APPLICATION OF EFFICIENT FREQUENCY-DOMAIN FULL WAVEFORM INVERSION USING TIME-DOMAIN ENCODED SIMULTANEOUS SOURCES

HYUNGGU JUN[1], HYOJOON JIN[2] and CHANGSOO SHIN[1]

[1] *Korea Institute of Ocean Science and Technology, 787, Haean-ro, Snagnok-gu, Ansan, Gyeonggi-do, South Korea.*
[2] *Department of Energy System Engineering, College of Engineering, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, South Korea. wlsgywns@naver.com*

(Received March 19, 2016; revised version accepted December 29, 2016)

ABSTRACT

Jun, H., Jin, H. and Shin, C.S., 2017. Application of efficient frequency-domain full waveform inversion using time-domain encoded simultaneous sources. *Journal of Seismic Exploration*, 26: 141-169.

Full waveform inversion (FWI) is used to determine accurate subsurface velocities through recursive calculation. FWI needs extensive computation; therefore, reducing the computational cost while inverting for an acceptable result is important for the practical application of FWI. Frequency-domain FWI has the advantages of selection of certain frequency components and reduced computational time because of the use of a matrix solver, which solves many sources simultaneously through one matrix factorization. However, the size of the matrix increases exponentially with the size of the computational domain and the number of parameters. The efficiency of frequency-domain FWI decreases in 3D FWI because of limited computational memory. To enhance the efficiency of frequency-domain FWI, time-domain modeling with a simultaneous source was exploited in this study. Although the time-domain modeling scheme is one of the most efficient methods for performing 3D frequency-domain FWI, it still requires time-marching for every source. However, the efficiency can be greatly improved by using the simultaneous source method. Moreover, this method is not limited by the amount of memory required because the time-domain modeling scheme is a matrix-free method. To suppress the crosstalk noise in the simultaneous source method, we use random phase (RP) encoding, random time delay (RTD), and the partial-source assembling method. The nonlinear conjugate gradient method (NLCG) is also used to accelerate the convergence speed. To validate the efficiency of the proposed algorithm, a numerical test is conducted using the 2D SEG/EAGE overthrust model and shows that determining the appropriate balance between the computational cost and the quality of the result can improve the efficiency of the encoded simultaneous source FWI (ESSFWI). The 3D numerical test also verified that the proposed algorithm enhances the computational efficiency and guarantees the quality of the inverted result.

KEY WORDS: time domain, frequency domain, full waveform inversion, encoded simultaneous source.

0963-0651/17/$5.00

INTRODUCTION

Full waveform inversion (FWI) inverts accurate subsurface velocities by reducing the residuals between the modeled and observed wavefields. Many well-known obstacles affect the practical application of FWI, such as its extensive computational costs. The computational cost of the FWI is directly related to the size of the computational domain. Thus, when the domain size is large, such as for 3D FWI, the impedance matrix for the frequency-domain modeling is large, and the use of computing nodes with a large amount of memory is essential. To avoid memory overflow, iterative matrix solvers (Plessix, 2009; Pyun et al., 2011) or a combination of time-domain modeling and the Fourier transform method is used (Sirgue et al., 2008; Kim et al., 2013; Butzer et al., 2013; Xu and McMechan, 2014; Jun et al., 2014). However, the numbers of forward and adjoint modeling calculations do not decrease and are considered to constitute the minimum required computational costs for implementing the FWI. To reduce the computational burden of performing forward and adjoint modeling, the simultaneous source method was proposed (Romero et al., 2000; Krebs et al., 2009; Boonyasiriwat and Schuster, 2010; Ben-Hadj-Ali et al., 2011; Schuster et al., 2011; Schiemenz and Igel, 2013).

The simultaneous source method assumes that the sources are given at the same time. An observed shot gather or modeled shot gather in the simultaneous source method is commonly called a "supershot gather". Several separate observed shot gathers are summed into one shot gather to generate an observed supershot gather, and several sources are simultaneously injected into the model (Krebs et al., 2009) in the modeling procedure to generate a modeled supershot gather. Thus, simultaneous source FWI has dramatically reduced computational costs, with reductions that are approximately equal to the number of shots per supershot. However, crosstalk noise is generated because the forward and adjoint wavefields from the different shots are cross-correlated, during the gradient calculation, and the crosstalk noises degrade the inverted velocity and reduce the convergence speed of simultaneous source FWI. Therefore, several studies have been performed to reduce the crosstalk noise.

Krebs et al. (2009) introduced the random phase (RP) -encoding method to the time-domain encoded simultaneous source FWI (ESSFWI) to suppress the crosstalk noise. They generated encoded shots by randomly multiplying the shot gather by $+1$ or -1 and summed the encoded shots to produce a supershot. The new encoding function was regenerated every iteration to increase the incoherency of the noise among the gradients of each iteration while maintaining the coherency of the signals. Consequently, the signal became large, and the crosstalk noise cancelled out when FWI was repeatedly performed.

Boonyasiriwat and Schuster (2010) and Schiemenz and Igel (2013) performed 3D time-domain ESSFWI using dual randomization methods. The

first randomization was random-phase encoding following Krebs et al. (2009). For the second randomization, they constructed a supershot using several sources selected from random locations. The source-encoding function was regenerated every iteration, and the values of the source-encoding function were randomly selected as -1, 0, or $+1$. This process made both the phase and the source location random. This dual randomization method gave a superior result than that of the single randomization (phase-encoding) method. However, the speedup factor was lower than that of the method of Krebs et al. (2009) because several supershots were used simultaneously.

Ben-Hadj-Ali et al. (2011) applied the simultaneous source method to frequency-domain FWI. For the frequency-domain phase encoding, they multiplied the modeled and observed wavefields by a complex term. The encoding function could be regenerated at every frequency group, every iteration, or every frequency in frequency-domain FWI, and the numerical examples revealed that regenerating the encoding function at every frequency gave the best result. Moreover, the partial-source assembling method was better at suppressing the crosstalk noises than the full-source assembling method; however, the partial-source assembling method incurred a higher computational cost. Two different types of subsource-selecting methods were tested, and the coarse selection method gave a better result than the close selection method. This result is in good agreement with that of Schiemenz and Igel (2013).

Yang et al. (2013) applied a time window to frequency-domain ESSFWI. The modeling was performed in the time domain to apply the time window, and the inversion processes were accomplished in the frequency domain. The use of time-domain modeling for implementing frequency-domain ESSFWI has two advantages: First, the time window can be flexibly applied, which reduces the local minima and enhances the FWI result (Yang et al., 2013). Second, the computation is relatively efficient. The direct matrix solver, which can solve several sources with one factorization, is widely used for frequency-domain FWI. It is an efficient method because the modeling operator is independent of the source location. However, time-domain modeling or the iterative matrix solver is more efficient than the direct matrix solver for frequency-domain ESSFWI because the simultaneous source technique cannot reduce the computational time and memory required for matrix factorization of the direct matrix solver.

Based on these studies, we demonstrate how to apply frequency-domain FWI using time-domain simultaneous sources with several methods for reducing the crosstalk. For the simultaneous source FWI, the supershots are computed in the time domain and the residual, gradient and Hessian are calculated in the frequency domain. To reduce crosstalk noise, four methods are used. The first method is the random encoding method. The second method is partial-source assembling (Ben-Hadj-Ali et al., 2011), and the third method is random

source-location (Boonyasiriwat and Schuster, 2010). Moreover, the nonlinear conjugate gradient method (NLCG) (Fletcher and Reeves, 1964) is used to enhance the convergence speed. The relationship among the number of supershots, computational cost, and quality of the inverted velocity is analyzed to identify the best combination for ESSFWI.

This paper introduces a brief algorithm for frequency-domain FWI using time-domain modeling. Next, four methods for reducing the crosstalk noise are explained. To compare the effect of each method and validate the efficiency of the proposed algorithm, numerical tests using the 2D SEG/EAGE overthrust model are performed. This algorithm is also applied to the 3D SEG/EAGE overthrust model to show that the proposed algorithm can effectively reduce the computational costs while generating acceptable inverted velocities. In discussion, we summarize the application methods of ESSFWI and analyze the results of the numerical tests.

THEORY

Review of frequency-domain FWI using time-domain modeling

Sirgue et al. (2008) used a time-domain finite-different method and discrete Fourier transform for frequency-domain FWI to enhance the efficiency of frequency-domain FWI. This study also uses the time-domain modeling method for the efficient frequency-domain FWI. The objective function of frequency-domain FWI is expressed as follows:

$$E(\mathbf{m}) = \sum_{i=1}^{n_S} \tfrac{1}{2}(\tilde{\mathbf{u}}_i - \tilde{\mathbf{d}}_i)^T(\tilde{\mathbf{u}}_i - \tilde{\mathbf{d}}_i)^* \quad , \tag{1}$$

where n_S is the number of sources, $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{d}}$ are the modeled and observed wavefield vectors in the frequency domain, respectively. T indicates transpose, and * denotes a complex conjugate. To obtain the modeled data $\tilde{\mathbf{u}}$ in the frequency domain, forward modeling is performed in the time domain, and the time-domain wavefields are transformed to the frequency domain via discrete Fourier transform. To find the model parameter $\mathbf{m}$ that minimizes the objective function $E(\mathbf{m})$, the local optimization method is used, and the model update $\Delta\mathbf{m}$ is obtained as follows:

$$\Delta\mathbf{m} = -\mathbf{H}^{-1}\nabla E(\mathbf{m}) \approx -Re[diag(\mathbf{J}^T\mathbf{J}^* + \lambda\mathbf{I})]^{-1}Re(\mathbf{J}^T\mathbf{r}^*) \quad , \tag{2}$$

where $\mathbf{H}$ is the Hessian matrix, $\mathbf{J}$ is the partial derivative wavefield matrix $\partial\tilde{\mathbf{u}}/\partial\mathbf{m}$, $\mathbf{r}$ is the residual matrix, λ is the stabilizing constant (Marquardt, 1963),

diag indicates diagonal elements and *Re* denotes the real part. To effectively calculate the Hessian matrix, we assumed the multiplication of the receiver side Green's function as the identity matrix and used only the diagonal elements (Shin et al., 2001). The gradient of the objective function is calculated using the adjoint state method (Plessix, 2006). The adjoint wavefields are calculated in the time domain using the adjoint wave equation, and the wavefields are transformed to the frequency-domain adjoint wavefields via discrete Fourier transform. The model parameter **m** of the k-th iteration is updated as follows:

$$\mathbf{m}^k = \mathbf{m}^{k-1} - \alpha \times \Delta \mathbf{m} \quad , \tag{3}$$

where α is the step-length calculated from parabolic interpolation (Vigh et al., 2009).

Simultaneous source encoding in the time domain

The simultaneous source encoding method reduces the computational cost of FWI, but crosstalk noise still remains as a deficiency. To reduce crosstalk noise and enhance the efficiency of ESSFWI, four well known approaches are applied individually and simultaneously.

