A comparison of iterative methods performance to exact and pseudo adjoint operators in least-squares migration.

Breno Bahia, UFBA, and Reynam Pestana, CPGG/IF/UFBA and INCT-GP/CNPq/Brazil

This paper was prepared for presentation during the 15th International Congress of the Brazilian Geophysical Society, held in Rio de Janeiro, Brazil, 31 July to 3 August, 2017.

Abstract

Least-squares migration is often used to attenuate migration artifacts that arise in conventional migration due to, for instance, data incompletness and limited recording aperture. It uses iterative methods to obtain a model that best fits the data, and it requires a forward/adjoint operator pair to do so. These methods use the gradient of the cost function to estimate search directions and update the current model. The gradient is obtained by migrating the residuals between estimated and observed data at each iteration. Migration operators are regarded as adjoint of forward operators but only those which pass the dot-product test are exact adjoints. These operators estimate more accurate search directions, enhancing the convergence rates of iterative methods. We test the performance of adjoint and pseudo-adjoint operators in LSM based on three different iterative methods (steepest descent, conjugate gradients and limited-memory BFGS) in order to assess their sensitivity to the adjointness of the migration operator. We then compare the performance of each method.

Introduction

Seismic migration aims to obtain accurate distribution of the subsurface reflectivity for a given velocity distribution. There are different migration methods in the literature. Each one of these methods defines a modelling/migration operator pair which is, in some way, derived from the wave equation and have its own advantages and disadvantages. Migration operators are regarded as adjoint to its corresponding forward modelling operators but only those which pass the dot-product test (Claerbout, 1992) are exact adjoints, otherwise it should be regarded as pseudo-adjoint operators (Xu et al., 2016). Adjoint and pseudo-adjoint operators perform similarly, and introduce migration artifacts into the final image due to its non-orthogonality.

The migration artifacts degrade the quality of the final image, and much attention has been placed in how to effectively attenuate it. The Least-Squares Migration (LSM) approach is known to be effective in mitigating these artifacts, thereby improving the resolution of the final seismic image. LSM can be adapted to different migration methods such as Kirchhoff migration (Nemeth et al., 1999), one-way wave equation migration (Kuehl and Sacchi, 2002) and RTM (Ji, 2009). For any of these, LSM comes down to solving the system of normal equations which requires the inverse of the Hessian matrix. In geophysical problems, however, computing and storing the inverse of this matrix is troublesome due to its large sizes.

As an alternative, LSM uses iterative methods to solve the system of normal equations without inverting matrices, only requiring the action of the forward/adjoint operator pair to be known. At each iteration, such methods use the adjoint operator to estimate the gradient of the cost function, which is used to obtain search directions and optimized reflectivity models. Therefore, exact adjointness of these operators is required to guarantee good convergence rates in the inversion process. For instance, Ji (2009) shows that the convergence rates in LSM is enhanced by the exact adjoint operators. Exact forward/adjoint operator pairs have been formulated for post- and pre-stack acoustic RTM by Ji (2009) and Xu and Sacchi (2016), respectively. Xu et al. (2016) extended their previous work for the elastic case.

To take better advantage of the LSM, iterative methods that offers good convergence rates stand out as preferential choices in such schemes. For instance, the conjugate gradients method is vastly used in the geophysical literature (e.g., Scales (1987); Ji (2009); Xu and Sacchi (2016)). Another class of iterative methods that has good convergence rates are the quasi-Newton methods, such as the L-BFGS (Nocedal, 1980). This method uses the past m gradients and models to approximate the inverse of the Hessian matrix, which improves its convergence rates when compared to other iterative methods. Wu et al. (2015) presented a L-BFGS based LSM using Kirchhoff operators.

In this paper we follow the idea presented in Ji (2009), and we use the routines provided by him, available in `http://software.seg.org`, to test three different iterative methods (steepest descent (SD), conjugate gradients (CG) and L-BFGS) in order to assess their sensitivity to exact and pseudo-adjoint operators, as well as their efficiency. We start with a short review of LSM principles in the first part. The next section discusses the iterative methods, focusing on the L-BFGS. We follow by presenting the numerical results obtained with SEG-EAGE salt model. Finally, the conclusions are presented.

