FULL-WAVEFORM INVERSION BASED ON DEEP LEARNING AND THE TEMPORAL MODIFIED AND SPATIAL OPTIMIZED SYMPLECTIC PARTITIONED RUNGE-KUTTA METHOD

CHENGUANG WANG, YANJIE ZHOU, XIJUN HE, XUYUAN HUANG and FAN LU

School of Mathematics and Statistics, Beijing Technology and Business University (BTBU), Beijing 100048, P.R. China. zhouyanjie9@163.com

(Received August 23, 2022; accepted October 4, 2022)

ABSTRACT

Wang, C.G., Zhou, Y.J., He, X.J., Huang, X.Y. and Lu, F., 2022. Full-waveform inversion based on deep learning and the temporal modified and spatial optimized symplectic partitioned Runge-Kutta method. *Journal of Seismic Exploration*, 31: 501-521.

This study uses deep learning techniques to propose a full-waveform inversion (FWI) method. This method uses the Temporal Modified and Spatial Optimized Symplectic Partitioned Runge-Kutta (TMSOS) method, for the forward modeling in the FWI process. Additionally, optimizer, loss function in deep learning are utilized to perform FWI. We used recurrent neural networks in deep learning to implement the forwarding modeling method–TMSOS. This method uses the second-order MSPRK scheme and an eighth-order optimized finite difference scheme for temporal and spatial discretization, respectively, to obtain high precision with lesser computational effort. The Nadam optimizer, packaged in the Tensorflow software, is used to optimize the model parameters in this study. Additionally, Huber loss is used as the objective function of FWI. Several numerical simulations were done to verify the effectiveness of the proposed method. First, the effectiveness of the TMSOS forward modeling method was compared with the finite difference (FD) method. Then FWI was performed using the TMSOS forward modeling method for the velocity anomaly model, the third-order model, and the complex Sigsbee velocity model. Finally, the inversion results under the three loss functions were compared. The numerical results suggest that the TMSOS forward modeling method has better computational efficiency and smaller resultant numerical dispersion than the traditional FD method. The inversion results of the TMSOS method are closer to those of the actual model, with a better inversion effect. Additionally, the Huber loss function has better stability in selecting the learning rate.

KEY WORDS: TMSOS method, RNN, Nadam optimizer, inversion, Huber loss.

0963-0651/22/$5.00

INTRODUCTION

Oil and gas are a country's lifelines and guarantee its people's livelihood. Oil is processed and refined to obtain essential derivatives, including gasoline, kerosene, diesel, paraffin, and asphalt. Even though vehicles fueled by new energy sources are gaining popularity, the importance of oil remains the same. The demand for natural gas, a clean energy source, is also expected to grow in the future since it can meet the growing energy demands and reduce carbon dioxide emissions. Seismic inversion is a core technology in oil and gas exploration. It is the process of inferring spatial structures and physical properties of subsurface rock formations from available data such as seismic and well log data. However, due to the high temperature and pressure characteristics of the Earth's interior, the deepest interior distance currently explored by humans is only 12 km. Compared with the radius of the Earth, which is approximately 6371 km, human exploration of the Earth's interior is far from over.

The current developments in signal processing, imaging theory, and practical aspects of oil and gas-related seismic exploration have contributed significantly to the study of inversion problems in seismic exploration. Additionally, the development of science and technology has attracted significant attention to the inversion problems (Wang, 2007). Full-waveform inversion (FWI), which is one of the mainstream geophysical exploration methods, uses waveform data and wave equations observed at the surface to infer the structure of the subsurface medium. FWI requires constructing a high-precision and high-efficiency method for numerical computation of the wave equations. The algorithms for computing the wave equations include spatial discretization, temporal discretization, and boundary condition handling (Carcione et al., 2002).

Among the reported methods that involve the solutions of the wave equations and wave field modeling, the more commonly used numerical simulation methods include the finite element method (Hrennikoff, 1941), reflectivity method (Booth and Crampin, 1983; Hanyga, 1986; Fuchs and Müller, 2010), finite difference method (Virieux, 1984, 1986; Dablain, 1986; Igel, 1995; Mora et al., 1995; Wang et al., 2002; Blanch and Robertsson, 2010), spectral element method (Komatitsch et al., 2000), discontinuous finite element method (Rivière and Wheeler, 2003; Michael and Martin, 2006; He et al., 2015), and staggered mesh format (Madariaga, 1976). Each of these methods has its specific advantages and disadvantages. For example, the current finite difference method, widely used in the numerical simulation of seismic waves, is based on a regular grid in the Cartesian coordinate system. Therefore, curved boundaries are bound to appear in numerical simulations of complex geological structure interfaces, and artificial spurious diffracted waves (i.e., numerical dispersion) are present on such boundaries. If a finer grid is used, it leads to increased storage, computational effort, and error accumulation. Furthermore, the numerical dispersion is large, and it adversely affects the information in forward modeling, which reduces the accuracy and resolution of the wavefield modeling. In particular, the correct waveform information is almost indistinguishable after a long-time simulation. Therefore, numerical

dispersion must be minimized in numerical computation. Additionally, numerical dispersion can be reduced by encrypting the grid and improving the accuracy of the numerical format. However, these measures significantly increase computational effort and storage capacity demand. Therefore, developing high-precision numerical simulation methods to reduce numerical dispersion is a fundamental research concern.

The nearly analytical discretization method (NADM) (Yang, 2004; Teng et al., 2003; Lang, 2017), which was applied to solve the two-dimensional acoustic and elastic wave equations, uses a temporal truncated Taylor expansion to characterize wave displacements and first-order partial derivatives, and a spatial truncated Taylor expansion to construct higher-order interpolation functions to approximate higher-order spatial partial derivatives (Yang, 2004; Teng et al., 2003). NADM effectively reduces numerical dispersion with high accuracy since it compensates for the higher-order seismic information due to the discrete characterization of seismic displacements. However, this method requires a large amount of storage capacity. Consequently, NADM was optimized, and an optimum nearly analytical discretization method (ONADM) for two-dimensional cases was obtained (Yang et al., 2003, 2006; Peng et al., 2006). This optimization improves the computational accuracy, efficiency, and resolution of NADM significantly and saves about 53% of the storage volume. Based on this, a new symplectic partitioned Runge-Kutta (NSPRK) method was proposed (Ma et al., 2011). NSPRK method transforms the elastic wave equation into a Hamiltonian system and approximates the higher-order spatial differential operator using a nearly analytic discrete operator. NSPRK can effectively suppress the numerical dispersion caused by the discretization of the wave equations and preserves the symplectic structure for long-time simulation. Additionally, an eighth-order spatial model method, which has eighth-order precision in spatial discretization and can suppress numerical dispersion well was proposed. Liu et al. (2016) proposed an optimized strategy to construct a time-exceeding symplectic structure and used the classical symplectic partitioned Runge-Kutta (SPRK) method for time discretization. Additionally, they introduced a spatial difference term to SPRK to form a modified symplectic partitioned Runge-Kutta (MSPRK) method, which is a modified time-advanced symplectic method with all positive symplectic coefficients. MSPRK method (Liu et al., 2016, 2017), which uses the conventional second-order partitioned Runge–Kutta (PRK) format, attains the SPRK format with third-order precision after the spatial difference term is introduced. Consequently, high precision is achieved with lesser computational effort.