First, we apply two different source encoding methods individually and simultaneously to generate a time domain supershot. One method is random phase encoding (RP) suggested by Krebs et al. (2009), and the other is random time delay (RTD) suggested by Berkhout and Blacquière (2011). To obtain the optimal result by considering the computational cost and crosstalk noise, we applied partial-source assembling, which divides the sources into several supershots and uses several supershots simultaneously (Ben-Hadj-Ali et al. 2011). The computational cost for the partial source assembling is shown in the Appendix. In the partial-source assembling, the method of constructing the supershots also influences the crosstalk of ESSFWI. We introduce 3 different methods of selecting the source location: regular interval (Fig. 1(a)], random [Fig. 1(b)], and random-in-subgroup [Fig. 1(c)]. The regular interval method divides the sources into several groups and then selects a source per group with same interval. The random method selects the sources randomly from all sources. The random-in-subgroup method also divides the sources into several groups, similar to the regular interval method, but randomly selects a source in a group. To reduce the crosstalk noise of the model update and enhance the convergence speed of ESSFWI, the NLCG (Fletcher and Reeve 1964) is also used.

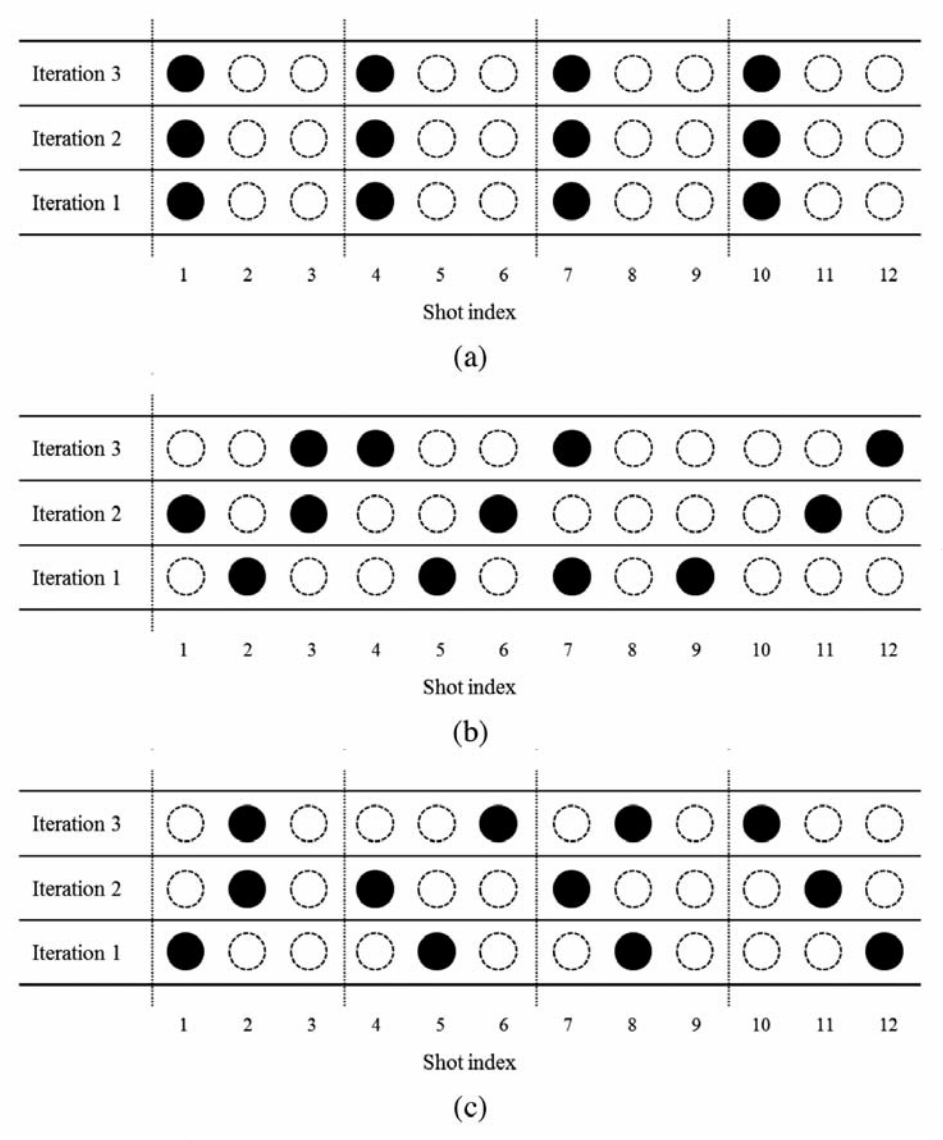


Fig. 1. The subsource locations of the first supershot using (a) the regular interval method, (b) the random method, and (c) the random-in-subgroup method. The black circles with solid lines are active sources, and the white circles with dashed lines are inactive sources.

NUMERICAL EXAMPLES

Preparation for ESSFWI

The numerical tests were performed using the 2D SEG/EAGE overthrust model [Fig. 2(a)]. We conducted forward modeling using the 2D time domain

finite-difference method to generate the time-domain observed data. The recording time was 8 s with an interval of 2 ms. In total, 200 sources were simulated with an interval of 25 m, and 801 receivers per source were used with an interval of 25 m. The starting model [Fig. 2(b)] is generated by applying the Gaussian filter to the true model with a window length of 500 m. Following Ban-Hadj-Ali et al. (2011), the true model is set in the first 100 m of the starting model to avoid instabilities in the weathered layer near the surface and accelerate the speed of the numerical test. We used 15 frequencies simultaneously ranging from 3.0 Hz to 10.0 Hz with an interval of 0.5 Hz. The stabilizing constant λ in eq. (2) is empirically chosen as the Hessian $\times$ 1.0e$-$3.

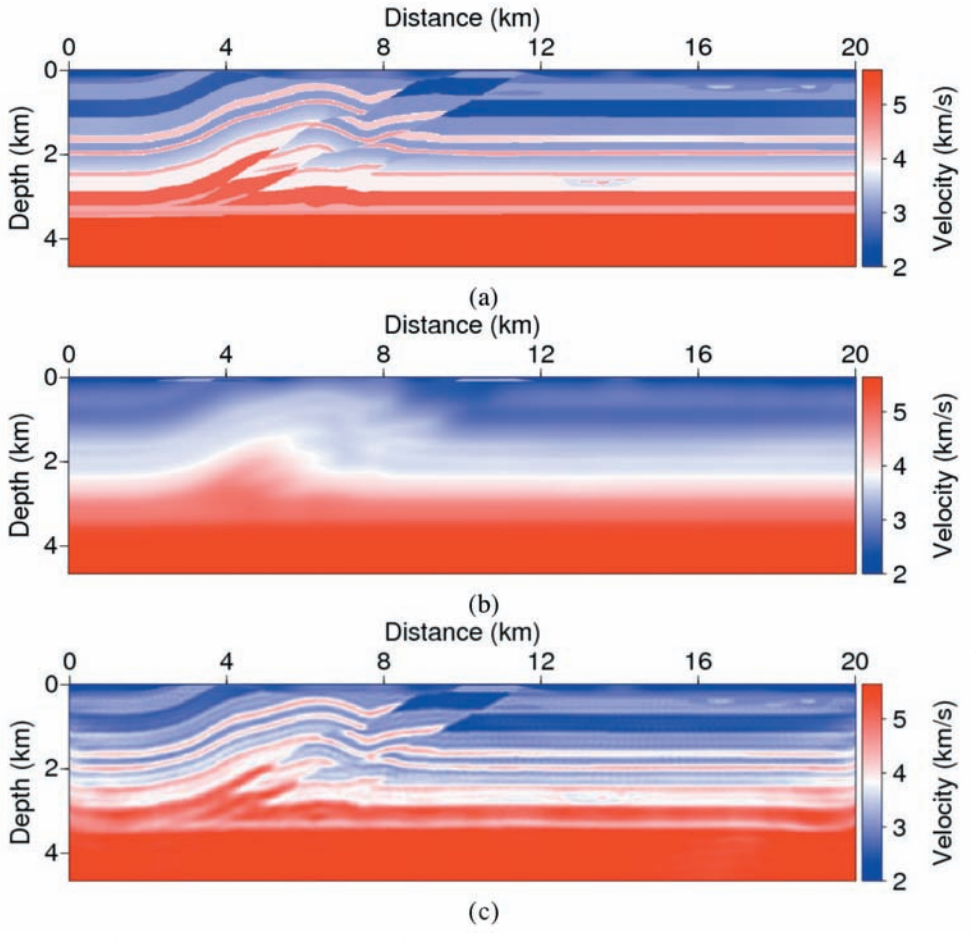


Fig. 2. (a) The true velocity model, (b) the starting velocity model and (c) the reference velocity model.

Fig. 2(c) is the reference velocity model, which is obtained using conventional frequency-domain FWI after 75 iterations. The computational cost of the reference velocity model is computed using eq. (A-2). $C_{forward}$ and $C_{adjoint}$ are assumed to be 1 because they are the same for all methods, and C_{etc} is assumed to be 0 because only a minor computational effort is required. Therefore, C_{conv} is 60,000.

Application of ESSFWI

In this section, we introduce the detailed methods of applying ESSFWI. We apply the methods proposed above and compare the results to find the most appropriate method that balances crosstalk noise and computational cost. First, we compare two types of source encoding methods. Then, we analyze how the NLCG influences crosstalk noise and ESSFWI convergence. Use of the source encoding method alone cannot properly attenuate crosstalk noise; thus we apply the partial-source assembling method and compare the results. Because the number of supershots to be used simultaneously influences both crosstalk noise and computational cost, we perform the numerical test by changing the number of supershots and compare the results to decide the optimal balance between crosstalk noise and computational cost. Last, we also perform tests using 3 different methods of selecting the sub-sources for a supershot; because the method of selecting the sub-sources affects the ESSFWI result. Through the numerical tests, we decide the most appropriate method for ESSFWI and apply the method to the 3D synthetic data.

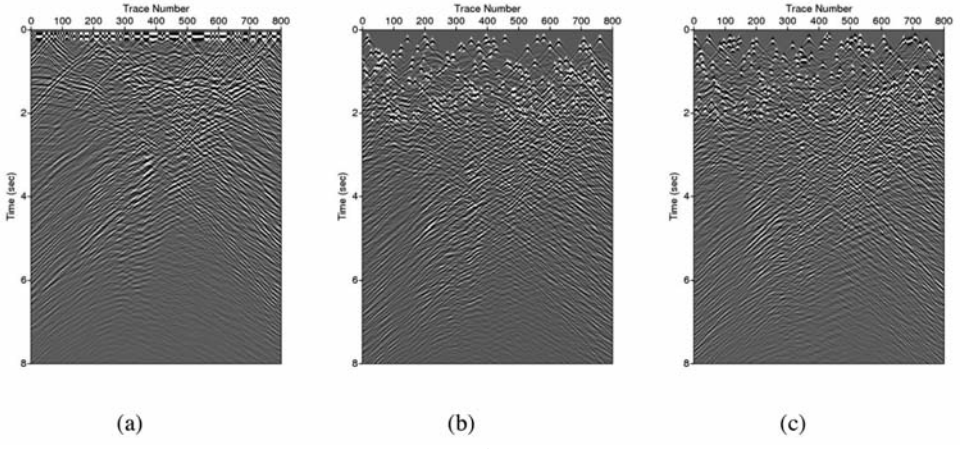


Fig. 3. (a) RP, (b) RTD and (c) RP&RTD encoded ESS observed data.

