\ t)$ is the source term. Building on this, if an inhomogeneous medium is to be considered, then Equation 1 is modified, as follows:

$$\frac{1}{v^2}\frac{1}{\rho}\frac{\partial^2}{\partial t^2}u=\frac{\partial}{\partial x}\left(\frac{1}{\rho}\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial u}{\partial z}\right)+s(\mathbf{x},t). \tag{2}$$

In this expression, $\rho = \rho\ (x, y, z)$ refers to the non-uniform density distribution.

The RTM method includes source forward-modelling and recorded data inversion. When both wavefields propagate to the same point of the reflective layer, or the same traveltime, the imaging condition is satisfied. This approach is a 4D problem (x, y, z, t), including time. Applying space discretisation, the pressure distribution of the solution domain can be expressed on the basis of a constructed interpolation function and node pressure value, as follows:

$$u(x,y,z,t) = \sum_{i=1}^{n} N_i(x,y,z)u_i(t). \tag{3}$$

In this expression, as $N_i\ (x, y, z)$ is the interpolation function, Equation 3 can be rewritten in vector form, as follows:

$$u = \mathbf{N}\mathbf{a}^e. \tag{4}$$

In addition, replacing Equation 4 into Equation 1, the Galerkin formulation of equivalent integral form will be established, as follows:

$$\int_{\Omega} \delta u \frac{1}{\rho}\left(\frac{1}{v^2}u_{,tt} - \nabla^2 u\right)\mathrm{d}\Omega - \int_{\Omega} \delta u \times s\mathrm{d}\Omega = 0. \tag{5}$$

Integration by parts of the second terms of Equation 5 generates:

$$\begin{aligned}\int_{\Omega} \delta u \frac{1}{\rho}\nabla^2 u\mathrm{d}\Omega &= \int_{\Omega} \frac{1}{\rho}\nabla(\delta u \nabla u)\mathrm{d}\Omega - \int_{\Omega} \frac{1}{\rho}\nabla(\delta u)\nabla u\mathrm{d}\Omega \\ &= -\int_{\Omega} \frac{1}{\rho}\nabla(\delta u)\nabla u\mathrm{d}\Omega + \int_{\Gamma} \frac{1}{\rho}(\delta u \nabla u)\mathrm{d}\Gamma\end{aligned}. \tag{6}$$

Taking into account the arbitrariness of δu_i replacement of Equation 4 and Equation 6 into Equation 5 generates:

$$\mathbf{M}\ddot{\mathbf{a}}(t) + \mathbf{K}\mathbf{a}(t) = \mathbf{Q}(t)\ . \tag{7}$$

The terms in this expression are as follows:

$$\begin{aligned}\mathbf{M} &= \sum \mathbf{M}_e, \mathbf{M}_e = \int_{\Omega_e} \frac{1}{\rho}\frac{1}{v^2}\mathbf{N}^{\mathrm{T}}\mathbf{N}\mathrm{d}\Omega \\ \mathbf{K} &= \sum \mathbf{K}_e, \mathbf{K}_e = \int_{\Omega_e} \frac{1}{\rho}(\mathbf{N}^{\mathrm{T}})\nabla \cdot \nabla(\mathbf{N})\mathrm{d}\Omega \\ \mathbf{Q}(t) &= \sum \mathbf{Q}_e(t), \mathbf{Q}_e(t) = \int_{\Omega_e} \mathbf{N}^{\mathrm{T}} \times s(t)\mathrm{d}\Omega\end{aligned}. \tag{8}$$

It is noteworthy that in Equation 5, the boundary integral term is pending. Semi-infinite boundary conditions can be formally used to absorb reflection from all borders, as follows:

$$\int_{\Gamma} \frac{1}{\rho}(\delta u \nabla u)\mathrm{d}\Gamma = 0. \tag{9}$$

In this paper, we used flat triangular elements to determine the discrete solution domain. These elements are shape-adaptable, easy to split and have a ‘stiffness matrix’ and ‘quality circle’, as follows:

$$\mathbf{M}_e = \int_{\Omega_e} \frac{1}{\rho}\frac{1}{v^2}\mathbf{N}^{\mathrm{T}}\mathbf{N}\mathrm{d}\Omega = \frac{S}{3}\frac{1}{\rho}\frac{1}{v^2}\begin{bmatrix}\frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2}\end{bmatrix} \tag{10}$$

and

$$\begin{aligned}\mathbf{K}_e &= \int_{\Omega_e} \frac{1}{\rho}(\mathbf{N}^{\mathrm{T}})\nabla \cdot \nabla(\mathbf{N})\mathrm{d}\Omega = \frac{S}{\rho}(\mathbf{N}^{\mathrm{T}})\nabla \cdot \nabla(\mathbf{N}) \\ &= \frac{S}{\rho}\frac{1}{4S^2}\begin{bmatrix} b_1b_1 + c_1c_1 & b_1b_2 + c_1c_2 & b_1b_3 + c_1c_3 \\ b_2b_1 + c_2c_1 & b_2b_2 + c_2c_2 & b_2b_3 + c_2c_3 \\ b_3b_1 + c_3c_1 & b_3b_2 + c_3c_2 & b_3b_3 + c_3c_3 \end{bmatrix}.\end{aligned} \tag{11}$$

For calculations along a time axis, we used the implicit Newmark scheme for time integration, as this method enforces several basic assumptions in the time domain (Fang, 1992), as follows:

$$\begin{aligned}\dot{\mathbf{a}}_{t+\Delta t} &= \dot{\mathbf{a}}_t + [(1-\delta)\ddot{\mathbf{a}}_t + \delta\ddot{\mathbf{a}}_{t+\Delta t}]\Delta t \\ \mathbf{a}_{t+\Delta t} &= \mathbf{a}_t + \dot{\mathbf{a}}_t\Delta t + \left[\left(\frac{1}{2} - \alpha\right)\ddot{\mathbf{a}}_t + \alpha\ddot{\mathbf{a}}_{t+\Delta t}\right]\Delta t^2.\end{aligned} \tag{12}$$

Equation 13 is obtained from Equation 12, as follows:

$$\ddot{\mathbf{a}}_{t+\Delta t} = \frac{1}{\alpha\Delta t^2}(\mathbf{a}_{t+\Delta t} - \mathbf{a}_t) - \frac{1}{\alpha\Delta t}\dot{\mathbf{a}}_t - \left(\frac{1}{2\alpha} - 1\right)\ddot{\mathbf{a}}_t. \tag{13}$$

Resolving the motion Equation 13 generates:

$$\begin{aligned}&\left(\mathbf{K} + \frac{1}{\alpha\Delta t^2}\mathbf{M} + \frac{\delta}{\alpha\Delta t}\mathbf{C}\right)\mathbf{a}_{t+\Delta t} = \\ &\quad \mathbf{Q}_{t+\Delta t} + \mathbf{M}\left(\frac{1}{\alpha\Delta t^2}\mathbf{a}_t + \frac{1}{\alpha\Delta t}\dot{\mathbf{a}}_t + \left(\frac{1}{2\alpha} - 1\right)\ddot{\mathbf{a}}_t\right). \\ &\quad + \mathbf{C}\left(\frac{\delta}{\alpha\Delta t}\mathbf{a}_t + \left(\frac{\delta}{\alpha} - 1\right)\dot{\mathbf{a}}_t + \left(\frac{\delta}{2\alpha} - 1\right)\Delta t\ddot{\mathbf{a}}_t\right)\end{aligned} \tag{14}$$

However, in order to facilitate writing, we rearranged Equation 14 as follows:

$$\begin{array}{llll} c_0 = \dfrac{1}{\alpha\Delta t^2} & c_1 = \dfrac{\delta}{\alpha\Delta t} & c_2 = \dfrac{1}{\alpha\Delta t} & c_3 = \dfrac{1}{2\alpha} - 1 \\ c_4 = \dfrac{\delta}{\alpha} - 1 & c_5 = \dfrac{\Delta t}{2}\left(\dfrac{\delta}{\alpha} - 2\right) & c_6 = \Delta t(1-\delta) & c_7 = \delta\Delta t \end{array}. \tag{15}$$