In this study, the time-advanced format of the MSPRK method is analyzed. In terms of spatial discretization, an eighth-order optimized finite-difference format is used to approximate the higher-order spatial derivatives included in the wave equation (Zhang and Yao, 2013). MSPRK method takes an infinite norm of the absolute error between the analytic and numerical wave numbers as the objective function of an optimization problem and solves this optimization problem using the simulated annealing algorithm (Kirkpatrick et al., 1983; Sen and Stoffa, 2013). Subsequently, the

absolute error between the analytic and numerical wave numbers of the obtained optimization coefficients corresponding to the finite difference format is smaller than the absolute error of the classical finite difference format. Thus, an efficient forward modeling method is developed.

FWI can be implemented in the time, frequency, and hybrid domains. Generally, after obtaining the gradients of a model by the adjoint state method, the model is continuously updated using optimization algorithms such as linear search or trust domain. Therefore, mathematically interpreting, FWI is a nonlinear optimization problem. Additionally, selecting an optimization algorithm is crucial in solving an optimization problem. Currently, the most widely used optimization algorithms for FWI include the Fortan Subroutines for Large-Scale Bound-Constrained Optimization (L-BFGS-B) method (Zhu, 1997; Byrd et al., 1997; Rao and Wang, 2017), which is based primarily on the gradient information of the model, the steepest descent method, and the conjugate gradient method. These algorithms are relatively simple to implement and interpret, and the results achieved in parametric model inversion are satisfactory.

The advancement of artificial intelligence has led to the maturation of the theory and algorithms of artificial neural networks, which find applications in many fields, especially in computer vision and natural language processing. Deep learning is a machine learning algorithm for training deep neural networks. The recurrent neural network (RNN) is mainly used to process sequential data. Compared to traditional neural network models, RNN model layers are not fully connected to each other. Therefore, RNN memorizes the information before the current moment and applies it to compute the current output. Thus, the RNN characteristics like memorability and parameter sharing are used to solve sequential input problems. Due to extensive research on RNN, many network models have been proposed to solve the problem of long-time dependency of sequences. Additionally, the deep learning algorithm uses back propagation and automatic differentiation to update network parameters by computing loss functions.

In this study, first, we introduce TMSOS for solving acoustic waveform equations as the forward modeling method for FWI. TMSOS method effectively suppresses numerical dispersion under the coarse spatial grid condition and can achieve higher-order precision with lesser computational effort. Then, with respect to the deep learning strategy, we select an RNN framework to characterize the temporal iterative format of the TMSOS method. Additionally, based on the Tensorflow framework, we use the Nadam optimizer (Dozat, 2016), small-batch strategy, and automatic differentiation to improve the precision and computational efficiency of the inversion. Finally, we use numerical simulations to validate the effectiveness of the TMSOS method.

TMSOS for solving a two-dimensional acoustic equation

In a two-dimensional homogeneous isotropic medium, the acoustic wave equation can be written as:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad , \tag{1}$$

where c, u, and t denote velocity of the acoustic wave propagating in the medium, displacement, and time, respectively. The variable $v = \frac{\partial u}{\partial t}$, where v denotes the velocity, is introduced to rewrite eq. (1) in the form of a Hamiltonian system (Ma et al., 2011). Then the two-dimensional acoustic wave equation can be rewritten in the form of a Hamiltonian system as:

$$\begin{cases} \dfrac{\partial u}{\partial t} = v \\ \dfrac{\partial v}{\partial t} = Lu \end{cases}, \tag{2}$$

where $L = c^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2} \right)$ is the spatial operator of the acoustic wave equation and is a linear function with respect to u. Thus, System (2) is a linearly differentiable Hamiltonian system of the two-dimensional acoustic wave equation with Hamiltonian function, $H(u,v) = \frac{1}{2}\int_\Omega v^2 + c^2 (\nabla u)^T \nabla u d\sigma$. Then, we define the generalized displacement vector, $U = \left(u, \partial u / \partial x, \partial u / \partial z\right)^T$ and the particle velocity vector, $V = \left(v, \partial v / \partial x, \partial v / \partial z\right)^T$. Thus, eq. (2) can be written in the following form:

$$\begin{cases} \dfrac{\partial U}{\partial t} = V \\ \dfrac{\partial V}{\partial t} = LU \end{cases}, \tag{3}$$

We discretize the temporal iteration format using the MSPRK format to solve eq. (3) (Liu et al., 2017). First, we derive the traditional third-order SPRK format.

$$\begin{cases} U_1 = U_n + c_1 \Delta t V^n, \\ V_1 = V^n + d_1 \Delta t \boldsymbol{L} U_1, \\ U_2 = U_1 + c_2 \Delta t V_1, \\ V_2 = V_1 + d_2 \Delta t \boldsymbol{L} U_2, \\ U^{n+1} = U_2 + c_3 \Delta t V_2, \\ V^{n+1} = V_2 + d_3 \Delta t \boldsymbol{L} U^{n+1}。 \end{cases} \tag{4}$$

According to the theoretical analyses and experimental results, the third-order SPRK format shown in eq. (4) is better than the second-order SPRK format obtained by different optimization approaches. However, the third-order format requires extensive computation compared to second-order format. Therefore, Liu et al. proposed that a term $\boldsymbol{L}v_1$ can be added to the third equation in the second-order SPRK format to improve its precision (Liu et al., 2016, 2017; Yang et al., 2016, 2017). Thus, the MSPRK format can be written as:

$$\begin{cases} U_1 = u^n + c_1 \Delta t V^n, \\ V_1 = v^n + d_1 \Delta t \boldsymbol{L} U_1, \\ U^{n+1} = U_1 + c_2 \Delta t V_1 + c_3 \Delta t^3 \boldsymbol{L} V_1, \\ V^{n+1} = V_1 + d_2 \Delta t \boldsymbol{L} U^{n+1} \end{cases} \tag{5}$$

where Δt, $\boldsymbol{L}$, and $c_i \left(i = 1,2,3\right), d_j \left(j = 1,2\right)$ are time increment, finite difference operator of the eighth-order optimization, and system coefficients of MSPRK, respectively. Detailed information about the system coefficients of MSPRK and the optimized finite difference method is given in the Appendix.

WRITTEN INTO RNN FRAMEWORK VIA TMSOS

An RNN consists of an input layer, a hidden layer, and an output layer. The output layer contains a classifier and a label. The classifier, also called the Softmax layer, classifies and estimates the output in the hidden layer and selects the one with the highest probability as the label.

RNN remembers the previous information and applies it to the current output calculation. Therefore, an important characteristic of the RNN is that the current output of a sequence is related to the previous output. The hidden layers are connected to each other, and their inputs contain not only the output of the input layer but also the output of the hidden layer at the previous moment. RNN is used to solve sequential input problems due to its memorability and parameter-sharing characteristics. The intensive research on RNN has led to more and more network models being proposed to solve the long-time dependence problem of sequences. Fig. 1 shows the schematic diagram of RNN.