Comparison of RP and RTD encoding methods

For efficient frequency-domain ESSFWI, we generated the ESS in the time domain and transformed it to the frequency domain. The RP and RTD encoding methods were used individually and simultaneously to generate the ESS in the time domain. The sources were encoded into one supershot (full-source assembling), and the encoding function was regenerated at each iteration. Fig. 3(a) shows the observed RP ESS data. The data do not have a time delay because all sources are assumed to be simulated simultaneously, and the sign of the phase changes randomly between "+" and "−". Fig. 3(b) gives the observed RTD ESS data. The phase of the data does not change; indeed, only the source simulated time changes randomly. Fig. 3(c) presents the observed RP&RTD ESS data. Here, the sources are assumed to be simulated with RP at random times, and thus, both the phase and source simulated times change randomly.

Fig. 4 is the inverted velocities after 100 iterations of ESSFWI using the RP, RTD and RP&RTD encoding methods. For the RTD and RP&RTD methods, we used 2 s as the maximum time delay, following Anagaw and Sacchi (2014). The velocities are properly inverted with a certain level of crosstalk noise, but as a result, the velocities are not sufficiently converged. The inverted velocity obtained using RP encoding was the poorest and the other two results show similar velocities. To quantitatively compare the three encoding methods, we plotted the normalized data misfit [Fig. 5(a)] and the model misfit [Fig. 5(b)]. The model misfit was calculated by following Brossier et al. (2009) and Jun et al. (2015):

$$\text{model misfit} = (1/n)\left\| (\mathbf{m}_{\text{inv}} - \mathbf{m}_{\text{true}})/\mathbf{m}_{\text{true}} \right\|_2 , \tag{4}$$

where n is the number of grids in the model, $\mathbf{m}_{\text{inv}}$ is the inverted velocity and $\mathbf{m}_{\text{true}}$ is the true velocity. Among the data misfit results, the RP method gives the highest misfit, while the RTD and RP&RTD methods give slightly lower misfits. Among the model misfit results, the RP method also gives the highest misfit, and the RTD method gives the lowest misfit. The RP&RTD method results in a slightly higher misfit than the RTD method. In this test, the RTD method's inverted velocity was much better than those of the other methods. Therefore, we used the RTD method for the following tests.

The computational costs (C_{ess}) of the three methods was 400, as determined using eq. (A-1), and the speedup according to eq. (A-3) was 150 relative to the computational costs of obtaining the reference velocity model. In other words, ESSFWI requires only 1/150-th of the computational costs incurred by conventional FWI. The inverted velocity obtained using ESSFWI was not as high quality as that of the reference velocity model because of the crosstalk noise and slow update speed; however, computational cost was greatly reduced.

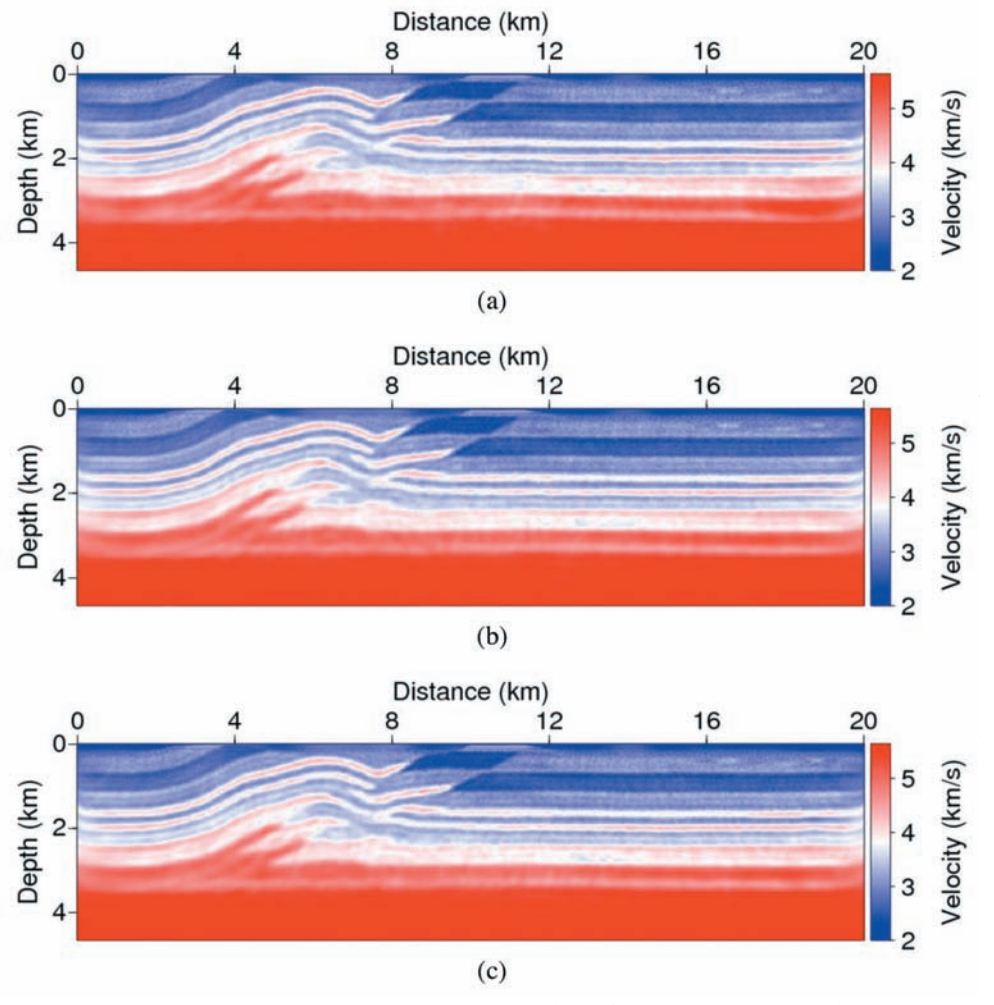
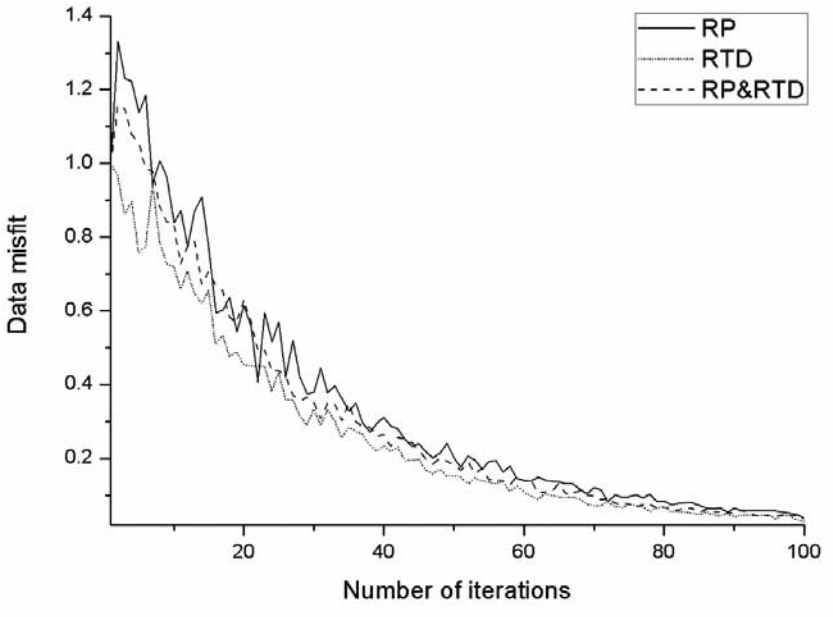


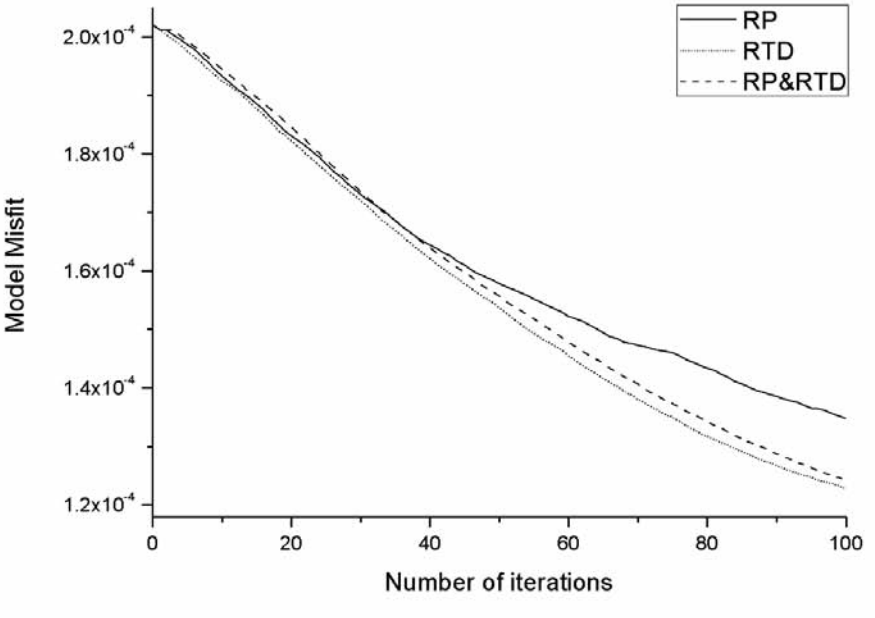
Fig. 4. The inverted velocities of ESSFWI using (a) RP, (b) RTD and (c) RP&RTD.

ESSFWI using NLCG

Crosstalk noise reduces the convergence speed of ESSFWI. The NLCG helps the model update by calculating the correct direction according to the sum of the previous model updates. The crosstalk noise of the summed model update is thus reduced, and the signals become stronger; therefore, NLCG is an appropriate method to reduce crosstalk noise and enhance the convergence speed of ESSFWI. We used the Fletcher and Reeve NLCG method (Fletcher and Reeve, 1964) in this work.



(a)



(b)

Fig. 5. (a) Data misfit and (b) model misfit of ESSFWI using RP, RTD and RP&RTD.

Fig. 6 shows the 30-th model updates of ESSFWI with and without NLCG. Both model updates are contaminated by crosstalk noise, but the SNR (signal to noise ratio) is higher in the model update with NLCG than in that without NLCG.