Theory

Forward-modelling operators (**L**) are usually linearized by means of the Born approximation, and it is assumed that it satisfies

$$\mathbf{d} = \mathbf{L}\mathbf{m}, \tag{1}$$

where **d** is the modeled data vector, and **m** is the reflectivity model vector (Nemeth et al., 1999). Conventional migration operators are regarded as the adjoint to modelling operators, and are represented by the mathematical transpose of **L**

$$\widehat{\mathbf{m}} = \mathbf{L}^T\mathbf{d}. \tag{2}$$

Even though Yao and Jakubowicz (2015) presented a explicit matrix representation of generalized diffraction-stack migration, the matrices correspondig to the modelling/migration operator pair are usually not built, and their action is computed on the fly by directly operating on matrix elements via indices as subroutines (Xu et al., 2016). In this case, it is necessary to ensure that the written codes correctly performs the action of $\mathbf{L}^T$ over vectors, properly accounting the aspects of matrix transposition. This is done by the dot-product test (Claerbout, 1992).

Ji (2009) and Xu and Sacchi (2016) describe similar procedures to obtain post- and pre-stack acoustic reverse time migration operators, respectively, that pass the dot-product test. Both start from the formulation of the problem using matrix-vector formulations. The difference, however, is that Ji (2009) starts from the matrix-vector form of the forward operator and take its transpose, while Xu and Sacchi (2016) derives the forward operator from the adjoint operator. Nevertheless, the idea is the same for both approaches; one should write the problem in its matrix-vector form and transpose it. Moreover, this procedure is used as guideline on how to properly write a code that passes the dot-product test but the matrices are never explicitly built (Xu and Sacchi, 2016). For more information on the operators, one could refer to Ji (2009), Xu and Sacchi (2016) or Xu et al. (2016).

Regardless of the adjointness of the operator, it is known that $\widehat{\mathbf{m}}$ is a blurred version of the true reflectivity vector **m**. Migration artifacts are introduced into the final section due to data incompleteness and approximations adopted to formulate the migration operators (Nemeth et al., 1999). This means that migration operators are non-orthogonal. In other words, the adjoint operator is useful in approximating the inverse operator, but it is not the true inverse.

A better way to approximate the inverse operator is to formulate migration as a least-squares problem (Nemeth et al., 1999). In such schemes, it is desired to find the model parameters **m** that minimizes the cost function

$$\mathbf{J} = \parallel \mathbf{L}\mathbf{m} - \mathbf{d} \parallel_2^2, \tag{3}$$

which measures the misfit between estimated and observed data. If perfect data fit is not desired (e.g., innacurate data), a regularization term can be added to equation 3. Minimizing this cost function requires its derivative with respect to the unknown **m**

$$\nabla\mathbf{J} = \mathbf{L}^T(\mathbf{L}\mathbf{m} - \mathbf{d}). \tag{4}$$

Setting it to zero results in the least-squares solution

$$\widehat{\mathbf{m}} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{d} \tag{5}$$

which is the solution to the system of normal equations.

Solving for $\widehat{\mathbf{m}}$ as given by equation 5 requires the inverse of the Hessian matrix, $\mathbf{L}^T\mathbf{L}$. Even if forward and adjoint operators are explicit matrices, storing and inverting the Hessian matrix is still troublesome due to its large sizes. For that reason, iterative techniques are employed when solving for $\widehat{\mathbf{m}}$.

Iterative methods

Iterative methods play an important role in solving linear problems, specially large ill-conditioned problems such as LSM. These methods are also referred as line search methods since, at each iteration, it computes a search direction ($\mathbf{D}_k$), and a step length (α_k) defines how far to move along this direction (Nocedal and Wright, 2006). The general expression for updating the solutions in iterative methods is given by

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\mathbf{D}_k. \tag{6}$$

The methods used in our work requires $\mathbf{D}_k$ as a descent direction ($\mathbf{D}_k^T\nabla\mathbf{J}_k < 0$) in order to guarantee the minimization of the cost function at each iteration. The search direction can be generalized as

$$\mathbf{D}_k = -\mathbf{B}_k\nabla\mathbf{J}_k \tag{7}$$

where $\mathbf{B}_k$ is a symmetric and nonsingular matrix (Nocedal and Wright, 2006). Likewise the SD method, CG approximate $\mathbf{B}_k$ as the identity matrix but ensures that estimated gradients are mutually orthogonal with respect to $\mathbf{L}^T\mathbf{L}$. The L-BFGS, on the other hand, approximates $\mathbf{B}_k$ to the inverse of the Hessian at each iteration.