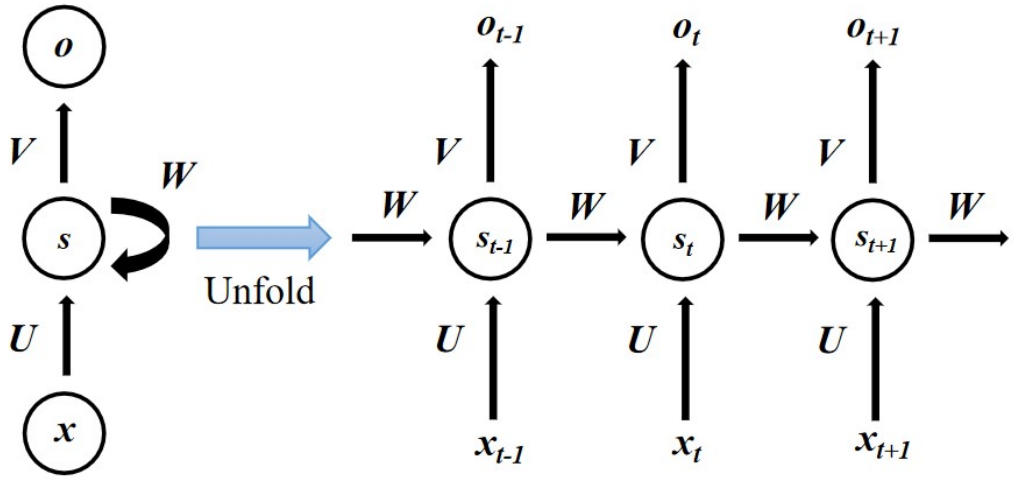

Fig. 1. Schematic diagram of RNN, where x is the input sample, s is the hidden layer, o is the output, W is the weight of the input, U is the weight of the input sample, and V is the weight of the output (Elman, 1990).

RNN works similar to most traditional forward modeling methods, which use the previous moment's wavefield to calculate the current moment's wavefield. Therefore, we present a traditional forward modeling method with an RNN framework. RNN framework and the convolution kernel can simplify the writing of the program, since the RNN framework can make the programming of the temporal iterative format faster, and the convolution kernel can write the higher-order derivatives easily. Additionally, the RNN can combine the forward modeling method with deep learning techniques. Consequently, high-precision numerical algorithms can be used to solve the forward modeling problem, and deep learning techniques, including optimizers and small-batch training strategies, can be used to solve the inversion problem. Additionally, parallel computation can also be performed on a distributed storage-based computer cluster to improve computational efficiency. Fig. 2 shows the schematic diagram of the RNN combined with the TMSOS method (RNN–TMSOS).

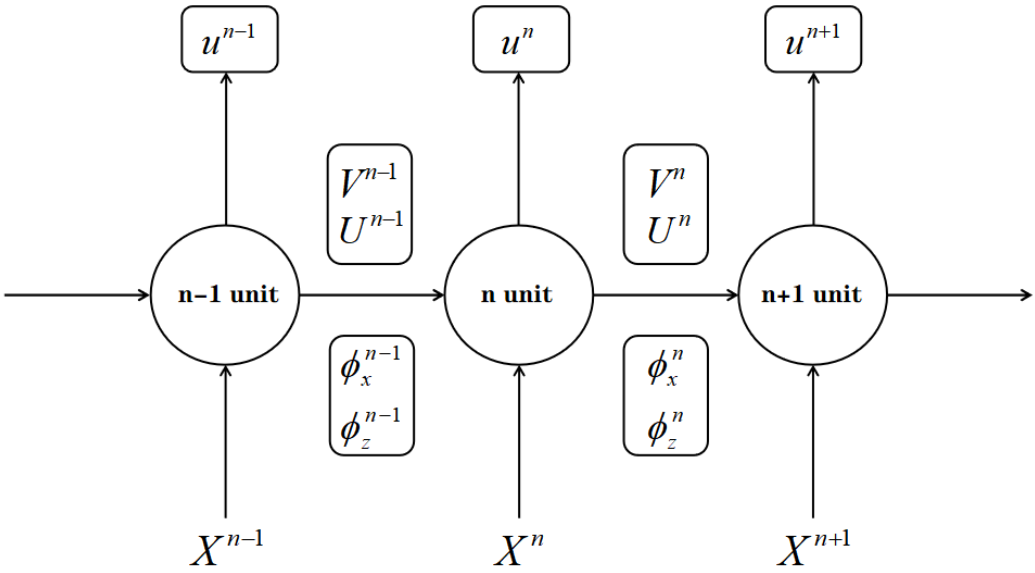

Fig. 2. Schematic diagram of RNN–TMSOS forward modeling method.

LOSS FUNCTION AND OPTIMIZER SELECTION

Inversion is essentially an optimization problem to minimize the error between real-world and synthetic data. The real-world data is the waveform information of the seismic wave received by a geophone. In contrast, the synthetic data is the simulated waveform information obtained from the current velocity model.

We often refer to the minimization function as the loss function, which mainly measures the predictive power of machine learning models. The gradient descent algorithm is the most common method for finding the minimization function. Although a loss function identifies the strengths and weaknesses of a model and provides directions for optimization, no single loss function is applicable to all models. The selection of the loss function depends on several factors such as the number of parameters, outliers, machine learning algorithms, the efficiency of the gradient descent algorithm, and the difficulty level in derivation. Commonly used regression loss functions include L^2 MSE (loss function) and L^1 MAE (loss function). The MSE is the quadratic sum of the differences between the predicted and the target values. The MAE is the sum of the absolute values of the difference value between the target and the predicted values, which characterizes the average error margin in the predicted value without considering the direction of the error. The equations for the MSE and the MAE loss functions are written as:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y_i^p)^2}{n}, \tag{6}$$

$$MAE = \frac{\sum_{i=1}^{n} \left| y_i - y_i^p \right|}{n} \quad , \tag{7}$$

where y_i is the real-world value and y_i^p is the predicted value.

The MSE and the MAE have their limitations. The MSE shows a much more significant loss at larger error points than the MAE. Additionally, the MSE assigns more weight to the outliers, which decreases the model's overall performance since it will try to reduce the error caused by the outliers. In contrast, the MAE is more stable in handling outliers. However, the gradient of the MAE undergoes a large jump at the extremes, especially for neural networks, which can be detrimental to the learning process since even small loss values can produce large errors. The learning rate needs to be dynamically reduced during the resolution of the extrema to solve this problem. The MSE has good properties at the extrema and converges even at a fixed learning rate. Its gradient decreases as the loss function decreases, making it possible to obtain more accurate results in the final training process.