Thus, the rearrangement generates:

$$\hat{\mathbf{K}}\mathbf{a}_{t+\Delta t} = \hat{\mathbf{Q}}_{t+\Delta t}. \tag{16}$$

The solution of Equation 16 is as follows:

$$\begin{aligned}\ddot{\mathbf{a}}_{t+\Delta t} &= c_0(\mathbf{a}_{t+\Delta t} - \mathbf{a}_t) - c_2\dot{\mathbf{a}}_t - c_3\ddot{\mathbf{a}}_t \\ \dot{\mathbf{a}}_{t+\Delta t} &= \dot{\mathbf{a}}_t + c_6\ddot{\mathbf{a}}_t + c_7\ddot{\mathbf{a}}_{t+\Delta t}\end{aligned}. \tag{17}$$

Finally, to calculate the start, $\ddot{\mathbf{a}}_0$ is unknown but can be obtained, as follows:

$$\ddot{\mathbf{a}}_0 = \mathbf{M}^{-1}(\mathbf{Q}_0 - \mathbf{C}\dot{\mathbf{a}}_0 - \mathbf{K}\mathbf{a}_0). \tag{18}$$

In order to compute Equation 18, calculations usually proceed directly after diagonalisation of $\mathbf{M}$, but in the Newmark method, $\hat{\mathbf{K}}$ has the same matrix structure as both $\mathbf{K}$ and $\mathbf{M}$, so solution can be achieved without diagonalising $\mathbf{M}$. In this way, we cannot only guarantee the accuracy of the mass matrix result, but also use a unified approach to solve Equations 18 and 16 because the mass and stiffness matrices have the same row and column number.

As there is no damping term $\mathbf{C}$ in calculation involving the actual use of random velocity boundary conditions, Equation 16 can be simplified, as follows:

$$\begin{aligned}\hat{\mathbf{K}} &= \mathbf{K} + c_0\mathbf{M} \\ \hat{\mathbf{Q}}_{t+\Delta t} &= \mathbf{Q}_{t+\Delta t} + \mathbf{M}(c_0\mathbf{a}_t + c_2\dot{\mathbf{a}}_t + c_3\ddot{\mathbf{a}}_t). \\ \ddot{\mathbf{a}}_0 &= \mathbf{M}^{-1}(\mathbf{Q}_0 - \mathbf{K}\mathbf{a}_0)\end{aligned} \tag{19}$$

Blocking and the FE parallel meshing principle

The basis of this idea involved the development of a fine-grained CPU parallel-computing method in master-slave mode. In this formulation, a single master process is responsible for data reading and controlling the progress of the calculation, while each subordinate process deals with a part of the depth domain space. These dispersed depth domain spaces are referred to as blocks and are distributed as a checkerboard as shown in Figure 1.

In the case of the HEP-FE-RTM, the numbers of blocks and meshes will jointly determine computational efficiency and computing time. Especially when using a parallel algorithm, there will be an obvious Buckets effect; if the number of element in block I is significantly more than in others, calculation time will be determined entirely by this group. Thus, this algorithm requires matched block-element relationship to balance the load, provided the lengths of the depth domain velocity model in the x- and z-axes directions are L_x and L_z, respectively. In order to suppress regular-boundary reflection, a random velocity field was respectively augmented to the true velocity model along both the positive and negative directions of the x- and z-axes at lengths of dL_x^-, dL_x^+, dL_z^- and dL_z^+. The combined velocity model area is as follows:

$$S = (L_x + dL_x^- + dL_x^+) \times (L_z + dL_z^- + dL_z^+). \tag{20}$$

In order to simplify this relationship, the whole solution domain was first sub divided into uniform rectangular grids with sizes of h_x and h_z, as follows:

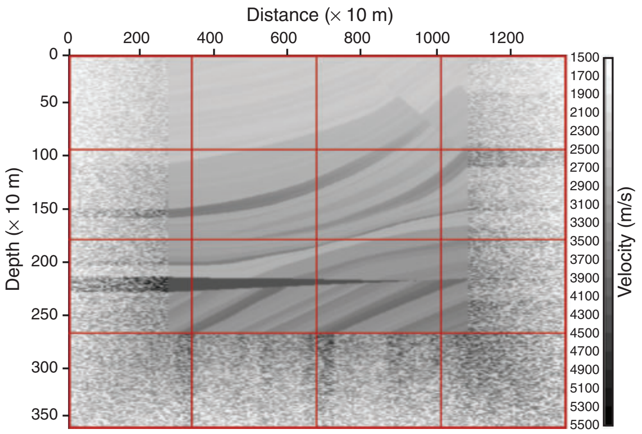

Fig. 1. Diagram of blocking scheme of the local Marmousi velocity model in depth domain after random boundary extension. In actual calculations, this extended model is evenly divided into 4 × 4 blocks (sixteen in total) along the x- and z-axes. The red line represents the block boundary with the external interface.

$$\begin{aligned} &\mathrm{mod}(L_x + dL_x^- + dL_x^+, h_x) = 0 \\ &\mathrm{mod}(L_z + dL_z^- + dL_z^+, h_z) = 0 \\ &n_{Grid}^x = (L_x + dL_x^- + dL_x^+)/h_x \\ &n_{Grid}^z = (L_z + dL_z^- + dL_z^+)/h_z \end{aligned}. \tag{21}$$

When block information was set, we assumed that both the x- and z-axes can be divided into n_x^{Set} and n_z^{Set} blocks, respectively. In addition, in order to ensure balanced parallel-computing load, these blocks must conform to the following requirements:

$$\begin{aligned} &\mathrm{mod}(n_x^{Grid}, n_x^{Set}) = 0 \\ &\mathrm{mod}(n_z^{Grid}, n_z^{Set}) = 0 \end{aligned}. \tag{22}$$

However, in the case of a 2D problem, especially involving an irregular topography area, a triangular element has better adaptability as shown in Figure 2*a*. For consistency, we retained the same triangular element in the regular solution domain (Figure 2*a*), equivalent to addition of a discrete point in the centroid position of a rectangle (Figure 2*b*), forming 1–3–2, 2–3–5, 3–4–5 and 4–3–1 triangular elements (i.e. counter-clockwise in a plane), and constructed a physical distribution that conforms to a coordinate quadratic function and thus achieves higher computing accuracy. From a computational point of view, the position of nodes 1–5 are the same as in Figure 2*b*, but in order to facilitate recognition output, we take into account that rectangular corner points 1, 2, 4, and 5 (solid circles) have corresponding ground nodes in the vertical direction, so only results at rectangular corner point positions correspond to outputs.

As a result, the element number within each block can be expressed, as follows:

$$N_{Set}^{Grid} = 4 \times \frac{n_x^{Grid}}{n_x^{Set}} \times \frac{n_z^{Grid}}{n_z^{Set}}. \tag{23}$$

In reference to Figure 2*b*, the total number of nodes within the block is as follows:

$$N_{Set}^{Node} = \frac{n_x^{Grid}}{n_x^{Set}} \times \frac{n_z^{Grid}}{n_z^{Set}} + \left(\frac{n_x^{Grid}}{n_x^{Set}} - 1\right) \times \left(\frac{n_z^{Grid}}{n_z^{Set}} - 1\right). \tag{24}$$

According to RTM theory and the Newmark time integration scheme, as far as scalar wave is concerned, each node only has one degree of freedom, so the time required for computing is proportional to the number of block nodes, N_{Set}^{Node}.