The effectiveness of iterative methods is related to the search direction and step length to be used in each iteration. From equation 7 we see that $\mathbf{D}_k$ depends on the gradient of the cost function, given in equation 4. The gradient is obtained by migrating the residuals between estimated and observed data at each iteration and, therefore, uses the adjoint operator. Pseudo-adjoint operators, thus, will estimate approximate gradients while adjoint operators will provide accurate estimations of the gradient. We then expect the efficiency of iterative methods that employ adjoint operators to be enhanced when compared to those employing pseudo-adjoint operators, as shown by Ji (2009) for the CG method. In the following, we focus on briefly describing the L-BFGS method as the SD and CG methods are already well known.

Limited-memory BFGS

Quasi-Newton methods are efficient in computing descent directions at each iteration in optimization algorithms. At each iteration, it approximates the inverse of the Hessian matrix to a matrix $\mathbf{B}_{k+1}$ from $\mathbf{B}_k$. The idea is, thus, to initialize a symmetric positive definite matrix ($\mathbf{B}_0$), such as the identity, as an approximation to the inverse of the Hessian matrix and build sucessive $\mathbf{B}_k$ in such a way that this approximations mantain its properties of symmetry and positive definiteness. Another constraint imposed to $\mathbf{B}_k$

is to satisfy the quasi-Newton condition, also known as secant equation

$$\mathbf{s}_k = \mathbf{B}_{k+1}\mathbf{y}_k \quad (8)$$

where $\mathbf{s}_k = \mathbf{m}_{k+1} - \mathbf{m}_k$ and $\mathbf{y}_k = \nabla\mathbf{J}_{k+1} - \nabla\mathbf{J}_k$.

Among the quasi-Newton methods, the BFGS seems to be the most effective. Its updating expression is given by

$$\mathbf{B}_{k+1} = \mathbf{T}_k^T\mathbf{B}_k\mathbf{T}_k + \rho_k\mathbf{s}_k\mathbf{s}_k^T \quad (9)$$

where $\mathbf{T}_k = (\mathbf{I} + \rho_k\mathbf{y}_k\mathbf{s}_k^T)$ and $\rho_k = \frac{1}{\mathbf{y}_k^T\mathbf{s}_k}$. By induction it is possible to see that the last $\mathbf{B}_{k+1}$ depends on $\mathbf{B}_0$ and all available pairs $\{\mathbf{s},\mathbf{y}\}$. Therefore, at least $2k$ vectors, as well as the elements of the $\mathbf{B}_0$, have to be stored and propagated which might exceed storage capacities of currenct computers.

The BFGS can be modified to deal with storage problems. The method is called limited-memory BFGS (L-BFGS) and it can preserve the good convergence rates from the BFGS method. It uses a user defined memory to store the last m gradients and models obtained in the iterative process to construct the Hessian approximation. Thus, instead of using all previous paris $\{\mathbf{s},\mathbf{y}\}$ as in the BFGS, the L-BFGS discards earlier information, which are less likely to carry second order information to the current iteration. In addition, when the number of iterations is smaller than m, one should use the available pairs $\{\mathbf{s},\mathbf{y}\}$. Also, to discard information means to define $\mathbf{T} = \mathbf{I}$ and $\rho\mathbf{s}\mathbf{s}^T = 0$.

So far it was discussed how to obtain an approximation to the inverse of the Hessian matrix. It is still necessary to compute the product of the obtained approximation with the gradient vector at the current iteration. This product is computed recursively, as shows Nocedal (1980), and does not requires the storage of the matrices $\mathbf{B}_k$ from previous iterations. Instead, it is suffice to store the m pairs $\{\mathbf{s},\mathbf{y}\}$, reducing the memory requirements and computational costs of the method. The product is computed with a two-loop recursion algorithm (Nocedal and Wright, 2006), and it is given in algorithm 1.

Algorithm 1 L-BFGS - Computes the direction $\mathbf{D}_k = -\mathbf{B}_k\nabla\mathbf{J}_k$

Start a matrix $\mathbf{B}_0$
$\mathbf{q} = \nabla\mathbf{J}_k$
for $i = k-1, \dots, k-m$ **do**
  $\alpha_i = \rho_i\mathbf{s}_i^T\mathbf{q}$
  $\mathbf{q} = \mathbf{q} - \alpha_i\mathbf{y}_i$
end for
$\mathbf{D}_0 = \mathbf{B}_0\mathbf{q}$
for $i = k-m, k-m+1, \dots, k-1$ **do**
  $\beta = \rho_i\mathbf{y}_i^T\mathbf{r}_i$
  $\mathbf{D}_{i+1} = \mathbf{D}_i + (\alpha_i - \beta)\mathbf{s}_i$
end for
Returns $\mathbf{D}$

Only for the L-BFGS method we use a backtracking algorithm to obtain step lengths that satisfy the Armijo's condition. The initial step length is always set to 1. More accurate step lengths might be obtained by using step lengths that satisfy the curvature condition (Nocedal and Wright, 2006).