In this study, the Huber loss is used to reduce the effect of the other loss functions on the inversion results. The Huber loss is less sensitive to outliers than the MSE loss. However, the Huber loss retains the property of derivability compared to the MAE loss. Therefore, the Huber loss has the advantages of reducing the sensitivity to outliers and avoiding model overfitting to a certain extent. Additionally, it achieves derivability everywhere, ensuring the stability of the model at extreme points (Ma, 2020; Li et al., 2020). Although the Huber loss is based on the MAE, it is turned into the MSE when the error is small. Additionally, the hyper-parameter δ can be used to adjust the threshold of this error. The Huber loss degenerates to the MAE when δ tends to 0 and to the MSE when δ tends to infinity. The expression for the Huber loss, which is a continuously differentiable piecewise function, is written as:

$$L_\delta(y, y^p) = \begin{cases} \frac{1}{2}(y - y^p)^2 & for \left| y - y^p \right| \le \delta, \\ \delta \left| y - y^2 \right| - \frac{1}{2}\delta^2 & otherwise. \end{cases} \quad . \tag{8}$$

The hyper-parameter δ is chosen based on the predicted and real-world values. Finally, in numerical computation, experiments demonstrate the stability and effectiveness of the Huber loss in inversion.

A small-batch training strategy is used to train the model once instead of training all sample data. The optimizer used is the Nadam optimizer packaged in the Tensorflow software, which integrates ideas from the Nesterov momentum and Adam optimizer. Several published experiments suggest that the Nadam optimizer is more effective and provides better

guidance for gradient descent than the Adam optimizer (Dozat et al., 2016; Zeyer et al., 2017). Hyper-parameters such as learning rate and small-batch size of the RNN have no standard selection method in this study. The choice of these hyper-parameters depends on the specific situation and model setting.

NUMERICAL SIMULATION

Numerical simulations are used to test the effectiveness of the TMSOS method. First, a homogeneous model and a corner–edge model are constructed to test the effectiveness of the method on forward modeling. Subsequently, a velocity anomaly model is constructed to perform FWI. Additionally, the effect of the learning rate on the experimental results is tested. Finally, the TMSOS method's effectiveness for complex model-based inversion is tested with the Sigsbee model.

Forward modeling of the homogeneous model

As shown in Fig. 3(a), the effectiveness of the TMSOS–RNN forward modeling is tested and compared with the conventional FD–RNN forward modeling. The area of the homogeneous model is set to 6.0 km × 6.0 km, with $\Delta x = \Delta z = 40$ m as spatial step, $\Delta t = 0.001$ s as temporal step, and 5000 m/s velocity. The hypocenter frequency, which is located at the center of the computational domain, is 20 Hz. The receiver setting is R_1 (3.0 km, 0 km). Finally, the boundary conditions are absorbed using the perfect matching layer (PML).

We compare the forward modeling results between TMSOS-RNN and FD-RNN. The wavefield snapshots of TMSOS and FD at T = 0.5 s are shown in Figs. 3(b) and 3(c), respectively. The waveforms in Fig. 3(b) are clearer with no obvious numerical dispersion. However, the waveform in Fig. 3(c) shows an obvious numerical dispersion.

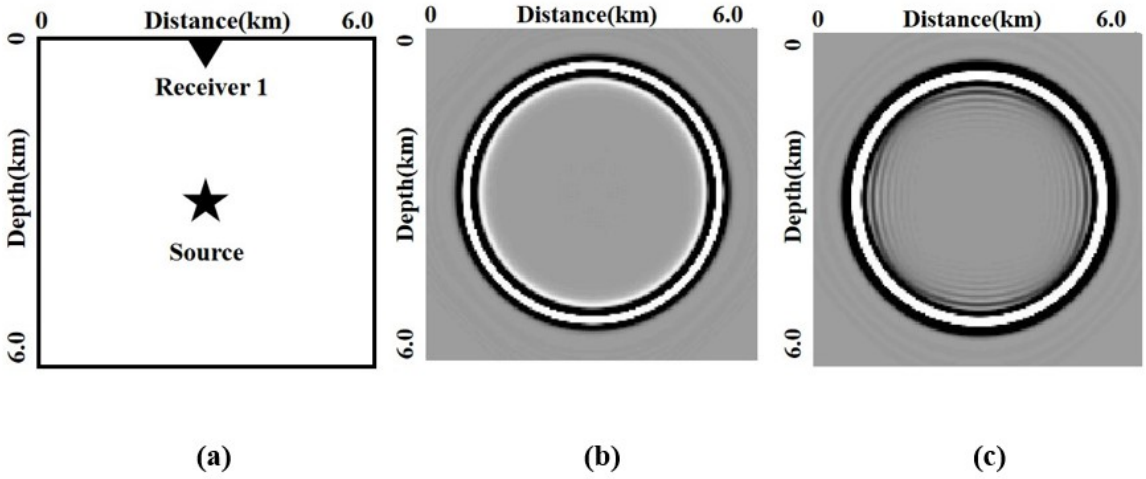

Fig. 3. The area of the homogeneous model in Fig. 3 (a). Wavefield Snapshot of TMSOS and FD at T = 0.5 under coarse mesh are shown in Figs. 3(b) and 3(c).

The Courant number is fixed, and the spatial mesh is refined to eliminate the numerical dispersion derived from the FD method. Successful mesh refinement is defined as the presence of visible numerical dispersion. The spatial step in the FD method is 30 m. Fig. 4 shows the wavefield snapshot of the FD method at T = 0.5 s under the fine mesh. At this point, the FD method yields few visible numerical dispersion. However, the computation time consumed by the two methods in achieving dispersion elimination is different. The TMSOS method takes about 2.734 s, whereas the FD method under the fine mesh takes about 4.121 s to achieve dispersion elimination. Therefore, the TMSOS method is superior to the FD method in computation time. The spatial step, number of mesh points, and computation time of the two methods are shown in Table 1.

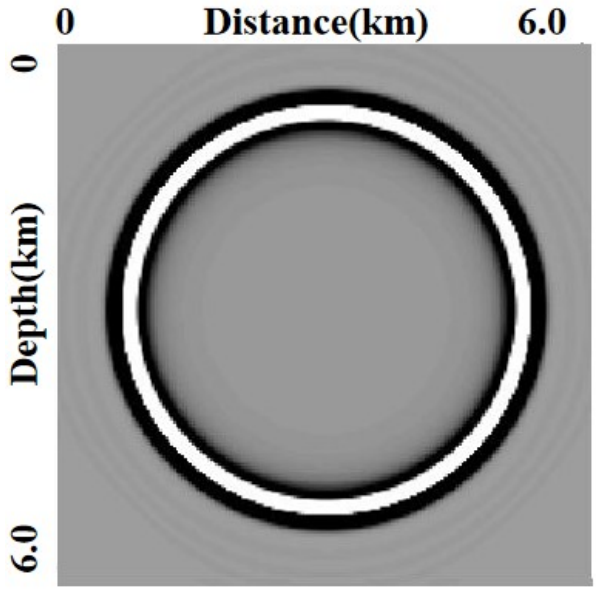

Fig. 4. Wavefield snapshot of FD at T = 0.5 s under fine mesh condition.

Table 1. Spatial step, number of mesh points, and computation time of the TMSOS and FD methods.