Thus, the general form of linear equations in a FE simulation can be written as follows:

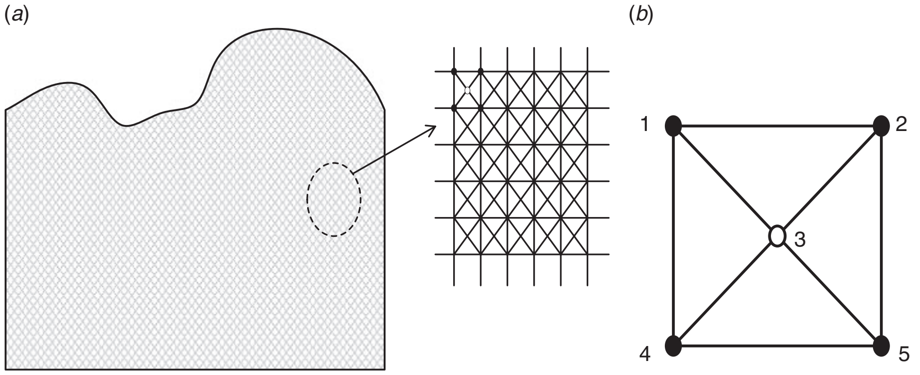

Fig. 2. Comparison diagram of (*a*) mesh and (*b*) triangular elements. For consistency, we retained the same triangular element in the regular solution domain (*a*), equivalent to addition of a discrete point in the centroid position of a rectangle (*b*), forming 1–3–2, 2–3–5, 3–4–5 and 4–3–1 triangular elements (i.e. counter-clockwise in a plane).

$$\sum_{j=1}^{N} \mathbf{k}_{ij} \cdot \mathbf{u}_j = \mathbf{p}_i, i = 1, 2, .., N. \tag{25}$$

In this expression, N refers to the number of nodes. Assuming that the number of degrees of freedom of each node is s, the total number of unknowns is sN. $\mathbf{k}_{ij}$ is the overall stiffness matrix, and $\mathbf{p}_i$ and $\mathbf{u}_j$ refer to the load and unknown displacement vectors, respectively. Equation 25 is established using the principle of minimum potential energy, and the expression for the total potential energy is as follows:

$$\Pi = \frac{1}{2}\sum_{i,j=1}^{N} \mathbf{u}_i \cdot \mathbf{k}_{ij} \cdot \mathbf{u}_j - \sum_{i=1}^{N} \mathbf{u}_i \cdot \mathbf{p}_i. \tag{26}$$

On the basis of the principle of minimum potential energy, we designed an iterative format algorithm, which constantly decreases the value of the function in Equation 26 and definitively solves Equation 25.

For parallel computing, we first divided N nodes into M blocks (or sets) according to their initial location (corresponding to M processors); thus, the number of degrees of freedom within each block is about sN/M. Segmentation was then used to set a dividing plane within each element, rather than on the nodes as shown in Figure 3*b*. This is a different concept from domain decomposition. Thus, an element crossed by a dividing plane is defined as the interface element is shared with adjacent blocks in order to ensure displacement continuity. Another important difference between this approach and the domain decomposition method is that it is unnecessary to introduce either a penalty function or Lagrange multipliers, but in the process of iteratively solving equations, each block needs to communicate with adjacent ones to obtain the solutions of external nodes on each interface element.

Inside each block, a linear combination of displacement incremental mode was used to represent node displacement. Thus, the node displacement value of block I (Set$_I$) can be written as follows:

$$\mathbf{u}_i = \mathbf{u}_i^{\mathrm{old}} + \sum_{k=1}^{q} a_k^{(I)} \hat{\mathbf{u}}_{ki}^{(I)}, \quad i \in \mathrm{Set}_I. \tag{27}$$

In this expression, $\mathbf{u}_i{}^{\mathrm{old}}$ is the approximate solution of the last iteration, q is the number of displacements incremental mode, while $\hat{\mathbf{u}}_{ki}^{(I)}$ and $a_k^{(I)}$ refer to the kth displacement incremental mode and coefficient of the Ith block, respectively. We condensed sN/M numbers that had inherent degrees of freedom in the block into a total of q higher-order degrees of freedom. In general, q is about ten orders of magnitude. The above blocking method, distinct to the substructure method, makes the degrees of freedom of block boundary points into an upper level calculation. Thus, replacing Equation 26 into Equation 27, we obtain potential energy in the form of $a_k^{(I)}$. Thus, according to the principle of minimum potential energy (i.e. functional extreme conditions), we derive:

$$\begin{aligned} \frac{\partial \Pi}{\partial a_m^{(I)}} &= \sum_{J=1}^{M} \sum_{\substack{i \in Set_I \\ j \in Set_J}} \hat{\mathbf{u}}_{mi}^{(I)} \cdot \mathbf{k}_{ij} \cdot \left(\mathbf{u}_j^{\mathrm{old}} + \sum_{l=1}^{q} a_l^{(J)} \hat{\mathbf{u}}_{lj}^{(J)} \right) \\ &\quad - \sum_{i \in Set_I} \mathbf{p}_i \cdot \hat{\mathbf{u}}_{mi}^{(I)} \\ &= \sum_{J=1}^{M} \sum_{l=1}^{q} K_{a_m^{(I)} a_l^{(J)}} a_l^{(J)} - P_{a_m^{(I)}} = 0. \end{aligned} \tag{28}$$

From this expression, the following addition derivations are possible:

$$\begin{aligned} K_{a_m^{(I)} a_l^{(J)}} &= \sum_{\substack{i \in Set_I \\ j \in Set_J}} \hat{\mathbf{u}}_{mi}^{(I)} \cdot \mathbf{k}_{ij} \cdot \hat{\mathbf{u}}_{lj}^{(J)} \\ P_{a_m^{(I)}} &= \sum_{i \in Set_I} \hat{\mathbf{u}}_{mi}^{(I)} \cdot (\mathbf{p}_i - \sum_{J=1}^{M} \sum_{j \in Set_J} \mathbf{k}_{ij} \cdot \mathbf{u}_j^{\mathrm{old}}) \end{aligned} . \tag{29}$$

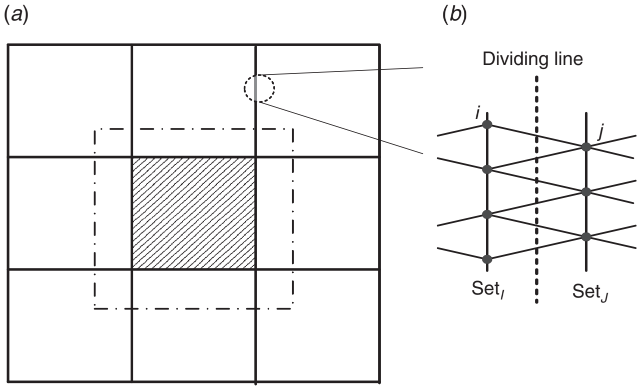

Fig. 3. (*a*) 2D finite-element problem segmentation method diagram. (*b*) Adjacent block share element. The dividing plane falls inside the element.

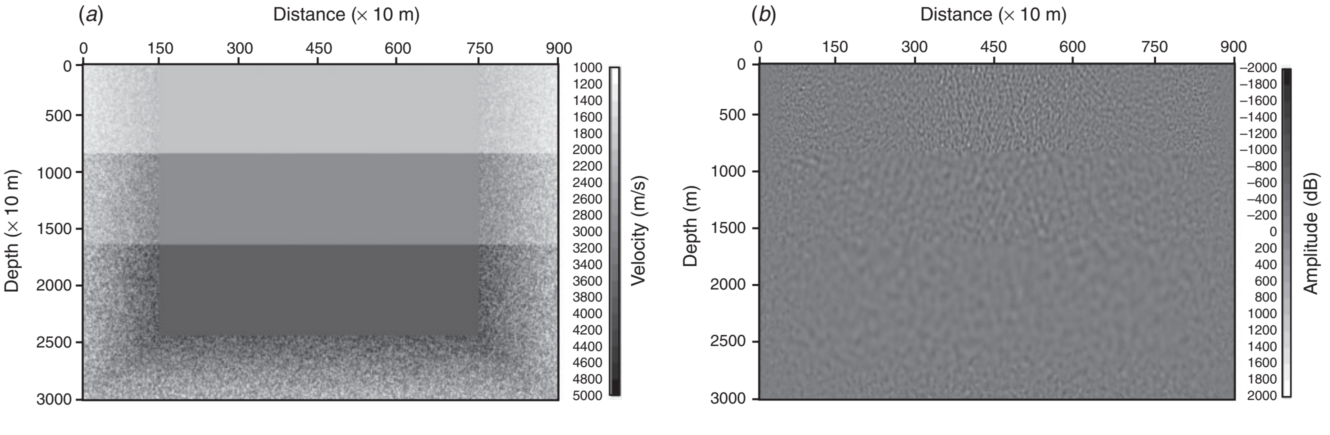

Fig. 4. (*a*) Velocity model in depth domain after the random boundary extension. The scale of the original velocity model is 800 m along the *x*-axis horizontally and 480 m along the *z*-axis in the depth direction. The extended random boundary layer scale is 200 m thick on both sides horizontally and 120 m thick in the depth direction. (*b*) Instant wavefield at the end of forward-modelling time (2.0 s).