To show the effect of applying NLCG to ESSFWI quantitatively, we calculated the SNR of the model updates. Eq. (5) calculates the SNR of the model updates based on the number of iterations. We obtained eq. (5) by modifying the equation calculating the SNR of the gradient direction suggested by Jeong et al. (2013). The SNR is calculated as follows:

$$\mathrm{SNR} = \| \Delta \mathbf{m} \| / \| \mathrm{NRM}(\sum_{i=1}^{N_{iter}} \Delta \mathbf{m}_i) - \Delta \mathbf{m}_{ref} \| \quad , \tag{5}$$

with

$$\Delta \mathbf{m}_{ref} = \mathrm{NRM}(\mathbf{m}_{start} - \mathbf{m}_{inv}) \quad , \tag{6}$$

where $\Delta \mathbf{m}_{ref}$ is the reference model update, $\Delta \mathbf{m}_i$ is the model update at the i-th iteration, $\mathbf{m}_{start}$ is the starting velocity, $\mathbf{m}_{inv}$ is the inverted velocity obtained without using source encoding [Fig. 2(c)], N_{iter} is the number of iterations and

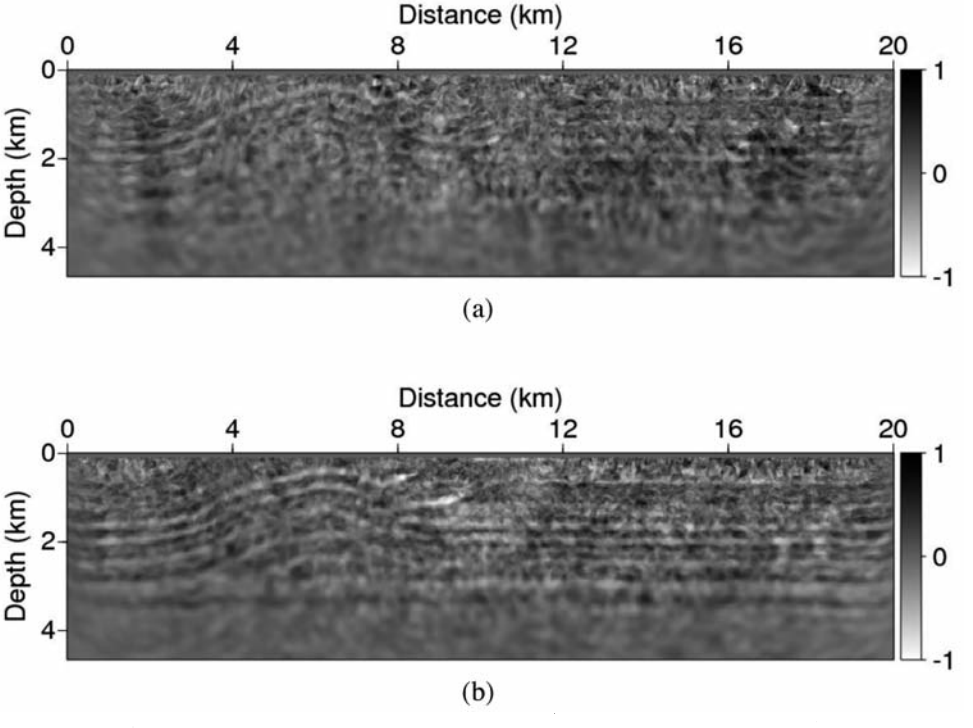


Fig. 6. The 30th model updates of RTD ESSFWI (a) without NLCG and (b) with NLCG.

NRM is a normalization operator. The numerator of eq. (5) signifies the model update of the ideal case, which does not include crosstalk noise; the 1st term of the denominator of eq. (5) indicates the summed model update with crosstalk noise. We applied the normalization operator to compare summed and reference model update. Therefore, the denominator of eq. (5) indicates the crosstalk noise, and eq. (5) represents the SNR of the model update. Fig. 7 shows the SNRs of the model updates for each iteration with and without NLCG. After multiple iterations, the SNR of the model update with NLCG becomes much higher than the SNR without NLCG. This result indicates that applying NLCG to ESSFWI helps reduce the crosstalk noise, which results in accelerating the convergence speed.

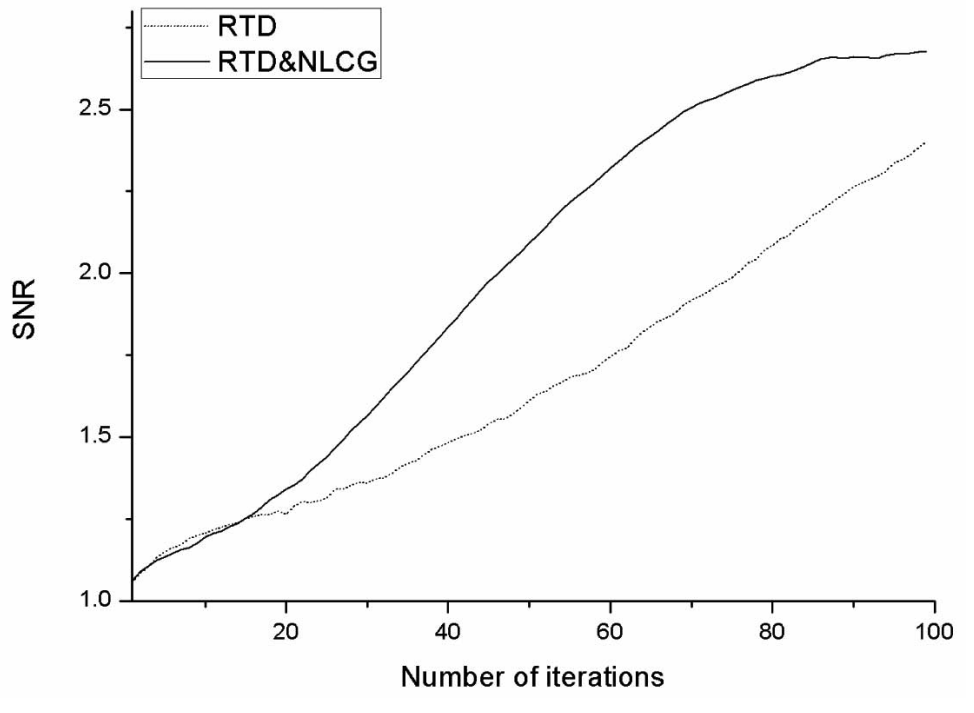


Fig. 7. SNR of the model update with and without NLCG.

Fig. 8 shows the inverted velocity after 100 iterations of ESSFWI with NLCG. The inverted velocity with NLCG is similar to the true velocity model, whereas that without NLCG [Fig. 4(b)] was updated slowly because of crosstalk noise. To quantitatively compare the results, we plotted the data misfit and model domain misfit results in Fig. 9. The data misfit and model misfit obtained using NLCG show better convergence and lower errors than those without NLCG. The computational cost of applying NLCG is minor, and thus, using NLCG enhances the efficiency of ESSFWI.

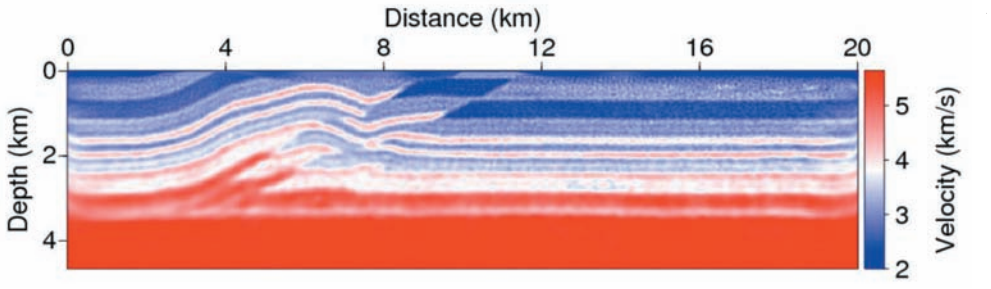
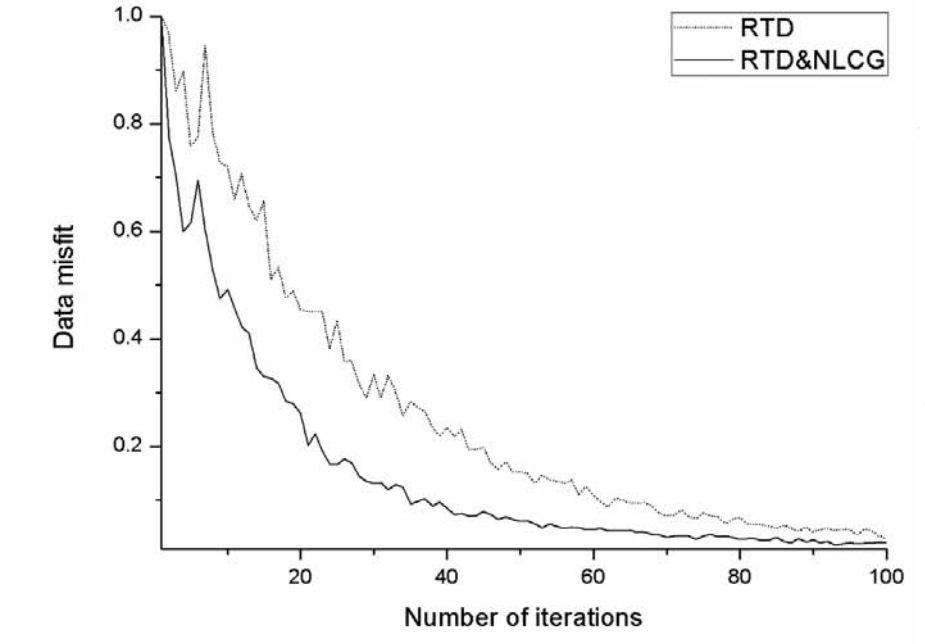


Fig. 8. The inverted velocities obtained by RTD ESSFWI with NLCG.

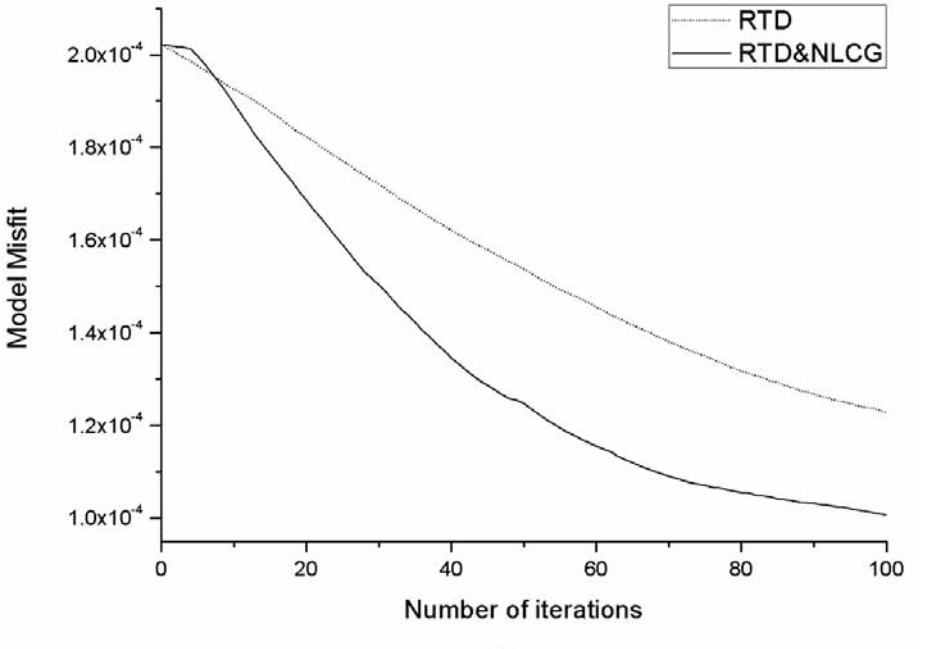
Partial-source assembling for ESSFWI

ESSFWI achieves the highest speedup when all sources are encoded into single supershot (full-source assembling). However, the crosstalk noise increases proportionally to the square root of the number of sources in the supershot (Schuster et al., 2011). If the level of crosstalk noise is increased, the number of iterations can be increased to achieve sufficient misfit convergence because the crosstalk noise slows the convergence speed of ESSFWI and, thus, reduces the speedup. Moreover, full-source assembling cannot sufficiently converge because of the existence of crosstalk noise. Therefore, using several supershots simultaneously (partial-source assembling) is a better method for ESSFWI. However, if many supershots are used simultaneously, the crosstalk noises can be significantly decreased, whereas the computational cost will increase. The purpose of ESSFWI is to reduce the computational costs associated with FWI while maintaining acceptable velocity inversion; thus, it is important to identify the appropriate balance between computational costs and crosstalk noise. For the numerical tests, we performed ESSFWI with full-source and partial-source assembling by increasing the number of supershots from 1 to 16. The crosstalk noise and convergence resulting from each assembling method are compared when the same computational cost (C_{ess}) was used. The computational costs were computed using eq. (A-1). For frequency-domain ESSFWI, we used RTD and NLCG simultaneously, and for the partial-source assembly, we used the regular interval [Fig. 1(a)] method.