	Spatial step (m)	Number of mesh points	Computation time (s)
TMSOS	40	150×150	2.734
FD	30	200×200	4.121

Forward modeling of the corner–edge model

As shown in Fig. 5(a), a complex corner–edge model consisting of three domains, namely, domain I (2 km/s), domain II (3 km/s), and domain III (4 km/s) was incorporated. The overall size of the computational domain is 4.8 km × 4.8 km. The position coordinates of domains I and III are [0 km, 4.8 km] × [0 km, 0.96 km] and [1.76 km, 4.8 km] × [1.76 km, 4.8

km], respectively. The hypocenter frequency is 20 Hz. Additionally, the spatial step $\Delta x = \Delta z = 12$ m and the temporal step Δt = 0.001 s. Fig. 5 (b) and (c) shows the wavefield snapshots generated by TMSOS–RNN at T = 0.3 s and T = 0.5 s, respectively. The snapshots show complex reflected and transmitted waves from the horizontal inner interface and the vertical inner interface, respectively, without any visible numerical dispersion. This suggests that the TMSOS–RNN method can effectively simulate wave propagation in comparable complex geological situations.

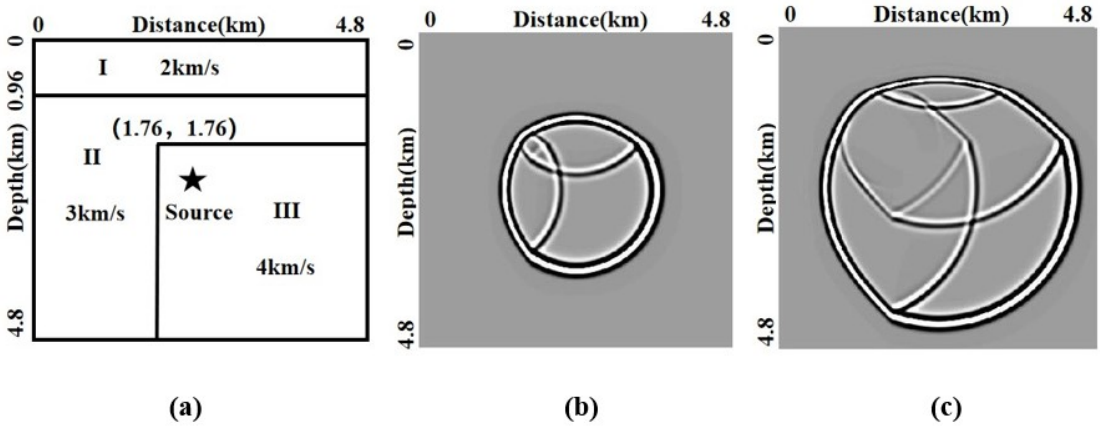

Fig. 5. (a) Corner–edge model. (b) Wavefield snapshots generated by TMSOS–RNN at T = 0.3 s. (c) Wavefield snapshots generated at T = 0.5 s.

VELOCITY ANOMALY MODEL

As shown in Fig. 6(a) , a velocity anomaly model is constructed to study the effect of the three loss functions on the inversion results. The size of the velocity anomaly model is set to 300 m × 300 m, with a spatial step, $\Delta x = \Delta z = 6$ m and a temporal step, Δt = 0.001 s. The hypocenter frequency is 20 Hz, and the velocities of domains I, II, and III are 800, 1000, and 1200 m/s, respectively. The position coordinates of domains I and III are [0 km, 0.3 km] × [0 km, 0.09 km] and [0.12 km, 0.18 km] × [0.17 km, 0.23 km], respectively. Gaussian perturbation of the real model with a standard deviation of 10% is performed to generate the initial inversion model. The real velocity model and the initial model are shown in Fig. 6 (b).

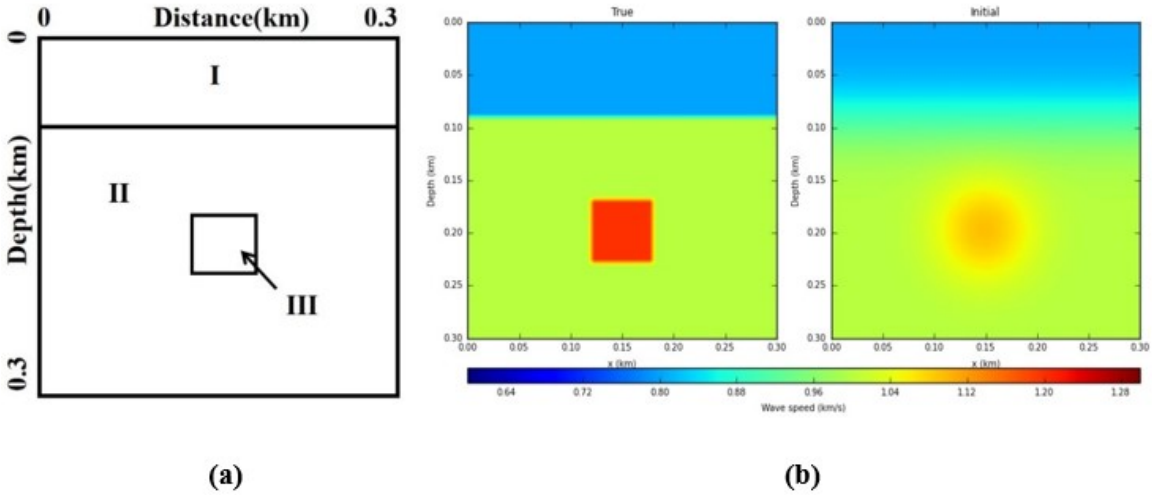

Fig. 6. (a) The Velocity anomaly model. (b) Left: real model; right: initial model.

Subsequently, we perform FWI at five different learning rates, 5, 15, 30, 50, and 70, using the MAE, MSE, and Huber loss functions. The inversion results derived using the three loss functions as the objective function at the different learning rates are shown in Fig. 7. The results suggest that when the learning rates are relatively low, the inversion effects of the three loss functions do not differ much (rank from best to worst: Huber □MAE □MSE). However, as the learning rate increases, the Huber loss exhibits higher stability, and its inversion effect is much better than the MAE and the MSE. Furthermore, the inversion results under Huber loss are better than the MAE and the MSE at all stages. To compare the Huber loss with the MAE and MSE rigorously and intuitively, we define an error *L,* as the L_1 norm between the predicted and real-world values at each mesh point of the inversion results. Furthermore, smaller *L* values indicate greater consistency between the inversion and real model results.

$$L = \sum_{i=0}^{N} \sum_{j=0}^{M} \frac{\left| y_{i,j} - y_{i,j}^{p} \right|}{n} \quad , \tag{10}$$

where *N* and *M* are the numbers of rows and columns of the mesh, respectively. $y_{i,j}$ and $y_{i,j}^{p}$ are the velocities at the *i*-th row and *j*-th column of the mesh point in the inverse velocity model and real velocity model, respectively. n is the number of mesh point.

Fig. 8 is the broken line graph obtained by plotting the error *L* of the MAE, MSE, and Huber loss functions under five different learning rates in the coordinate system. This shows the characteristics of the three loss functions. When the learning rate is low, the errors *L* of MAE and the Huber loss are closer due to the small effect of the learning rate on the MAE. The sensitivity of outliers causes the error *L* of the MSE to be significantly larger

than that of the MAE and the Huber loss. As the learning rate increases, MAE cannot converge at the extreme value point, resulting in the error L of the MAE exceeding that of the MSE. Fig. 8 also shows that the growth of the Huber loss is flatter, indicating that the Huber loss has better stability at higher learning rates. The error L calculated for the MAE, MSE, and Huber loss functions at the different learning rates are given in Table 2.