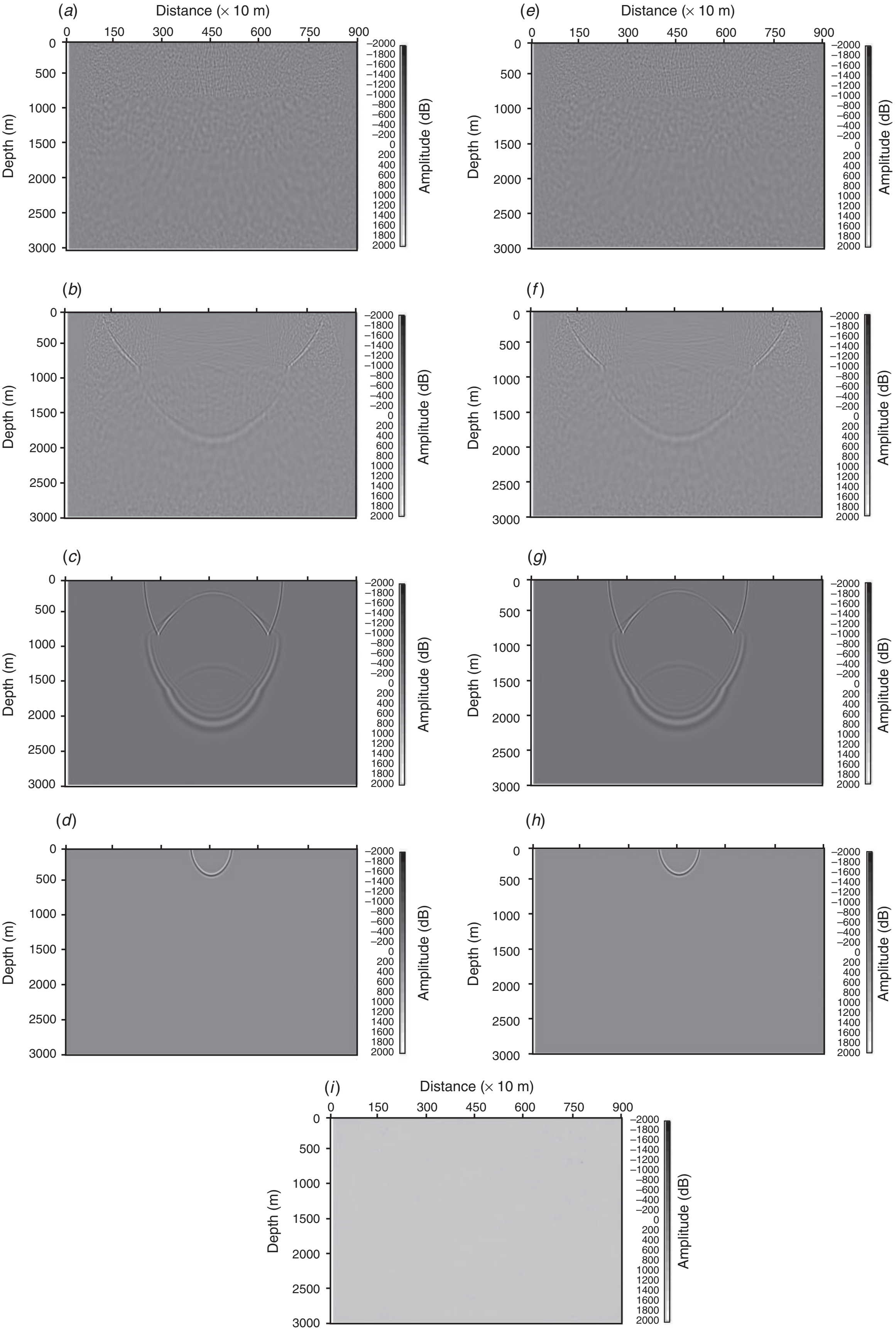

Fig. 5. Wavefield snapshots contrast between source forward-modelling and inversion wavefields with interval 0.6 s from 2.0 to 0.2 s. (*a–d*) Forward-modelling wavefields; (*e–h*) inversion wavefields; (*i*) difference between (*c*) and (*g*).

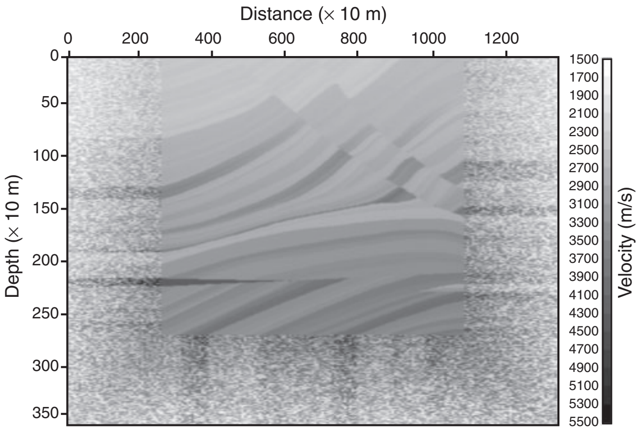

Fig. 6. Local Marmousi velocity model with extended random velocity boundary. It contains 120 traces, each of which has 300 sampling points across 25 m horizontal and 10 m depth directions. The thicknesses of the random boundary layer are 1000 and 1000 m in the horizontal direction and 0 and 1000 m in the depth direction. The final size of the extended model is 5000 × 4000 m.

These expressions comprise the stiffness matrix and imbalance force vector that corresponds with the highest order degree of freedom $a_k^{(I)}$, respectively. The overall balance equation that expresses the cohesion degree of freedom is therefore:

$$\sum_{J=1}^{M}\sum_{l=1}^{q} K_{a_m^{(I)} a_l^{(J)}}\, a_l^{(J)} = P_{a_m^{(I)}};\; m=1,2,..,q;\; I=1,2,..,M. \quad (30)$$

This expression is an equation for overall balance control. For ease of operation, we applied a specialised (master) processor to solve Equation 30, while Equations 27–30 show that the HEP-FE-RTM algorithm does not require assembly of the overall stiffness matrix. However, taking it into account the fact that we need fewer displacement increment modes to express inherent degrees of freedom, equivalent to the introduction of additional restrictions, , displacement increment modes in the iteration process must be self-adaptive within the block and adjusted via a relaxation calculation. In FE analysis, because each node is connected with a very small number of surrounding nodes, and the obtained stiffness matrix is usually banded and sparse, each

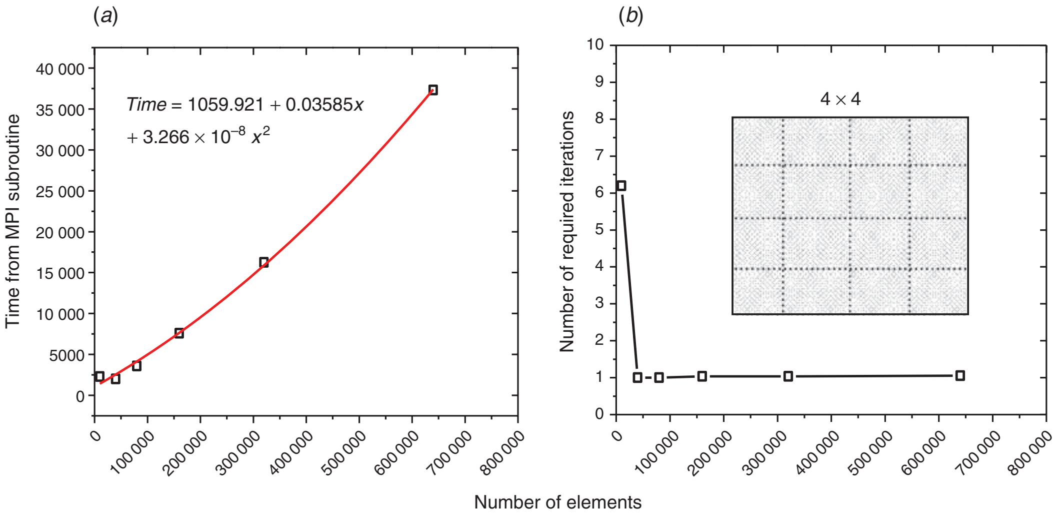

Fig. 7. (*a*) Variation graph of single-shot imaging computation time consumption with the increase of element numbers in the block. MPI = message passing interface. (*b*) Variation graph of the required average iteration numbers with the increase of element numbers when the HEP-FE-RTM algorithm achieve convergence.