Fig. 10 shows the ESSFWI results achieved with full-source assembling (1 supershot) and partial-source assembling (2, 4, 8 and 16 supershots) until became 1600; in this case, the speedup was 37.5 relative to the computational cost of generating the reference velocity model [Fig. 2(c)]. The number of iterations of each method is shown in Table 1. The crosstalk noise was most severe in the inverted velocity obtained with full-source assembling [Fig. 10(a)].



(a)



(b)

Fig. 9. (a) Data misfit and (b) model misfit with and without NLCG.

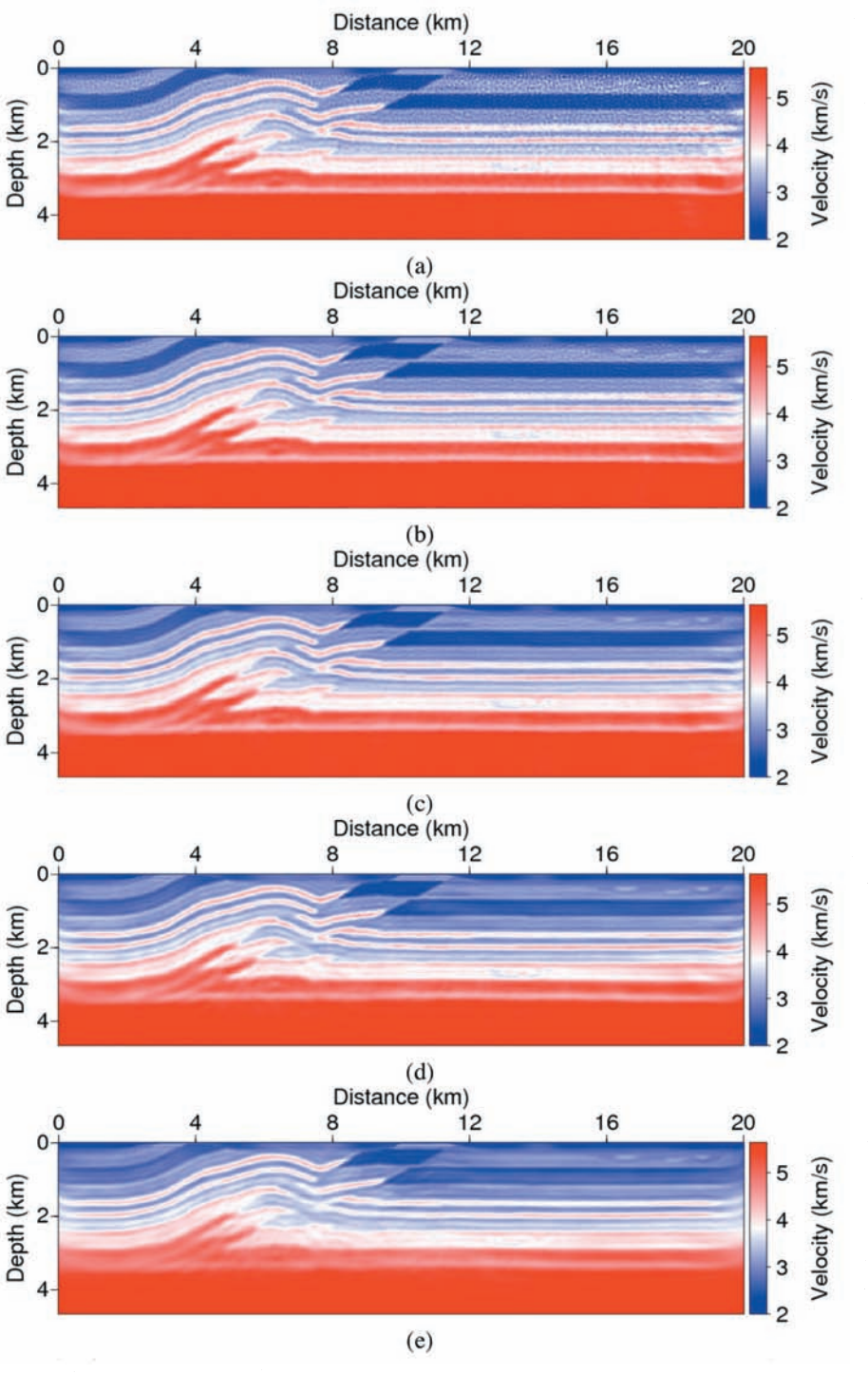


Fig. 10. The inverted velocities obtained using (a) full-source assembling and partial-source assembling with (b) 2, (c) 4, (d) 8 and (e) 16 supershots at a computational cost of 1600.

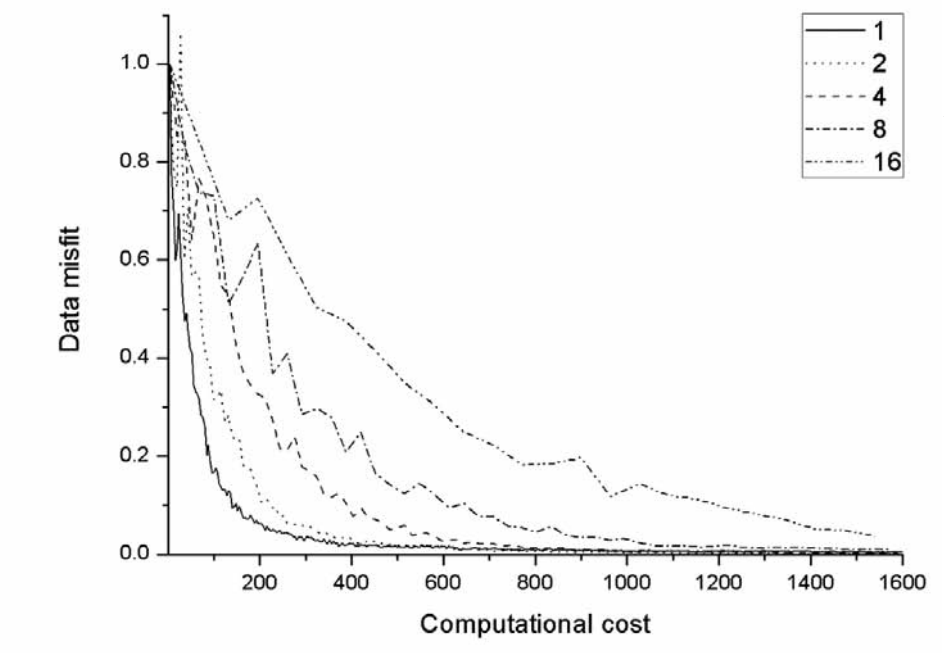
Fig. 10(b) presents the inverted velocity obtained using 2 supershots simultaneously, in which the crosstalk noise was more suppressed than that obtained using 1 supershot. The inverted velocity obtained using partial-source assembling with 4 supershots [Fig. 10(c)] converged to the true model better than other, previous results. Crosstalk noise was rarely observed, and most of the structures were clearly inverted. Fig. 10(d) presents the ESSFWI result determined using 8 supershots. No crosstalk noise was observed, but the velocity did not sufficiently converge to the true model. Using 16 supershots [Fig. 10(e)] simultaneously for ESSFWI, we did not observe any crosstalk noise, but the velocity was less updated than the previous results.

We compared the normalized data misfits and model misfits of the each assembling method at the same computational cost (Fig. 11). When the computational cost reaches 1600, the data misfits exhibited similar levels of convergence when 1 to 8 supershots were used; however, the partial-source assembling method using 16 supershots required more iterations to achieve improved convergence. The model misfit using 1 supershot converges rapidly at the early stage of ESSFWI, but the misfit increases during the later stages because the crosstalk noises are not properly suppressed. The model misfits of the ESSFWI using 2 and 4 supershots decreased rapidly and converged well. The model misfits of the ESSFWI using 8 and 16 supershots also decreased well but did not sufficiently converge.

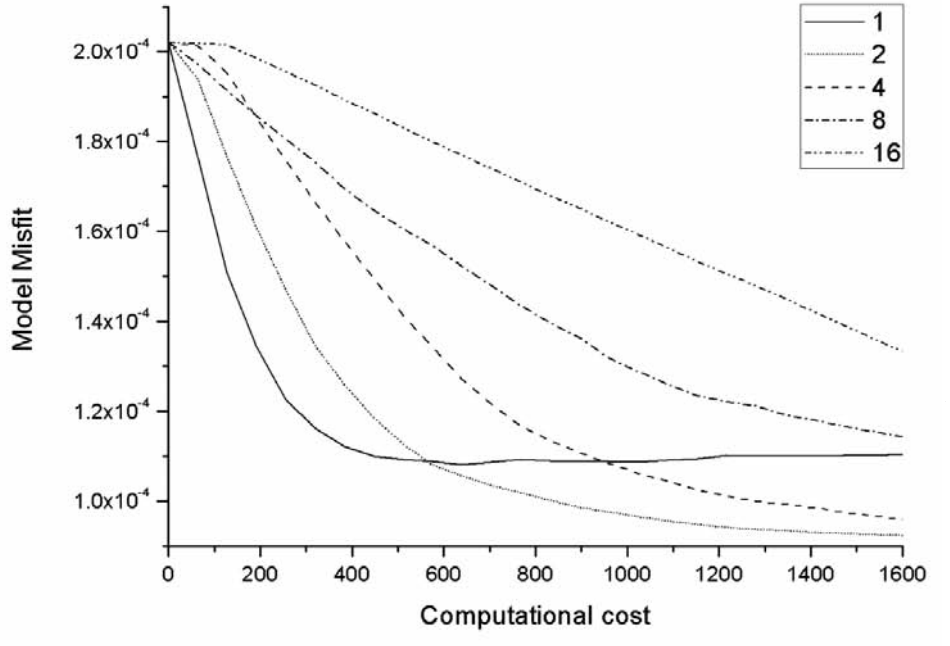
Comparison of the ESS grouping methods

The previous tests verified that partial-source assembling enhances the efficiency of ESSFWI. Now, we address how to assemble the sources. We used three assembling methods to perform ESSFWI and compared their results. For ESSFWI, partial-source assembling using 4 supershots, RTD, and NLCG methods was implemented. The number of iterations was 100, and C_{ess} was 1600.