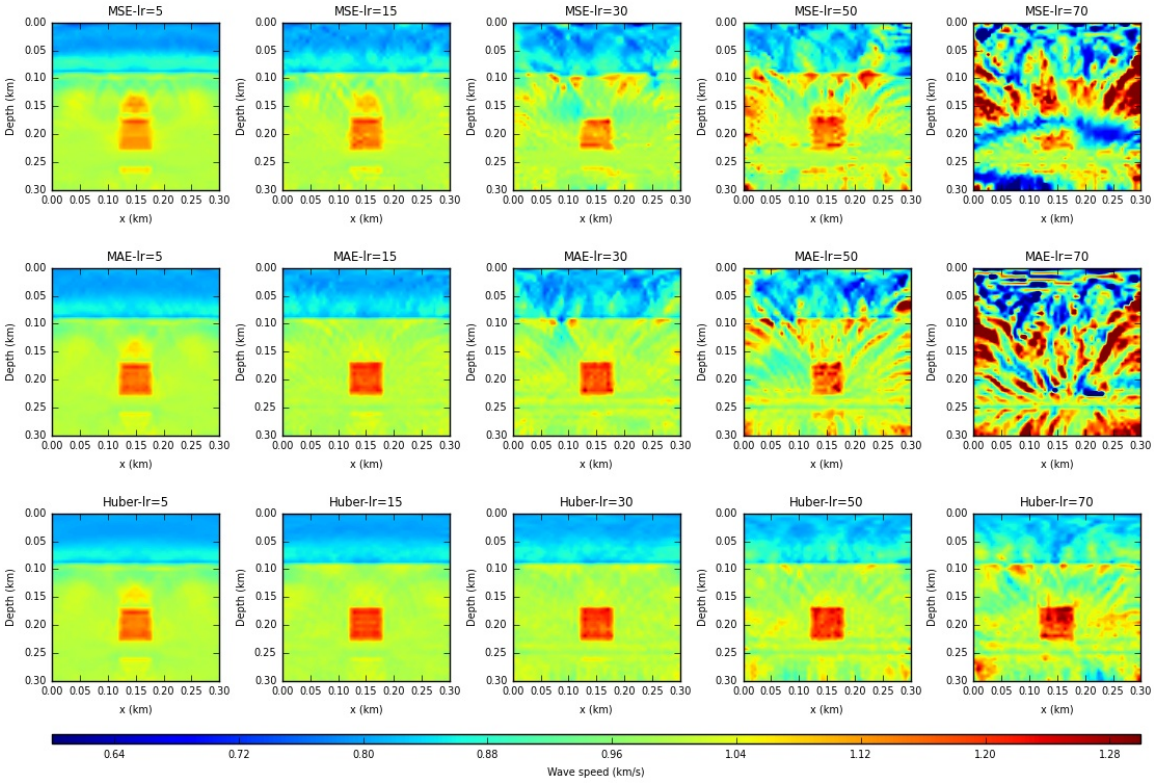

Fig. 7. Inversion results obtained by TMSOS–MSE, TMSOS–MAE, and TMSOS–Huber loss functions at different learning rates.

Table 2. Errors L calculated for the three TMSOS-based loss functions at five learning rates (5, 15, 30, 50, and 70).

Learning rates	*5*	*15*	*30*	*50*	*70*
MSE	1.275	1.183	1.399	2.359	5.175
MAE	1.118	1.127	1.302	3.089	5.291
Huber	1.086	1.055	0.981	1.315	2.051

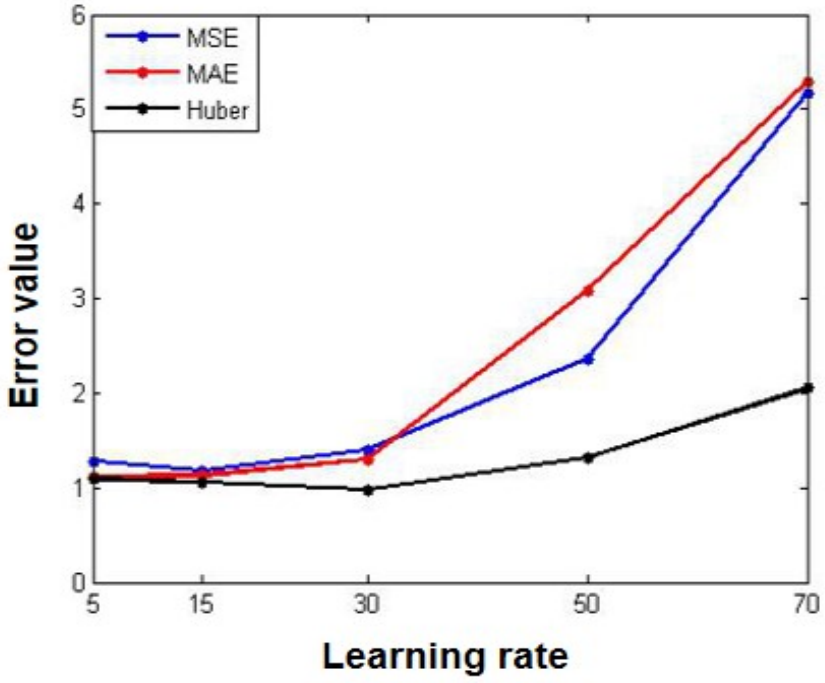

Fig. 8. Broken line graph obtained by plotting the error L of the TMSOS–MSE, TMSOS–MAE, and TMSOS–Huber loss functions at different learning rates in the coordinate system

THIRD-ORDER LADDER MODEL

A third-order ladder model, as shown in Fig. 9, is constructed. Its size is set to 600 m × 600 m, with a spatial step $\Delta x = \Delta z = 6$ m, and a temporal step $\Delta t = 0.001$ s. The hypocenter frequency is 20 Hz, and the velocities of domains I, II, and III are 800, 1200, and 1600 m/s, respectively.

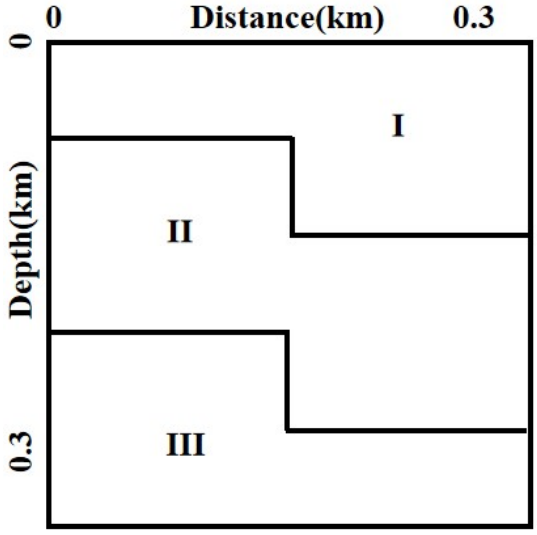

Fig. 9. Third-order ladder model.