Numerical example

We test all iterative methods employing both adjoint and pseudo-adjoint operators. We use the post-stack RTM operators published by Ji (2009) with the SEG-EAGE salt model (Figure 1) to generate the zero-offset section shown in Figure 2. This section is considered the observed data. The used spacial spacing is of $4.5\ m$ in both dimensions, and temporal sampling rate of $1\ ms$.

First we illustrate the differences in adjoint and pseudo-adjoint operators. We migrate the data with both operators to obtain the results shown in Figures 3a and 3b. As discussed by Ji (2009), both operators perform same action and have similar results. It is possible to notice a difference in amplitude in the migrated images, as well as the presence of migration artifacts due to the non-orthogonality of the operators.

In order to attenuate the migration artifacts, we then test both operators in LSM schemes. Ji (2009) has already shown that the adjoint operator fits the data better than the pseudo-adjoint operator using CG. We extend his analysis to the steepest descent and L-BFGS methods in order to assess its sensibility to the adjointness of the operators. All methods are gradient-dependent requiring the migration operator for its computation. As already discussed, we expect the search direction estimated by the pseudo-adjoint operators to be less accurate than those obtained by the adjoint, leading to slower convergence rates for the prior. The initial model was set to a null vector, thus the first iteration for all methods is the same as the SD method. Moreover, the number of iterations was set to 10. We compared the results obtained in the last iteration using adjoint and pseudo-adjoint operators in Figures 4, 5 and 6 for the SD, CG and L-BFGS methods, respectively. For all methods, the LSM was able to effectively attenuate the migration artifacts when the adjoint operator was employed. On the other hand, the schemes where the pseudo-adjoint was used show less improvement at the end of 10 iterations. Figure 7 shows the obtianed convergence curves which illustrate the superior performance of adjoint operators over pseudo-adjoint operators.

Finally, we compare the iterative methods efficiency in the LSM. Figure 8 shows the convergence rates for all methods using the adjoint operator only. We notice that both CG and L-BFGS have better convergence rates than SD. In addition, CG and L-BFGS have very similar convergence rates where CG was outperformed by the L-BFGS method from iteration number 7, approximately.

Conclusion

We have reviewed the concepts of adjoint and pseudo-adjoint operators by means of the dot-product test. We have used the operators provided by Ji (2009) to test its effects on the performance of iterative methods when applied in a least-squares migration scheme. We formulate LSM based on the steepest descent, conjugate gradients and L-BFGS methods. The performance of gradient-based iterative methods is affected by the adjointness of the migration operator. The convergence rates of each method is enhanced by the adjoint operator when compared to the pseudo-adjoint. This enhancement is associated to the search direction adopted by each method, which is more accurately estimated by the exact adjoint operator. We confirm the results presented by Ji (2009), where the exact adjointness between the modelling and migration operators has direct influence in least-squares inversion schemes. However, although not recommended, this does not means that pseudo-adjoint operators cannot be used in LSM schemes. In this case, one should bear in mind that either the number of iterations required to obtain good data fit would be much bigger, which increases computational costs, or that the SD method could be a better option. Furthermore, we compare the performance of the iterative methods. We observe that the L-BFGS method stands as a good alternative to the LSM technique when employing numerically exact adjoint operator pairs, showing similar to better convergence rates than CG.

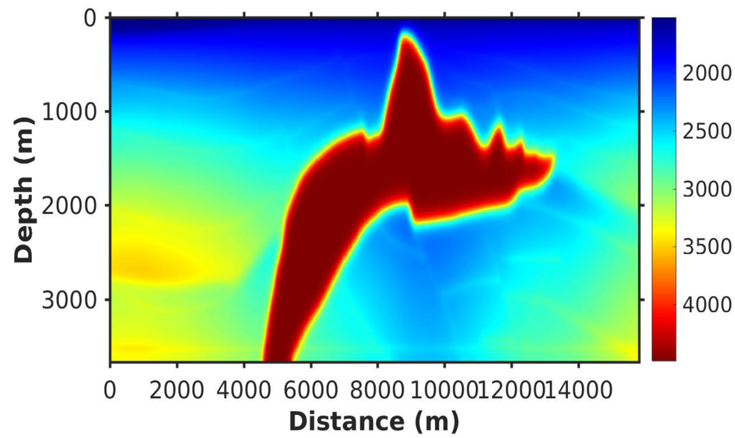


Figure 1: Smooth velocity field of SEG-EAGE salt model.