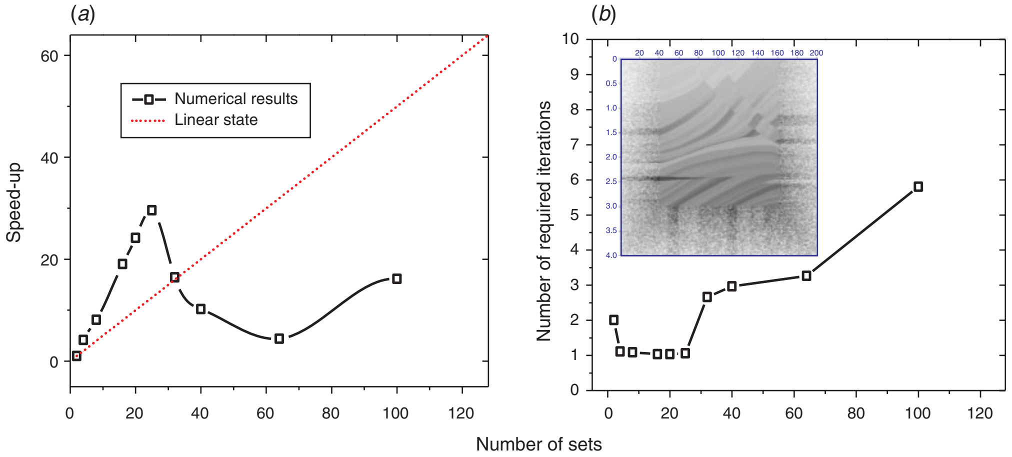

Fig. 8. (*a*) Speed-up ratio of the algorithm aiming at Marmousi problem. (*b*) Variation graph of the required average iteration times with the increase of block number when the HEP-FE-RTM algorithm achieve convergence.

block in Figure 3 needs to only exchange interface information with its neighbours (Figure 3*b*) and most communication is local. This avoids large-scale information exchange overall, so the burden of communication is low. Application of this method also enables the establishment of hierarchical equilibrium solving format through multi-level cohesion degrees of freedom and achieves simulation of larger-scale problems.

Method performance analysis

HEP-FE-RTM algorithm convergence analysis

The HEP-FE-RTM algorithm applied in this study for solving linear equations is as follows:

$$\left(\mathbf{K}+\frac{1}{\alpha\Delta t^2}\mathbf{M}\right)\mathbf{a}_{t+\Delta t}=\hat{\mathbf{Q}}_{t+\Delta t}. \tag{31}$$

In this expression, the stiffness matrix, **K**, is usually a non-positive definite because it has a zero eigenvalue, while the mass matrix, **M**, is a positive definite. Thus, the coefficient matrix on the left of the upper formula is a positive definite matrix. Because of this, functional II obtained via cohesion degree of freedom after each blocking must contain an extreme value, equivalent to Equation 32, which enables convergence of the solution through an iterative calculation:

$$\begin{aligned}\Pi=\sum_{I=1}^{M}\sum_{J=1}^{M}\sum_{\substack{i\in Set_I\\ j\in Set_J}}&\left(u_i^{\text{old}}+\sum_{l=1}^{q}a_l^{(I)}\hat{u}_{li}^{(I)}\right)\cdot k_{ij}\cdot\\ &\left(u_j^{\text{old}}+\sum_{l=1}^{q}a_l^{(J)}\hat{u}_{lj}^{(J)}\right)\\ -\sum_{I=1}^{M}\sum_{i\in Set}&p_i\cdot\left(u_i^{\text{old}}+\sum_{l=1}^{q}a_l^{(I)}\hat{u}_{li}^{(I)}\right)\end{aligned}\quad. \tag{32}$$

In terms of linear equations possessing definite coefficient matrices, convergence of the HEP-FE-RTM algorithm has been demonstrated from a numerical point of view. In this study, to ensure precision of the time integration scheme, the HEP-FE-RTM algorithm uses a higher convergence accuracy by default and thus meets the requirements of:

$$\begin{cases}\dfrac{\left|\left(\mathbf{K}+\dfrac{1}{\alpha\Delta t^2}\mathbf{M}\right)\mathbf{a}_{t+\Delta t}^{i}-\hat{\mathbf{Q}}_{t+\Delta t}\right|_1}{|\hat{\mathbf{Q}}_{t+\Delta t}|_1}\le 10^{-9}\\[2ex] \dfrac{|\Delta\mathbf{a}_{t+\Delta t}^{i}|_1}{|\Delta\mathbf{a}_{t+\Delta t}^{1}|_1}\le 10^{-15}\end{cases}. \tag{33}$$

In this expression, $\Delta\mathbf{a}_{t+\Delta}{}^{i}$ represents the incremental amplitude vector of the *i*th iteration:

$$|\Delta\mathbf{a}_{t+\Delta t}^{i}|_1=-\sum_{l=1}^{N}|\Delta a_{t+\Delta t}^{il}|. \tag{34}$$

This expression is the first-order norm of incremental amplitude vector. Indeed, on the basis of model tests, we were able to determine that between two and four iterations of the solving equations are sufficient to meet the convergence criteria of Equation 34.

Accuracy analysis of the implicit dynamic Newmark time integration scheme

The accuracy of this time integration scheme directly affects the effectiveness of computed wavefield and further impacts imaging results. To address this issue, we applied the implicit dynamic Newmark time integration scheme. First, we applied a simple model with three velocity layers to analyse the accuracy of the integration scheme above. The depth domain velocity model profile extended by random boundary layers is shown in Figure 4*a*. Our meshing generation scheme was 2 × 2 m and a total of 864 000 triangular elements were used. Thus, a virtual source was applied at the centre of the surface and a forward-modelling wavefield profile of 2.0 s was recorded. We also reversed the forward-modelling wavefield along the time axis. This approach enabled a direct impression of the effectiveness of our time integration scheme by comparing the forward-modelling and inversion results at the same point along the time axis. The instant wavefield profile at the end of the forward-modelling time (2.0 s) is shown in Figure 4*b*; due to the existence of a random velocity layer, the wavefield (2.0 s) is completely transformed to diffuse reflection. In principle, if the time integration scheme meets the accuracy requirement, the simultaneous instant wavefield profile of forward-modelling and inversion will have the same amplitude distribution.

Different forward-modelling and inversing wavefield times are shown in Figure 5. On the basis of these comparisons, it is clear that these two groups of wavefields conform very well at different times, which shows that the implicit dynamic Newmark time integration scheme meets the requirements of this study and is of reliable accuracy. The difference between forward-modelling and inversing wavefields (0.8 s) as shown in Figure 5*i*, further exhibits the reliability of the accuracy.

HEP-FE-RTM algorithm scalability and parallel efficiency analysis

We selected a portion of the Marmousi velocity model as our test object for this analysis. The depth domain velocity model treated with a random velocity boundary approach is shown in Figure 6. For the test we used a cluster of ten nodes, each containing eight cores (i.e. 80 cores in total). This test was applied for a single-shot imaging calculation, set in the midpoint of the horizontal direction in Figure 6.