Fig. 12 shows the inverted velocities obtained using 2 different source-assembling methods: random [Fig. 12(a)] and random-in-subgroup [Fig. 12(b)]. The result using regular interval method is in Fig. 10(c). All the methods suppressed the crosstalk noise appropriately, and the results show only minor differences. To compare the results, we plotted the data misfits and model misfits in Fig. 13. At the early stage of ESSFWI, the regular interval method converged faster than the other methods in both the data and model misfits. In the later stage of ESSFWI, each method converges differently. The random method gave the lowest model misfit, and the regular interval showed the highest model misfit. Schiemenz and Igel (2013) reported that the regular interval method gave a slightly better result than the random method, in contrast



(a)



(b)

Fig. 11. (a) Data misfits and (b) model misfits using full-source and partial-source assembling.

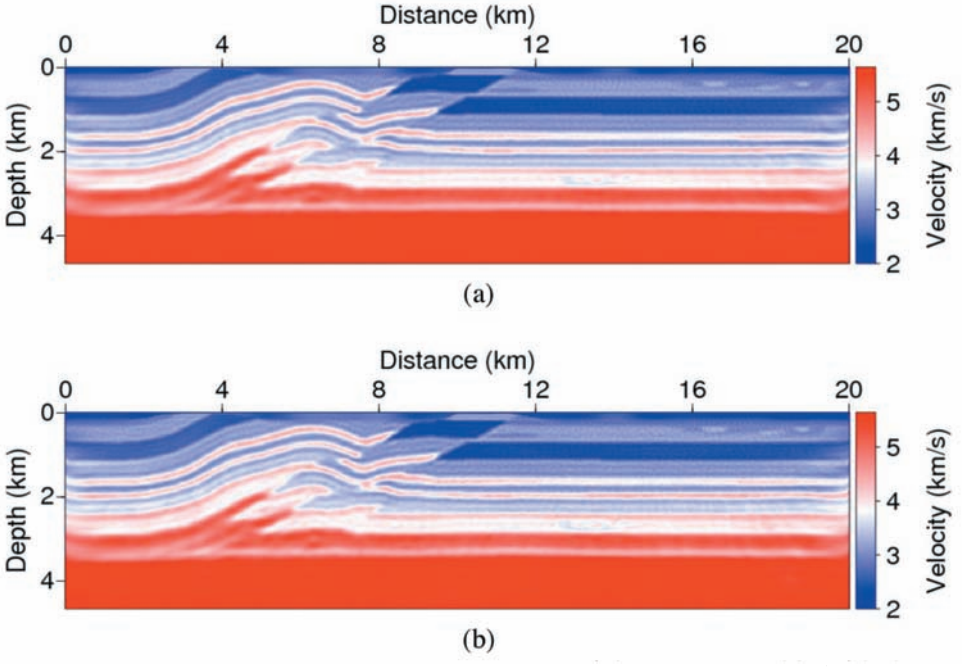
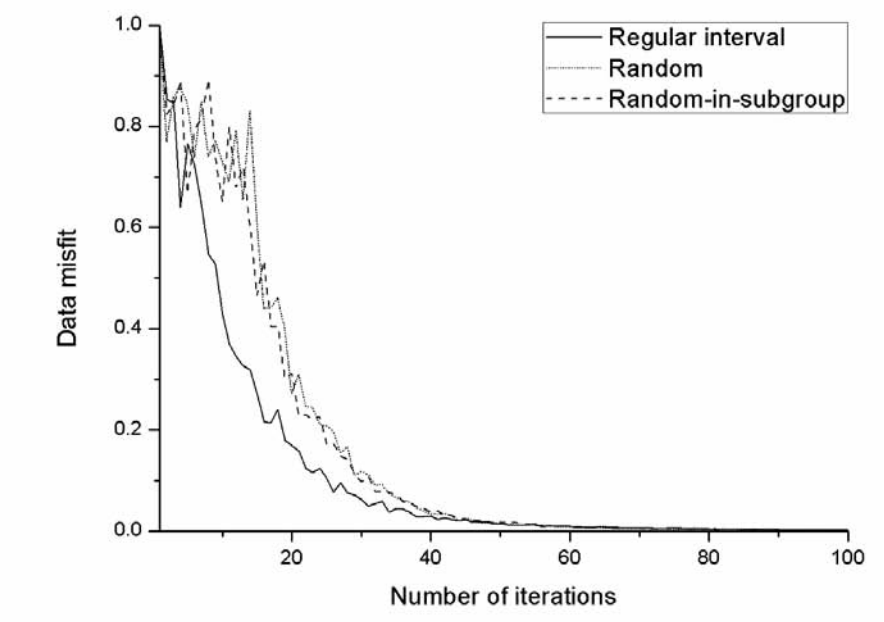


Fig. 12. The inverted velocities obtained by applying partial-source assembly with the (a) random and (b) random-in-subgroup methods.

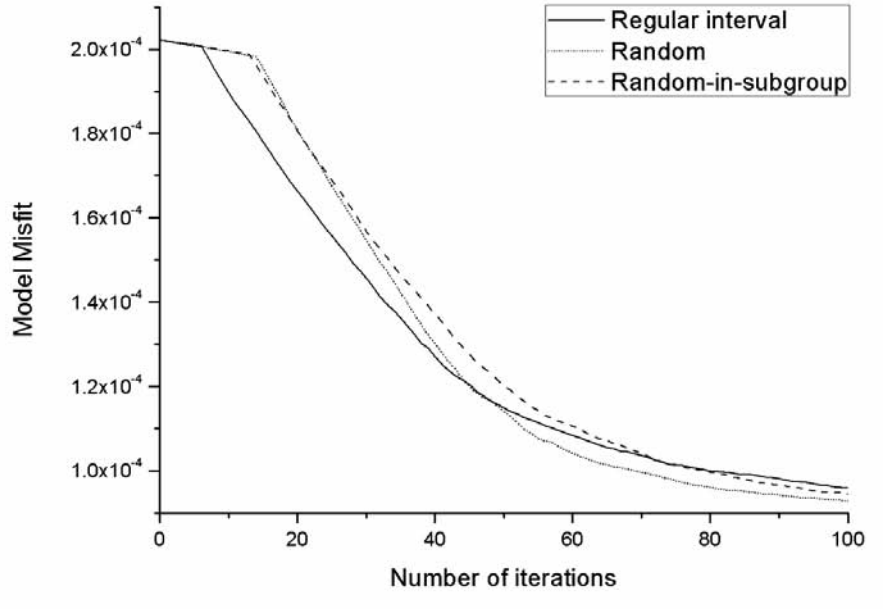
to the result of this study. However, they only performed 15 iterations of ESSFWI, and thus, the velocity may not have sufficiently converged. This study also shows that the misfits produced by the regular interval method are lower than those of other methods in the early stage of ESSFWI, in agreement with the result of Schiemenz and Igel (2013).

3D ESSFWI case study

We applied ESSFWI to the 3D SEG/EAGE overthrust model. First, we resized the original 3D SEG/EAGE overthrust model and selected a certain part of the model for the numerical test. The size of the selected model was 251 × 51 × 65 (x, y, z), and the grid spacing was 25 m. The observed data were generated using 3D time-domain finite-difference method. The recording time was 4 s, with an interval of 2 ms. The acquisition involved 50 × 10 (500) sources with intervals of 125 m × 125 m and 251 × 10 (2510) receivers with intervals of 25 m × 125 m. The starting model was generated by applying the Gaussian filter to the true model with a window length of 375 m. The true and starting models are shown in Figs. 14(a) and 14(b).

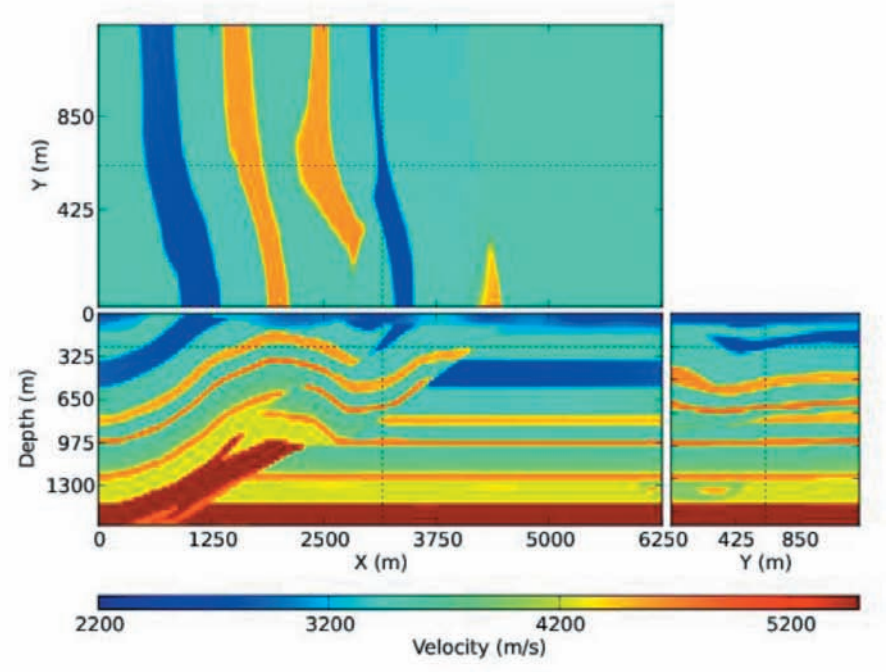


(a)

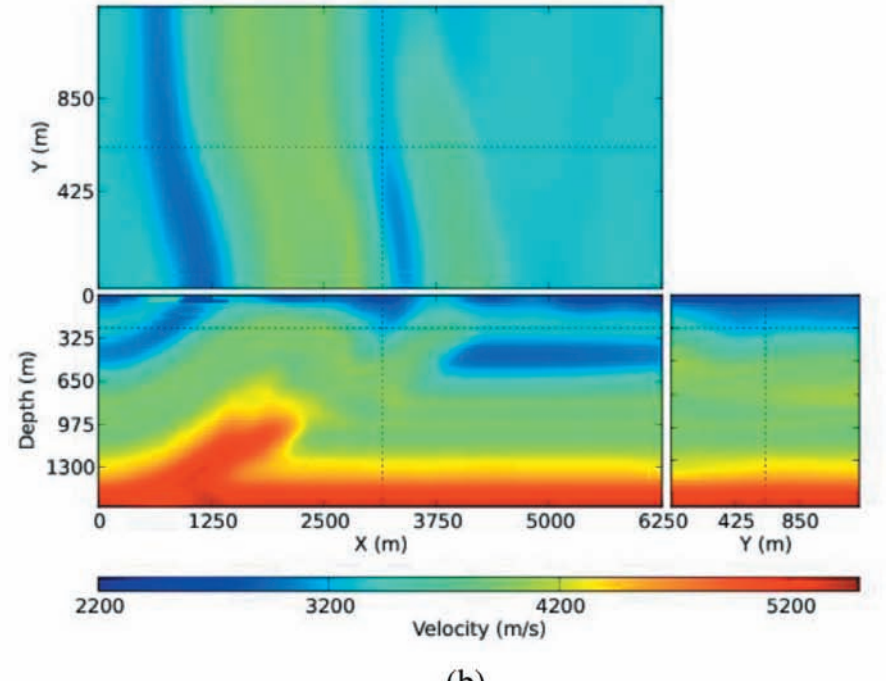


(b)

Fig. 13. (a) Data misfits and (b) model misfits using the regular interval, random and random-in subgroup methods.