The TMSOS method adopts the second-order MSPRK method in spatial or temporal discrete format. This method has a precision close to the third-order precision and is more efficient than the third-order model in computational efficiency. The advantages of the TMSOS method in terms of precision and computational efficiency are tested by inversion of the third-order model using TMSOS and the traditional third-order SPRK, respectively. The initial model is generated by +10% Gaussian perturbation of the real model. The number of iterations, errors, and time required for iteration for the two models are displayed in Table 3. A smaller difference in the inversion effect implies that the iteration time required for the TMSOS method is much shorter than that of the traditional third-order SPRK method. Therefore, TMSOS has a higher computational efficiency while achieving a precision close to that of the third-order model.

Table 3. Number of iterations, errors, and time required for iteration for the two methods.

	Number of iterations	Errors	Time required for iteration
TMSOS	135	0.508	2922.63s
Third-order SPRK	135	0.481	5180.35s

SIGSBEE MODEL

Finally, the complex Sigsbee model is studied. The size of the sampled model is 101 × 61, with spatial step $\Delta x = \Delta z = 8$ m, and temporal step $\Delta t = 0.001$ s. The velocity ratio of the high-velocity salt mound to the sediment, and the gradient structure of the sediment lead to a strong multiple scattered wavefield, making precise inversion of the model difficult. Fig. 10 shows the real and initial models as well as their inversion results. The initial model is generated by performing a Gaussian perturbation on the real model with a standard deviation of 5. There are 101 hypocenters and 101 receivers. All receivers were placed horizontally with an interval distance of 8 m at a distance of 0.024 km below the surface. Additionally, a small-batch inversion strategy with a batch size of 5, a traversal number of 5, and a learning rate of 15 is used. The velocities in the real model range from 1400 to 4600 m/s. The bottom of the initial model is slightly different from the real model, which basically characterizes the high-velocity salt mounds in the high-velocity region. This indicates the TMSOS method's effectiveness in performing complex model inversions.

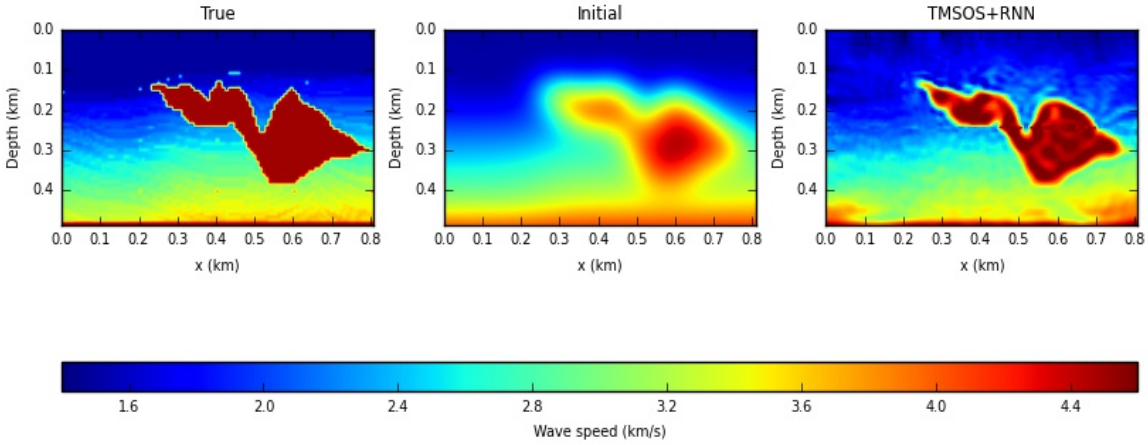

Fig. 10. Inversion results of the real model, initial model, and the TMSOS method.

CONCLUSION

In this study, the MSPRK method, TMSOS, is combined with RNN (TMSOS–RNN) as the forward modeling method used in the FWI process. Additionally, deep learning techniques such as small-batch strategies and optimizers are used to implement FWI. Furthermore, the Huber loss function is introduced. Two forward modeling methods and three inversion modeling methods are designed. The results of the forward modeling methods suggest that the TMSOS–RNN method provides precise wavefield information and can effectively suppress numerical dispersion compared to the conventional FD–RNN method. The first inversion model, velocity anomaly inversion model, reveals that the Huber loss has higher stability and higher learning rates than the MSE and MAE. The second inversion model, the third-order ladder model, suggests that the TMSOS method adopts a second-order format for time discretization but has a precision close to the third-order format and is more efficient than the third-order format in computational efficiency. Finally, the Nadam optimizer is used to invert a complex velocity model, the Sigsbee model. The results of this model suggest that deep learning techniques can be used to implement FWI reasonably and efficiently. Therefore, deep learning can be used to implement forward modeling and inverse modeling. This study only incorporates the forward and inverse modeling of a two-dimensional acoustic wave equation. However, it is possible to extend the results obtained to the three-dimensional acoustic wave equations and the elastic wave equations, which we intend to study in the future. Furthermore, selecting the hyper-parameters of the Huber loss function is a key factor in determining the inversion results. In this study, the hyper-parameters are selected based on the experimental results. Therefore, a systematic study for the selection of hyper-parameters is possible. Furthermore, factors such as the effect of the learning rate decay strategy on inversion results, the effect of the initial-value model on inversion results, and the selection of certain optimizers need to be further explored.

ACKNOWLEDGMENTS

We thank Alan Richardson for providing his open source code on https://arxiv.org/abs/1801.07232. His code provides an important basis for the progress of this study.

FUNDING

This study was supported by Supported by the National Key Research and Development Project of China (Grant No. 2017YFC1500301), Joint key project of the National Science Foundation of China and the China earthquake administration (Grant No. U1839206).

REFERENCES

Blanch, J.O. and Robertsson, J., 2010. A modified Lax-Wendroff correction for wave propagation in media described by Zener elements. Geophys. J. Roy. Astronom. Soc., 131: 381-386.

Booth, D.C. and Crampin, S., 1983. The anisotropic reflectivity technique: anomalous reflected arrivals from an anisotropic upper mantle. Geophys. J. Roy. Astronom. Soc., 72: 767-782.

Carcione, J.M., Herman, G.C. and ten Kroode, A., 2002. Seismic modeling. Geophysics, 67: 1304-1325.

Dablain, M.A., 1986. The application of high-order differencing to the scalar wave equation. Geophysics, 51: 54-66.

Dozat, T., 2016. Incorporating Nesterov Momentum into Adam. http://cs229.stanford.edu/proj2015/054_report.pdf.

Elman, J.L., .1990. Finding structure in time. Cognit. Sci., 14: 179-211.

Fuchs, K. and Müller, G., 2010. Computation of synthetic seismograms with the reflectivity method and comparison with observations. Geophys. J. Roy. Astronom. Soc., 23: 417-433.

Hanyga, A., 1986. Gaussian beams in anisotropic elastic media. Geophys. J. Roy. Astronom. Soc., 85: 473-504.

He, X., Yang, D. and Wu, H., 2015. A weighted Runge-Kutta discontinuous Galerkin method for wavefield modelling. Geophys. J. Internat., 200: 1389-1410.