Initially, we tested the scalability of the HEP-FE-RTM algorithm. To do this, the blocking scheme in the depth domain velocity model was fixed into 4 × 4 blocks in horizontal and depth directions and 17 processes were used. Grid density was increased by adjusting the grid size. Figure 7*a* shows the time consumption variation of single-shot imaging computation with an increase in the number of elements in the block. Although the fitting result is a quadratic curve, the magnitude of the quadratic term coefficient is much lower than of the linear term, which indicates that the HEP-FE-RTM algorithm

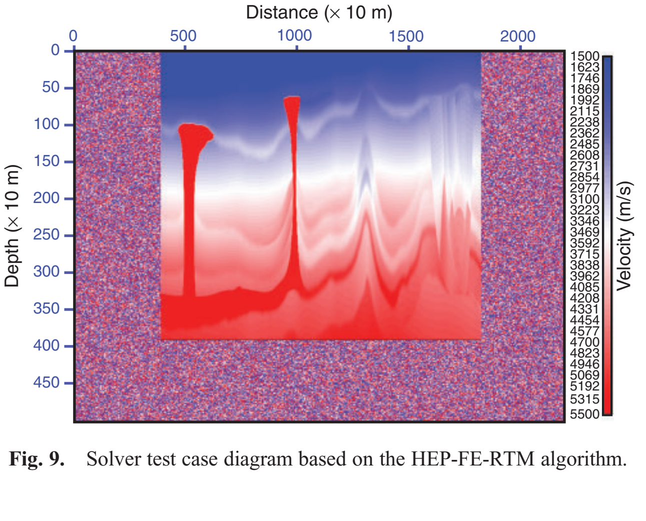

Fig. 9. Solver test case diagram based on the HEP-FE-RTM algorithm.

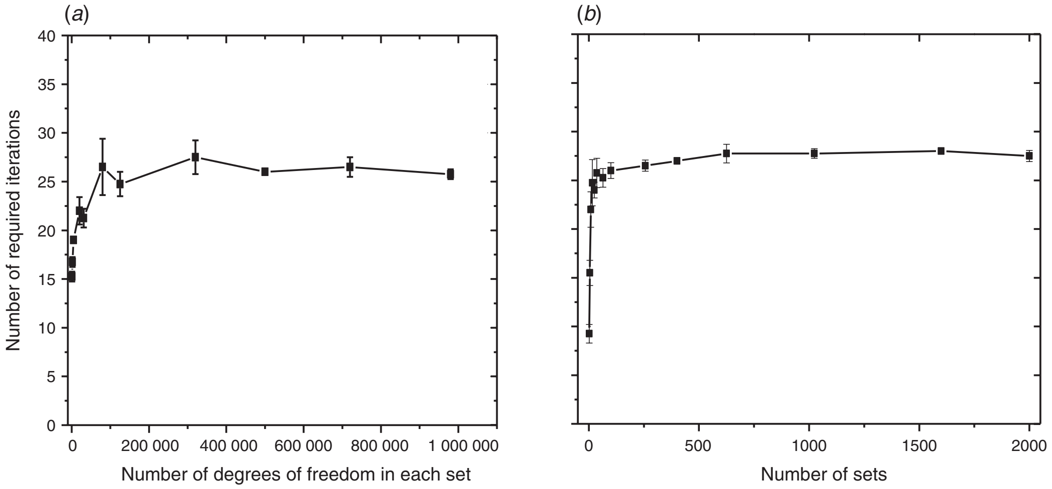

Fig. 10. (*a*) The relationship between required iteration times and the unknown quantity in each block. (*b*) The relationship between required iteration times and the block number.

exhibits very good scalability. Figure 7*b* shows that the required average number of iterations changes as the number of elements increase because the HEP-FE-RTM algorithm achieves convergence in the imaging procedure. The results of this analysis (Figure 7*b*) show that when the number of elements is large enough (i.e. greater than 40 000), iteration times are almost the same; this also confirms that the parallel efficiency and scalability of our algorithm is very good.

Second, we fixed the number of triangular elements in the depth domain velocity model to 2 560 000, and then increased the number of blocks (i.e. processes). The biggest number of blocks we applied was 100 (corresponding to 101 processes), and the speed-up situation of the HEP-FE-RTM algorithm is shown in Figure 8*a*. The results of this analysis show that, from a parallel efficiency point of view, the best blocking scheme is 5 × 5, which is the turning point. Figure 8*b* shows the results of variability in the required average iteration number to achieve convergence with the HEP-FE-RTM algorithm as block number increases. It is well known that, after fine blocking, a decrease in parallel efficiency is mainly due to increases in iteration. However, the blocking scheme has a turning point as well. Therefore, from the perspective of parallel efficiency, we need to test the best blocking scheme with regard to different scale data.

In order to test the scalability and efficiency of the HEP-FE-RTM algorithm in detail, we used the same triangular element to test the BP model (Figure 9). This model is much more complex and is treated similarly with a random velocity boundary. We used the HEP-FE-RTM algorithm and the IFE time integral scheme to calculate the BP model; this test uses up to 2001 processors to compute the wavefield propagation problem of one billion unknowns.

We tested two cases to investigate scalability. First, we divided the solution domain into 64 blocks with an 8 × 8 form and increased the number of unknowns in each block from 400 to 980 000 (Figure 10*a*). The results of this approach show that although the number of unknowns in each block and the whole problem changes a great deal, the number of required iterations to achieve algorithm convergence no longer increases if the problem is large enough (i.e. greater than 100 000 degrees of freedom). Second, we enforced 500 000 degrees of freedom within each block and increased the block number from 2 to 2000 (i.e. up to one billion unknowns). The results from this approach (Figure 10*b*) again show that the required iteration times for algorithm convergence do not change if the problem is large enough.

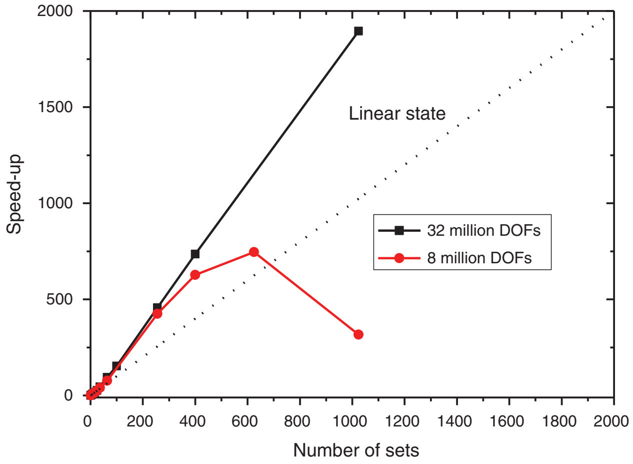

Fig. 11. Speed-up test results on HEP-FE-RTM algorithm. DOFs = degrees of freedom.

Last, we applied test cases comprising eight and 32 million unknowns to test the parallel speed-up of the HEP-FE-RTM algorithm (Figure 11). In the first of these cases, speed-up begins to decrease after 625 processors, while in contrast, the larger the number of unknowns the later the speed-up decrease. In other words, these results corroborate the good parallel efficiency of the HEP-FE-RTM algorithm.

These results show that by optimising both structure and algorithm, the HEP-FE-RTM approach enables computation of more than 2000 processors and one billion unknowns.

Numerical examples

A synthetic data test (Sigsbee2)

In order to test the performance of our method, we investigated synthetic data from the standard Sigsbee2 model. The global Sigsbee2 velocity model is 9144 m in depth and 24 384 m in length (Figure 12*a*) and contains a sedimentary sequence broken up by several normal and thrust faults. Additionally, there is a complex salt structure found within the model that