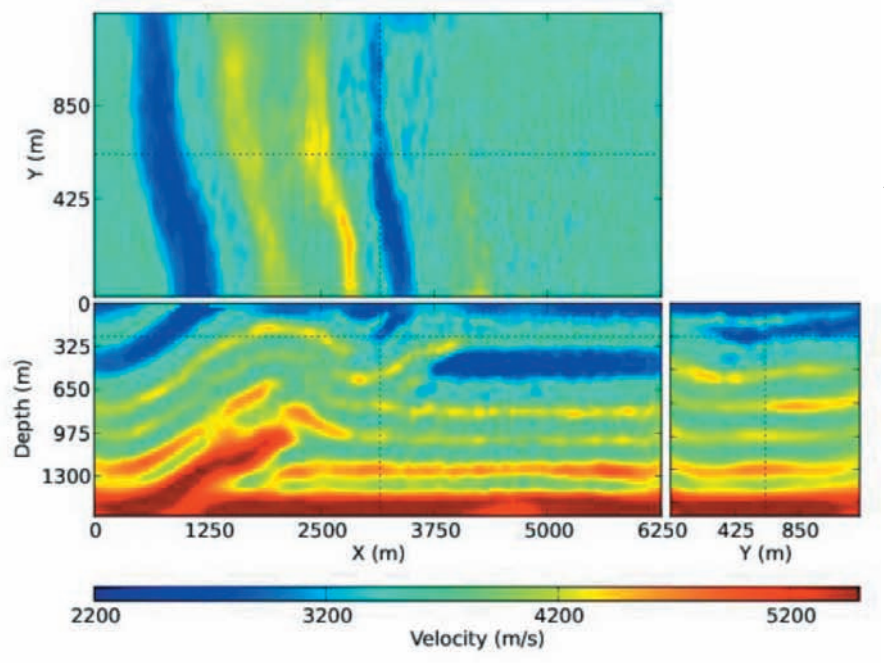


(a)

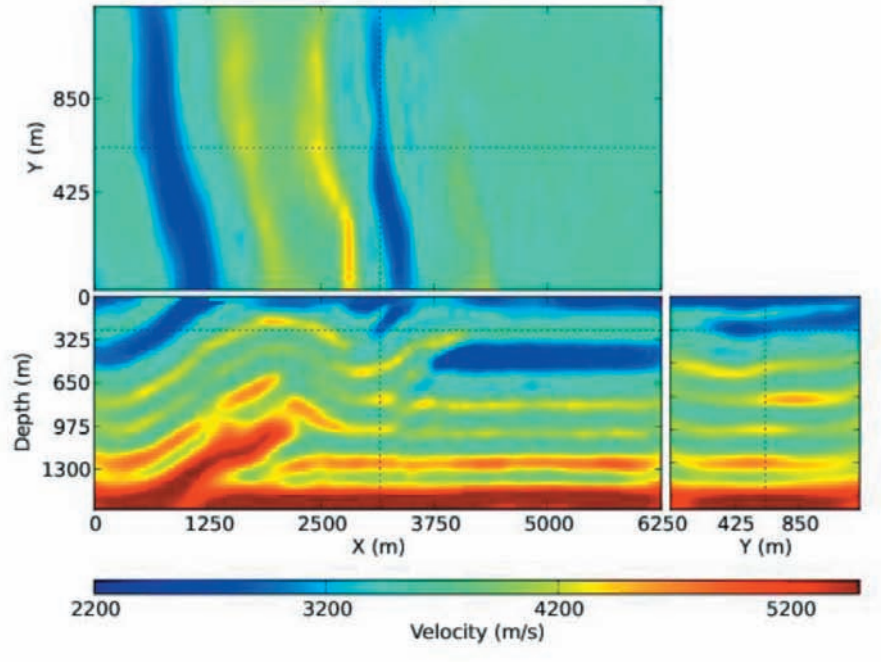


(b)

Fig. 14. (a) True velocity model and (b) starting velocity model



(a)



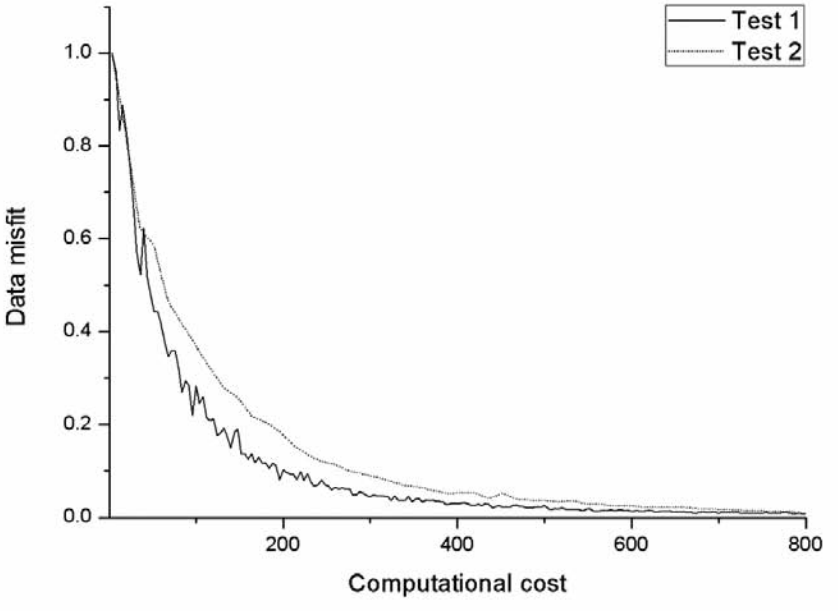
(b)

Fig. 15. The inverted velocities produced in (a) the first test and (b) the second test.

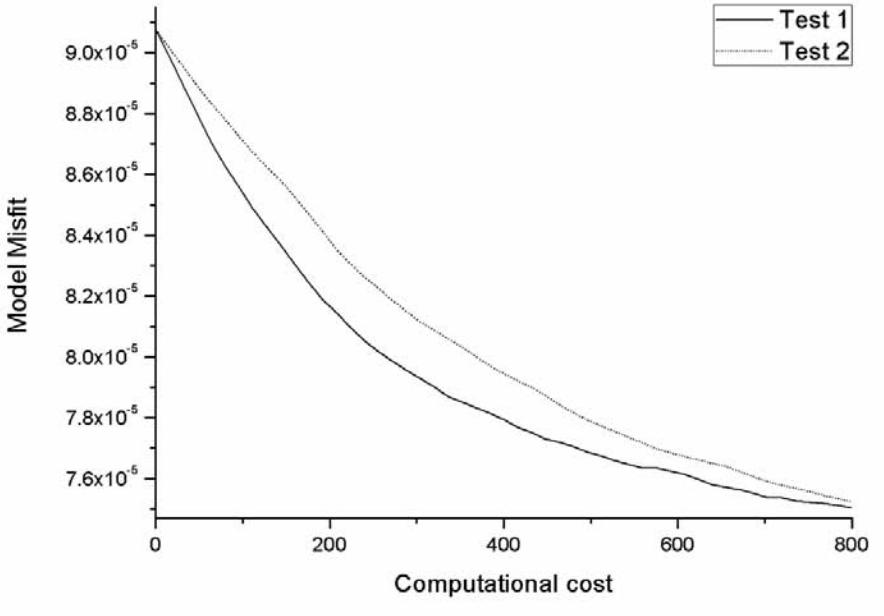
We performed 3D ESSFWI tests via 2 different methods: one of the least- and the most-efficient methods identified by the 2D ESSFWI tests. The first test was performed using the RTD method and full-source assembling, and the second test was performed using the RTD method, partial-source assembling with 4 supershots, random source selection and NLCG. The maximum time-delay was 2 s. The frequencies used for the numerical tests were 4.0 to 10.0 Hz, with an interval of 0.5 Hz. We performed ESSFWI until C_{ess} reached 800, which corresponds to 200 iterations for the first test and 50 iterations for the second test. The inverted velocities are given in Fig. 15. Fig. 15(a) shows the inverted velocity obtained in the first test. The structures are properly inverted but underestimated, and the crosstalk noise is severe, especially in the shallow part of the model. Fig. 15(b) shows the inverted velocity found in the second test. Although the velocity was updated in only 50 iterations, it was updated to a similar level as the first test, with a low level of crosstalk noise. We plotted the data and model misfits in Fig. 16 for comparison. Both misfits converged to similar values at the later stage of ESSFWI. Additionally, the model misfits show that ESSFWI did not fully converge, and when more iterations were performed, the model misfit of the second test was lower than that of the first test. These findings demonstrate that the proposed ESSFWI methods can be applied to 3D data and give results that are in good agreement with the numerical tests of the 2D SEG/EAGE overthrust model.

DISCUSSION

We have applied four methods to reduce the crosstalk noise and enhance the efficiency of frequency domain ESSFWI. The first method was random source encoding. ESSFWI using the source encoding method increases the randomness of FWI by regenerating the encoding function every iteration to alleviate crosstalk noise. In the numerical test, we compared two different types of encoding methods, RP and RTD, and applied them individually and simultaneously to make the time-domain simultaneous sources; then, we compared the inverted velocities. The data and model misfits of the inverted result using the RP method are higher than with other methods, whereas misfits using the RTD method are lower than with other methods. The randomness of the encoding method is highest in the RP&RTD method, but the misfits did not reach the lowest level because the crosstalk was not sufficiently reduced, despite the high randomness. These results indicate that high randomness in the encoding method does not always guarantee the best result. Second, we applied NLCG to ESSFWI. Applying NLCG makes the model update of ESSFWI less contaminated by crosstalk noise because it uses several previous model updates to construct the current model update. When we apply NLCG, the SNR of the model update is higher and the data and model misfits are lower than those obtained without NLCG. These results indicate that applying NLCG to ESSFWI enhances the inverted results.



(a)



(b)

Fig. 16. (a) Data misfits and (b) model misfits of the first and second tests.

To balance crosstalk noise and computational cost, we used partial-source assembling. We compared the quality of inverted velocities and computational costs by increasing the number of simultaneously-used supershots from 1 to 16. Based on the numerical tests, we can conclude that the use of many supershots simultaneously successfully suppresses crosstalk noise but reduces the computational efficiency and that using fewer supershots simultaneously does not properly suppress the crosstalk noise, despite the high computational efficiency. Therefore, it is important to use an appropriate number of supershots. In this study, 4 to 8 supershots was found to be the most efficient choice. However, the appropriate number of supershots might vary based on the accuracy of the starting velocity, the SNR of the data, or the frequencies selected for ESSFWI.