Hrennikoff, A., 1941. Solution of problems of elasticity by the frame-work method. J. Appl. Mechan. A, 8: 169-175.

Igel, H., Mora, P. and Riollet, B., 1995. Anisotropic wave propagation through finite-difference grids. Geophysics, 60: 1203-1216.

Virieux, J., 1984. SH-wave propagation in heterogeneous media; velocity-stress finite-difference method. Geophysics, 49: 1933-1957.

Virieux, J., 1986. P-SV wave propagation in heterogeneous media; velocity-stress finite-difference method. Geophysics, 51: 889-901.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. Science, 220: 671-680.

Komatitsch, D., Barnes, C. and Tromp, J., 2000. Simulation of anisotropic wave propagation based upon a spectral element method. Geophysics, 65: 1251-1260.

Lang, C., 2017. Frequency-domain full waveform inversion based on nearly analytic discrete methods. Ph.D. Thesis, Tsinghua University, Tsinghua.

Liu, S., Yang, D., Lang, C., Wang, W. and Pan, Z., 2016. Modified symplectic schemes with nearly-analytic discrete operators for acoustic wave simulations. Comput. Phys. Communic., 213: 52-63.

Liu, S., Yang, D. and Ma, J., 2017). A modified symplectic PRK scheme for seismic wave modeling. Comput. Geosci., 99: 28-36.

Ma, S., Li, D., Hu, T., Xing, Y. and Nai, W., 2020. Huber Loss function based on variable step beetle antennae search algorithm with Gaussian direction. 12th Internat. Conf. Intellig. Human-Machine Syst. Cybernet. (IHMSC).

Ma, X., Yang, D. and Liu, F., 2011. A nearly analytic symplectically partitioned Runge–Kutta method for 2-D seismic wave equations. Geophys. J. Internat., 187: 480-496.

Madariaga, R., 1976. Dynamics of an expanding circular fault. Bull. Seismol. Soc. Am., 66: 639-666.

Michael, D. and Martin, K., 2006. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - I. The two-dimensional isotropic case with external source terms. Geophys. J. Internat., 167: 319-336.

Rao, Y. and Wang, Y., 2017. Seismic waveform tomography with shot-encoding using a restarted L-BFGS algorithm. Scient. Rep., 7: 1-9.

Rivière, B. and Wheeler, M.F., 2003. Discontinuous finite element methods for acoustic and elastic wave problems. Contemp. Mathemat., 329: 271-282.

Sen, M.K. and Stoffa, P.L., 2013. Global Optimization Methods in Geophysical Inversion. Cambridge University Press, Cambridge, 4: 1-22.

Wang, S., Yang, D. and Yang, K., 2002. Compact finite difference scheme for elastic equations. J. Tsinghua Univ. (Sci. Technol.), 42: 1128-1131.

Wang, Y., 2007. Calculation method of inversion and its application. Inst. Geol. Geophys., Chinese Academy of Sciences 2006 Ann. Conf., Beijing, China.

Yang, D., 2004. An optimal nearly analytic discrete method for 2D acoustic and elastic wave equations. Bull. Seismol. Soc. Am., 94: 1982-1991.

Yang, D., Peng, J., Lu, M. and Terlaky, T., 2006. Optimal nearly analytic discrete approximation to the scalar wave equation. Bull. Seismol. Soc. Am., 96: 1114-1130.

Yang, D., Teng, J., Zhang, Z. and Liu, E., 2003. A nearly analytic discrete method for acoustic and elastic wave equations in anisotropic media. Bull. Seismol. Soc. Am., 93: 882-890.

Zhang, J. and Yao, Z., 2013. Optimized explicit finite-difference schemes for spatial derivatives using maximum norm. J. Computat. Phys., 250: 511-526.

Zhu, C., Byrd, R.H., Lu, P. and Nocedal, J., 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. Acm Transact. Mathemat. Softw., 23: 550-560.

APPENDIX

SYSTEM COEFFICIENTS OF MSPRK AND OPTIMIZATION OF THE FINITE DIFFERENCE METHOD

A second-order MSPRK format to discretize time was adopted to solve the acoustic equation. The system coefficients of MSPRK given in eq. (6) are as follows:

$$\begin{cases} d_1 = 2/3, \\ d_2 = 1/3, \\ c_1 = 1/4, \\ c_2 = 3/4, \\ c_3 = 1/24. \end{cases} \tag{A-1}$$

Tables A-1 and A-2 show the optimization coefficients of the first-order and second-order derivatives in finite-difference format for different operator lengths when the error limit is 0.0001, respectively. Zhang et al. verified that the absolute errors of the numerical and analytic wave numbers corresponding to these optimization coefficients in the finite-difference format are smaller than those of the classical finite difference format.

The optimal finite difference expression of the first-order derivative is written as:

$$\left.\frac{\partial u}{\partial x}\right|_{x=x_0} \approx \frac{1}{\Delta x}\sum_{j=-N}^{N} b_j^1 u(x_0 + j\Delta x), \tag{A-2}$$

where the coefficient b_j^1 ($b_{-j}^1 = -b_j^1 (j = 1,2,\mathrm{K}\ ,N)$ is shown in Table 1, and 2N is the order of the optimized finite-difference format.

The optimal finite difference expression of the second-order derivative is as follows:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x=x_0} \approx \frac{1}{\Delta x^2}\sum_{j=-N}^{N} b_j^2 u(x_0 + j\Delta x), \tag{A-3}$$

where the coefficient b_j^2 is shown in Table 2. Unlike the first-order derivative, $b_{-j}^2 = b_j^2 (j = 1,2,\mathrm{K}\ ,N)$, and 2N is the order of the optimized finite-difference format. For the optimized finite-difference format of the second-order derivative, the coefficient satisfies:

$$b_0^2 = -2\sum_{j=1}^{N} b_j^2 = -\left(\sum_{j=-N}^{-1} b_j^2 + \sum_{j=1}^{N} b_j^2 \right), \tag{A-4}$$

The coefficient of $u(x_0)$ is the opposite of the sum of all the coefficients other than $u(x_0)$.

Table A-1. Optimization coefficients for the first-order derivative in finite-difference format under different operator lengths.

	4th-order	6th-order	8th-order	10th-order	12th-order
b_0^1	0.0	0.0	0.0	0.0	0.0
b_1^1	0.678803	0.777931	0.841496	0.884147	0.910679
b_2^1	-0.089627	-0.173887	-0.245330	-0.302336	-0.341879
b_3^1		0.023387	0.060819	0.102751	0.138340
b_4^1			-0.008398	-0.026815	-0.048807
b_5^1				0.003981	0.013021
b_6^1					-0.001990

Table A-2. Optimization coefficients for the second-order derivative in finite-difference format under different operator lengths.

	4th-order	6th-order	8th-order	10th-order	12th-order
b_0^1	0.0	0.0	0.0	0.0	0.0
b_1^1	0.678803	0.777931	0.841496	0.884147	0.910679
b_2^1	-0.089627	-0.173887	-0.245330	-0.302336	-0.341879
b_3^1		0.023387	0.060819	0.102751	0.138340
b_4^1			-0.008398	-0.026815	-0.048807
b_5^1				0.003981	0.013021
b_6^1					-0.001990