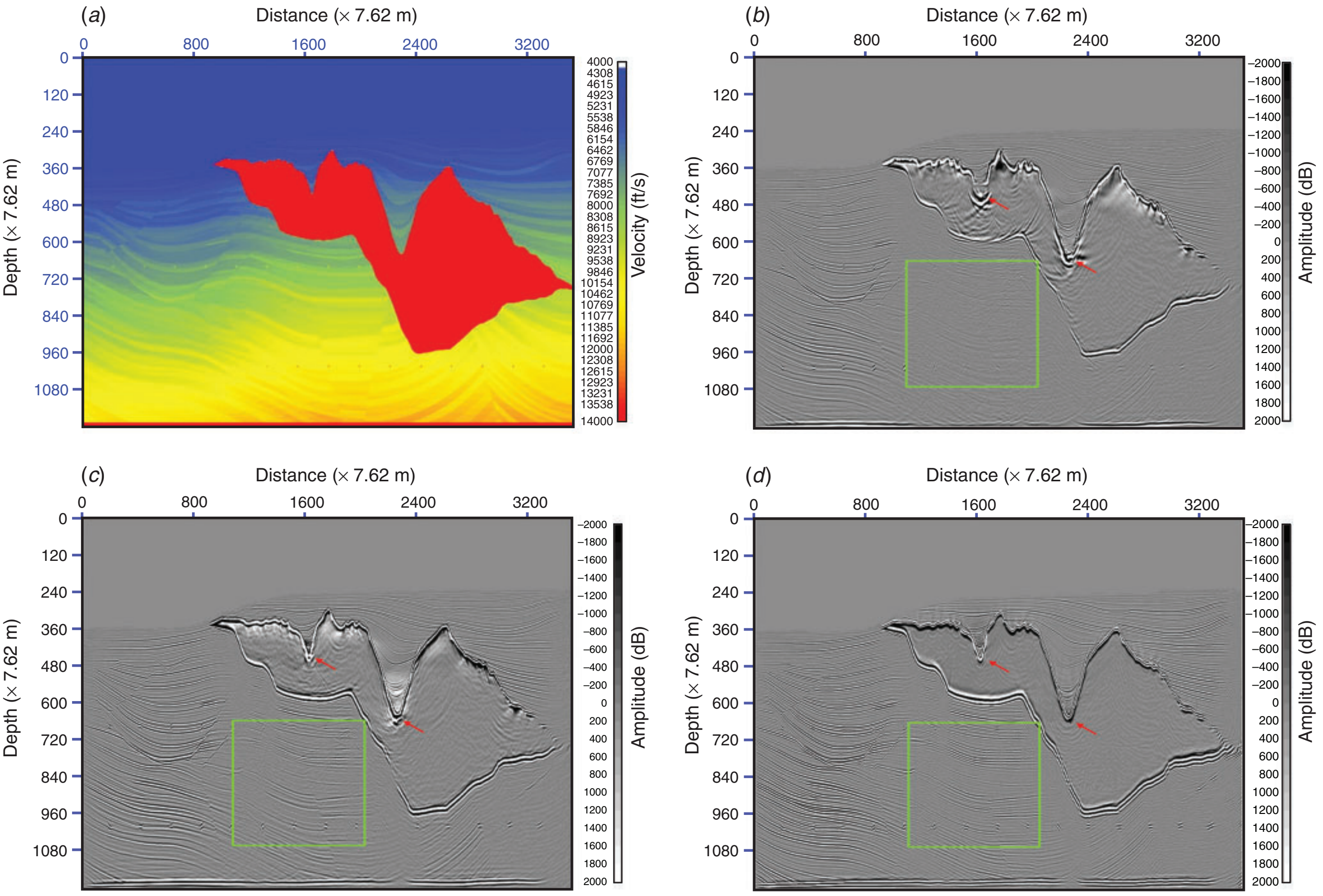

Fig. 12. (*a*) Global Sigsbee2 velocity model. (*b*) WEM. (*c*) FD-RTM. (*d*) HEP-FE-RTM.

Table 1. Comparison of the computational cost for WEM, FD-RTM and HEP-FE-RTM for Sigsbee2 model migration imaging using the 5 × 5 blocking scheme.

	Elapsed time	
Method	Single shot (min)	Total 500 shots (h)
WEM	11.25	93.75
FD-RTM	18.95	157.92
HEP-FE-RTM	19.05	158.75

results in illumination problems when processing and migrating data. In contrast, migration imaging results using the one-way wave equation migration (WEM) are shown in Figure 12*b*. However, as a result of the disadvantage of imaging dip-angle limits using the one-way wave equation, the images of the areas where lateral velocity changes greatly (red arrow) are seriously distorted, while the subsalt complicated structural areas (bright green box) are also inaccurately imaged.

Figure 12*c* shows migration imaging results using the finite-difference RTM method (FD-RTM); because this method uses a two-way wave equation without dip limitation, the image of the subsalt complicated structural areas (bright green box) is improved greatly. However, because finite-difference mesh forms are usually rectangular or hexahedron, it has proved difficult to cope with these dramatic changes in structure, and images of the areas where lateral velocity change greatly (red arrow) are still not ideal. Figure 12*d* shows migration imaging results obtained using the HEP-FE-RTM method proposed in this paper; as this approach incorporates a variety of elements and can deal with any complex region, images of the areas where lateral velocity change greatly (red arrow) and the subsalt fault blocks (bright green box) are all very clear. Comparing the results presented in Figure 12*b–d*, it is clear that the method presented in this paper exhibits a high level of computational accuracy, as it captures almost all stratigraphic characteristics.

The analysis domain size of each shot in the Sigsbee2 imaging test is 9144 m in depth and 24 384 m in length, while the computational domain after random velocity boundary extension is 10 668 m in depth and 27 432 m in length. At the same time, the rectangular grid size is 30.48 × 15.24 m, the blocking scheme applied was 5 × 5 and there are 630 000 triangular elements within each block. We used 26 processors for parallel computations within a cluster system, and the comparison of the computational elapsed time for WEM, FD-RTM and HEP-FE-RTM methods is shown as in Table 1. Compared to the 18.95 min per shot required by FD-RTM, the HEP-FE-RTM approach took 19.05 min to complete the imaging computation of single shot. Therefore, although the HEP-FE-RTM approach took less than an hour longer than the FD-RTM method, imaging accuracy is improved significantly. In addition, the average required iteration time for each time step was 1.0865 before the HEP-FE-RTM algorithm equations achieved convergence.

Synthetic data test from the Subei Basin

We also tested our method using 2D data from the Subei Basin, located within the Jiangsu Oilfield, SINOPEC. This basin is well known for several small, complex fault block reservoirs (Mao et al., 2006). A typical fault block geological model for the Subei Basin is shown in Figure 13*a*, encompassing a model grid of 10 × 10 m. This model consists of a large steep fault, many