The source-selection method for partial-source assembling also influenced the ESSFWI results because of the methods' different degrees of randomness. We compared the regular interval, random, and random-in-subgroup methods. The convergence speed and the inverted results of each ESSFWI stage are different based on the source-selection method. These are related to both the randomness of the source-selection method and characteristic of NLCG. When we apply the NLCG to the ESSFWI, the model misfits decrease slowly at the early stage of FWI and faster after a few iterations because NLCG stacks the previous model updates to calculate the current model update; however, the number of the iterations required for NLCG to be effective varies according to the randomness of the assembling method. Based on the results, we conclude that using the regular interval method at the early stage and transitioning to the random method at the middle stage of ESSFWI will improve the results.

Table 1 presents the data and model misfits, crosstalk noise and speedup of inverted velocity with each ESSFWI method. By taking data and model misfits, crosstalk noise and speedup into account, we conclude that the most efficient approach for ESSFWI is simultaneously using RTD, NLCG, partial-source assembling with 4 supershots and the random source-selection method. This result indicates that considering both computational cost and quality of inverted velocity, and using various randomizing methods simultaneously are essential for efficient ESSFWI.

CONCLUSIONS

In this study, we used the time-domain simultaneous source method for the efficient frequency-domain FWI and applied four methods to reduce the crosstalk noise.

The first was the random encoding method. We applied the RP and RTD methods individually and simultaneously to make the time-domain simultaneous sources. Using the RTD encoding method individually gave a best result. The

second method was partial-source assembling. When partial-source assembling was used, increasing the number of supershots decreased the crosstalk noise. However, doing so also increased the computational cost. Based on the numerical tests, we concluded that using an appropriate number of supershots to balance the crosstalk noise and computational cost was the most efficient approach for ESSFWI. How the supershots were constructed for the partial-source assembling also influenced the ESSFWI results. We found that the regular interval method converged rapidly at the early stage, whereas the random method converged rapidly at the later stage of ESSFWI. We also applied NLCG to reduce crosstalk and enhance the convergence speed.

Table 1. Results of the 2D ESSFWI numerical tests. P-S(n) is partial-source assembling using n supershots. Rand is the random method and R-S is the random-in-subgroup method.

Figure	Methods	Number of iterations	Crosstalk noise	Data misfit	Model misfit	Speed up
2(b)	Starting		-	-	2.02×10^{-4}	-
2(c)	Reference	75	Not observed	2.63×10^{-3}	1.05×10^{-4}	1.0
4(a)	RP	100	Moderate	3.83×10^{-2}	1.35×10^{-4}	150.0
4(b)	RTD	100	Moderate	2.93×10^{-2}	1.23×10^{-4}	150.0
4(c)	RP&RTD	100	Moderate	3.66×10^{-2}	1.24×10^{-4}	150.0
8(b)	RTD NLCG	100	Severe	2.09×10^{-2}	1.01×10^{-4}	150.0
10(a)	RTD NLCG	400	Severe	4.43×10^{-3}	1.10×10^{-4}	37.5
10(b)	RTD NLCG P-S(2)	200	Moderate	8.11×10^{-4}	9.23×10^{-5}	37.5
10(c)	RTD NLCG P-S(4)	100	Low	1.75×10^{-3}	9.58×10^{-5}	37.5
10(d)	RTD NLCG P-S(8)	50	Not observed	9.24×10^{-3}	1.14×10^{-4}	37.5
10(e)	RTD NLCG P-S(16)	25	Not observed	3.72×10^{-2}	1.33×10^{-4}	37.5
12(b)	RTD NLCG P-S(4) Rand	100	Low	1.29×10^{-3}	9.28×10^{-5}	37.5
12(c)	RTD NLCG P-S(4) R-S	100	Low	1.33×10^{-3}	9.45×10^{-5}	37.5

Based on these studies, we applied ESSFWI to the 3D case. Two numerical tests were performed: The first test consisted of 3D ESSFWI using RTD and full-source assembling, and the second test comprised 3D ESSFWI using RTD, partial-source assembling with 4 supershots, random source selection and NLCG. The results corresponded well to those of the 2D ESSFWI tests.

This study showed that ESSFWI is an efficient method of reducing the computational costs associated with FWI and demonstrated several efficient techniques for applying ESSFWI. However, the ESSFWI described here can only be applied to a fixed receiver array. For a more practical application, ESSFWI should be investigated using marine streamer data in the future.

ACKNOWLEDGEMENTS

This work was supported by the Energy Efficiency & Resources CoreTechnology Program (No. 20132510100060) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), and by the Korea Institute of Ocean Science and Technology (Grant No. PE99534), and granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea.

REFERENCES

Anagaw, A.Y. and Sacchi, M.D., 2014. Comparison of multifrequency selection strategies for simultaneous-source full-waveform inversion. Geophysics, 79: R165-181.

Ben-Hadj-Ali, H., Operto, S. and Virieux, J., 2011. An efficient frequency-domain full waveform inversion method using simultaneous encoded sources. Geophysics, 76: R109-R124.

Berkhout, A.J. and Blacquière, G., 2011. Blended acquisition with dispersed source arrays, the next step in seismic acquisition?. Expanded Abstr., 81st Ann. Internat. SEG Mtg., San Antonio: 16-19.

Boonyasiriwat, C. and Schuster, G.T., 2010. 3D multisource full-waveform inversion using dynamic random phase encoding. Expanded Abstr., 80th Ann. Internat. SEG Mtg., Denver: 1044-1049.

Brossier, R., Operto, S. and Virieux, J., 2009. Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion. Geophysics, 74: WCC105-WCC118.

Butzer, S., Kurzmann, A. and Bohlen, T., 2013. 3D elastic full-waveform inversion of small-scale heterogeneities in transmission geometry. Geophys. Prosp. 61: 1238-1251.

Fletcher, R. and Reeves, C.M., 1964. Function minimization by conjugate gradients. Comput. J., 7: 149-154.

Jun, H., Kim, Y., Shin, J., Shin, C.S. and Min, D.-J., 2014. Laplace-Fourier-domain elastic full-waveform inversion using time-domain modeling. Geophysics, 79: R195-R208.

Jun, H., Park, E., Shin, C.S., 2015. Weighted pseudo-Hessian for frequency-domain elastic full waveform inversion. J. Appl. Geophys., 123: 1-17.

Jeong, W., Pyun, S., Son, W. and Min, D.-J., 2013. A numerical study of simultaneous-source full waveform inversion with l_1-norm. Geophys. J. Internat., 194: 1727-1737.

Krebs, J.R., Anderson, J.E., Hinkley, D., Baumstein, A., Lee, A., Neelamani, R. and Lacasse, M.D., 2009. Fast full wave seismic inversion using source encoding. Geophysics, 28: 2273-2277.

Kim, Y., Shin, C.S., Calandra, H. and Min, D.-J., 2013. An algorithm for 3D acoustic time-Laplace-Fourier-domain hybrid full waveform inversion. Geophysics, 78: R151-R166.

Marquardt, D.W., 1963. An algorithm for least squares estimation of non-linear parameters. J. Soc. Industr. Appl. Mathemat., 11: 431-441.

Plessix, R.-E., 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. Geophys. J. Internat., 167, 495-503.

Plessix, R.-E., 2009. Three-dimensional frequency-domain full-waveform inversion with an iterative solver. Geophysics, 74: WCC149-WCC157.

Pyun, S., Son, W. and Shin, C.S., 2011. 3D acoustic waveform inversion in the Laplace domain using an iterative solver. Geophys. Prosp., 59: 386-399.

Romero, L., Ghiglia, D., Ober, C. and Morton, S., 2000. Phase encoding of shot records in prestack migration. Geophysics, 65: 426-436.

Schiemenz, A. and Igel, H., 2013. Accelerated 3-D full-waveform inversion using simultaneously encoded sources in the time domain: Application to Valhall ocean-bottom cable data. Geophys. J. Internat., 195: 1970-1988.

Schuster, G.T., Wang, X., Huang, Y., Dai, W. and Boonyasiriwat, C., 2011. Theory of multisource crosstalk reduction by phase-encoded statics. Geophys. J. Internat., 184: 1289-1303.

Shin, C.S., Jang, S. and Min, D.-J., 2001. Improved amplitude preservation for prestack depth migration by inverse scattering theory. Geophys. Prosp., 49: 592-606.

Sirgue, L., Etgen, J.T. and Albertin, U., 2008. 3D frequency domain waveform inversion using time domain finite difference methods. Extended Abstr., 70th EAGE Conf., Rome: F022.

Vigh, D., Starr, E.W. and Kapoor, J., 2009. Developing earth model with full waveform inversion. The Leading Edge, 28: 432-435.

Xu, K. and McMechan, G.A., 2014. 2D frequency-domain elastic full-waveform inversion using time-domain modeling and a multistep-length gradient approach. Geophysics, 79: R41-R53.

Yang, J.Z., Liu, Y.Z. and Dong, L.G., 2013. Time-windowed frequency domain full waveform inversion using phase-encoded simultaneous source. Extended Abstr., 75th EAGE Conf., London: We 11 11.

APPENDIX

COMPUTATIONAL COST FOR THE PARTIAL-SOURCE ASSEMBLING

Partial-source assembling divides the sources into several supershots and uses several supershots simultaneously to suppress crosstalk noise (Ben-Hadj-Ali et al., 2011). The computational efficiency of the partial-source assembling method is inversely proportional to the number of supershots used simultaneously, whereas the quality of the inverted velocity is proportional to the number of supershots used simultaneously.

The computational cost of frequency-domain FWI using ESS (C_{ess}) is expressed as follows:

$$C_{ess} = N_{iter\_ess} \times N_{ess} \times (C_{forward} + C_{adjoint} + 2 \times C_{forward}) + C_{etc} \ , \qquad \text{(A-1)}$$

where N_{iter_ess} is the number of ESSFWI iterations; N_{ess} is the number of ESSs; $C_{forward}$ and $C_{adjoint}$ are the computational costs for the forward and adjoint modeling, respectively; and C_{etc} is the cost for several minor computations, including the gradient and Hessian calculations, model updating and communication between nodes. The computational cost of conventional frequency-domain FWI using individual sources (C_{conv}) is expressed as follows:

$$C_{conv} = N_{iter\_conv} \times N_{source} \times (C_{forward} + C_{adjoint} + 2 \times C_{forward}) + C_{etc} \quad , \quad \text{(A-2)}$$

where N_{iter_conv} is the number of conventional FWI iterations, and N_{source} is the number of sources to be calculated. C_{etc} is minor compared to that for forward and adjoint modeling; as a result, we can ignore it when computing the speedup achieved using simultaneous source FWI. The speedup can be expressed as follows:

$$\text{Speedup} = C_{conv}/C_{ess} = (N_{iter\_conv} \times N_{source})/(N_{iter\_ess} \times N_{ess}) \quad . \quad \text{(A-3)}$$

The speedup is thus equivalent to the number of sources in an ESS when the same number of iterations is performed.