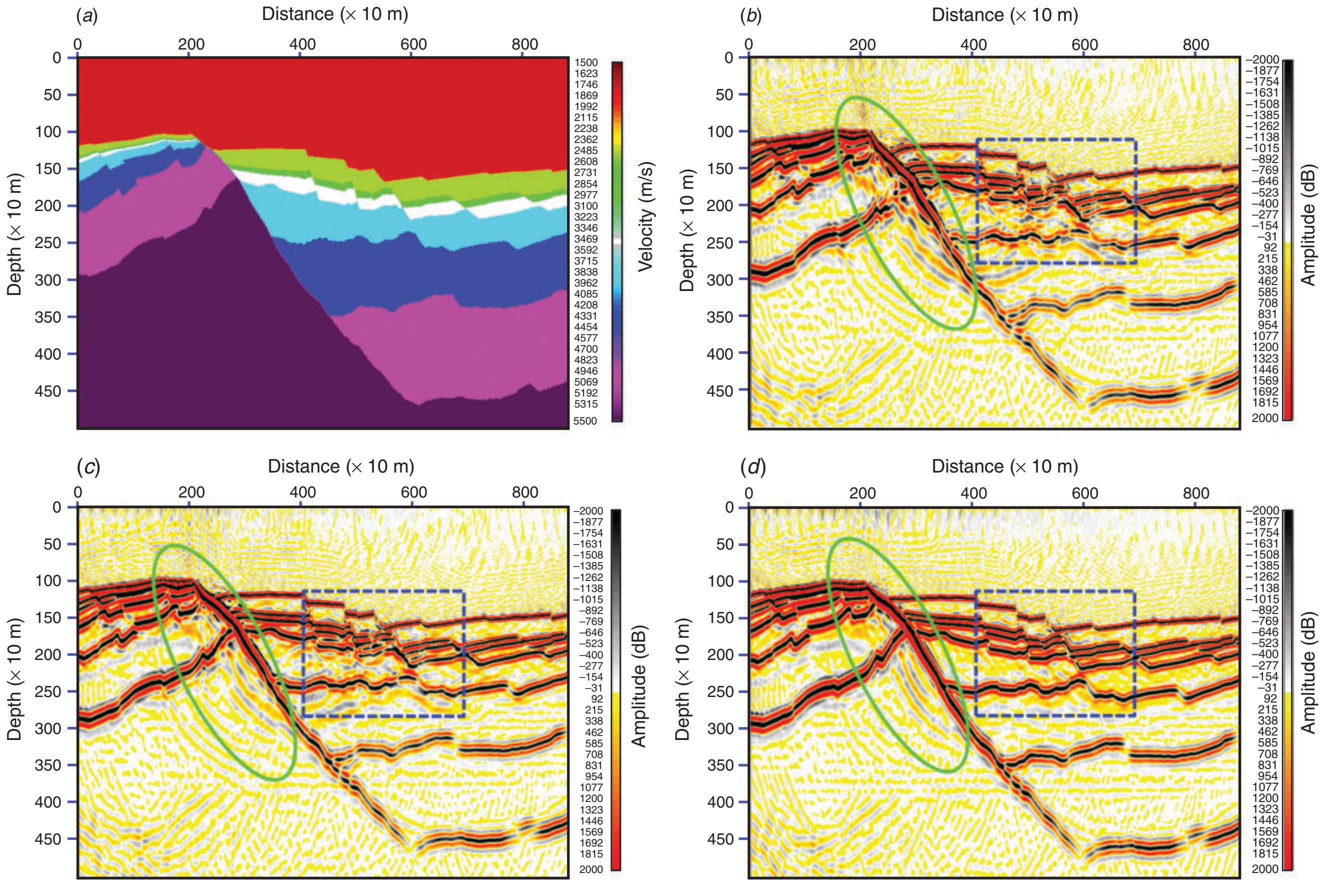

Fig. 13. (*a*) Velocity model based on Subei Basin in China. (*b*) WEM. (*c*) FD-RTM. (*d*) HEP-FE-RTM.

Table 2. Comparison of the computational cost for WEM, FD-RTM and HEP-FE-RTM for Subei Basin model migration imaging using the 8 × 8 blocking scheme.

	Elapsed time	
Method	Single shot (h)	Total 672 shots (days)
WEM	0.982	27.5
FD-RTM	1.655	46.34
HEP-FE-RTM	1.741	48.75

smaller faults and seven reflectors: T23, T24, T25, T30, T31, T33 and T40 from top to bottom.

This forward-modelling seismic dataset contains a total of 672 shots. The migration velocity model encompasses a total of 2278 traces, each of which comprise 501 sampling points spaced at 10 m intervals, and the computational model size is 12 000 × 5000 m. For imaging computations, the random velocity boundary was arranged around the original velocity model, with the exception of the free-boundary surface, and the final size of the solution domain was 14 000 × 7000 m. Our parallel processing approach used an 8 × 8 blocking scheme and 65 processes. Comparison between imaging results obtained using WEM, FD-RTM and the HEP-FE-RTM advocated in this paper are shown in Figure 13*b*–*d*, respectively.

The comparison of the computational elapsed time for WEM, FD-RTM and HEP-FE-RTM methods on Subei Basin model migration imaging using the 8 × 8 blocking scheme is shown in Table 2. Compared to the 1.655 h per shot required by FD-RTM, the HEP-FE-RTM approach took 1.741 h to complete the imaging computation of single shot. Similarly, the HEP-FE-RTM approach took only less than two days longer than the FD-RTM method, but obviously improved imaging accuracy.

Conclusions

A complete HEP-FE-RTM method is proposed in this paper. Compared with traditional approaches, this method incorporates an algorithm in a fine-grained parallel CPU scheme, applies the implicit Newmark time integration method to achieve exact parallel numerical integration computation of the two-way wave propagation equation to strictly ensure integration convergence and makes use of the HEP-FE-RTM to integrate CPUs and process super-large-scaled linear equations. This algorithm can use more than 2000 processors and analyse more than one billion elements (i.e. grids) simultaneously.

However, design of this algorithm could still be improved. First, it would be possible to design a multiple shot coarse-grained parallel algorithm on the basis of single shot parallel computation to process multiple shots simultaneously and complete data stacking. At the same time, an IFE imaging algorithm based on GPU could also be developed. Second, because actual underground media are significantly anisotropic, a scalar wave equation applied based on isotropic theory cannot effectively solve the problem of seismic wave migration in anisotropic media, affecting imaging accuracy. Thus, introduction of an anisotropy vector wave equation is necessary. In terms of the FE method, as the discrete physical quantities of the FE method load onto all nodes, vector wave propagation computation can be easily achieved by increasing degrees of freedom. Third, in the process of wavefield reverse time extrapolation imaging, both the source and receiver wavefields include upgoing and downgoing waves, which will produce a non-zero correlation coefficient on the entire travel path. In this case, by applying the downgoing component of the source wavefield, as well as the upgoing component of the receiver wavefield to make related calculations, imaging noise suppression can be realised.

Conflicts of interest

The authors declare no conflicts of interest.

Acknowledgements

We are grateful for the financial support provided by the National Natural Science Foundation of China (No. 41425017). We also thank the Jiangsu Oilfield of SINOPEC for providing us with a synthetic dataset. In addition, we are very grateful to International Science Editing (ISE) for editing this paper. We especially thank the anonymous reviewers and Professor Nori Nakata (University of Oklahoma) for reviewing this manuscript and giving constructive suggestions.

References

Baysal, E., Kosloff, D. D., and Sherwood, J. W. C., 1983, Reverse time migration: *Geophysics*, **48**, 1514–1524. doi:10.1190/1.1441434

Berenger, J.-P., 1994, A perfectly matched layer for the absorption of electromagnetic wave: *Journal of Computational Physics*, **114**, 185–200. doi:10.1006/jcph.1994.1159

Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M., 1985, A non-reflecting boundary condition for discrete acoustic-wave and elastic-wave equation (shot note): *Geophysics*, **50**, 705–708. doi:10.1190/1.1441945

Clapp, R. G., 2009, Reverse time migration with random boundaries: 79th Annual International Meeting, SEG, Expanded Abstracts, 2809–2813.

Du, Q.-Z., and Qin, T., 2009, Multicomponent prestack reverse-time migration of elastic waves in transverse isotropic medium: *Chinese Journal of Geophysics*, **52**, 471–478. doi:10.1002/cjg2.1367

Fang, Q., 1992, Studies on the accuracy of finite element analysis of implicit Newmark method for wave propagation problems: *Explosions and Shock Waves*, **12**, 45–53.

He, B., Zhang, H., and Han, Y., 2010, Reverse-time depth migration of two way acoustic equations and its parallel algorithm: *Journal of China Coal Society*, **35**, 458–462.

Hemon, C., 1978, Equations d'onde et modeles: *Geophysical Prospecting*, **26**, 790–821. doi:10.1111/j.1365-2478.1978.tb01634.x

Hu, L.-Z., and McMechan, G. A., 1986, Extrapolation in 2-D variable velocity media: *Geophysical Prospecting*, **34**, 704–734. doi:10.1111/j.1365-2478.1986.tb00489.x

Liu, H., Li, B., Liu, H., Tong, X. L., and Liu, Q., 2010, The algorithm of high order finite difference pre-stack reverse time migration and GPU implementation: *Chinese Journal of Geophysics*, **53**, 600–610. doi:10.1002/cjg2.1530

Mao, F. M., Chen, A. D., Yan, Y. F., Ye, S. D., Liu, Q. D., Chen, L. Q., and Tan, Y, 2006, Hydrocarbon pooling features and seismic recognizing technologies of small complex fault blocks in Subei basin: *Oil & Gas Geology*, **27**, 827–839.

Mora, P., 1989, Inversion = migration + tomography: *Geophysics*, **54**, 1575–1586. doi:10.1190/1.1442625

Symes, W. M., 2007, Reverse time migration with optimal checkpointing: *Geophysics*, **72**, SM213–SM221. doi:10.1190/1.2742686

Whitmore, N. D., and Lines, L. R., 1986, Vertical seismic profiling depth migration of a salt dome flank: *Geophysics*, **51**, 1087–1109. doi:10.1190/1.1442164

Zhang, H., and Ning, S., 2002, Pre-stack reverse time migration of elastic wave equations: *Journal of China University of Mining & Technology*, **31**, 371–375.