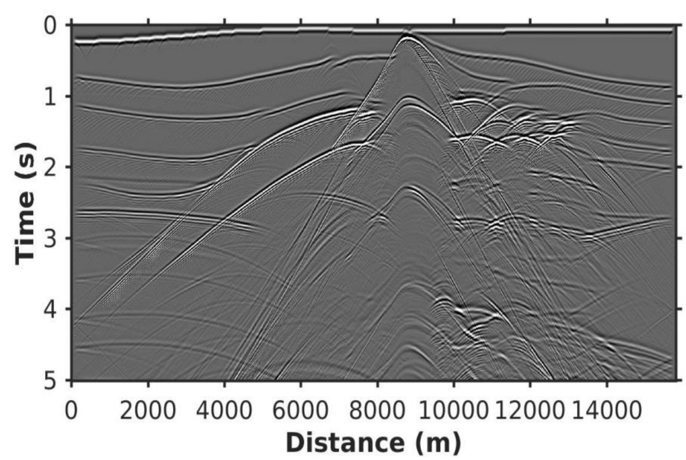


Figure 2: Synthetic data obtained through forward modelling.

References

Claerbout, J. F., 1992, Earth soundings analysis: Processing versus inversion, volume **6**: Blackwell Scientific Publications Cambridge, Massachusetts, USA.

Ji, J., 2009, An exact adjoint operation pair in time extrapolation and its application in least-squares reverse-time migration.: Geophysics, **74**, H27.

Kuehl, H. and M. Sacchi, 2002, Robust avp estimation using least-squares wave-equation migration, *in* SEG Technical Program Expanded Abstracts 2002, 281–284, Society of Exploration Geophysicists.

Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data.: Geophysics-Wisconsin then Tulsa- Society of Exploration Geophysicists, **64**, 208 – 221.

Nocedal, J., 1980, Updating quasi-newton matrices with limited storage.: Mathematics of Computation, 773.

Nocedal, J. and S. J. Wright, 2006, Numerical optimization. [electronic resource]. Springer series in operations research: New York : Springer, c2006.

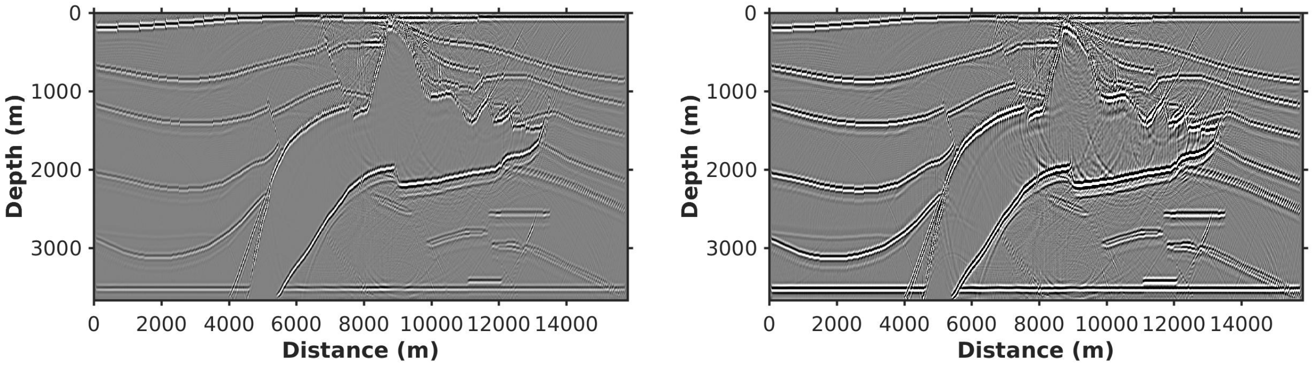
Scales, J. A., 1987, Tomographic inversion via the conjugate gradient method: Geophysics, **52**, 179–185.

Wu, S., Y. Wang, Y. Zheng, and X. Chang, 2015, Limited-memory bfgs based least-squares pre-stack kirchhoff depth migration: Geophysical Journal International, **202**, 738–747.

Xu, L. and M. Sacchi, 2016, Least squares reverse time migration with model space preconditioning and exact forward/adjoint pairs: Presented at the 78th EAGE Conference and Exhibition 2016.

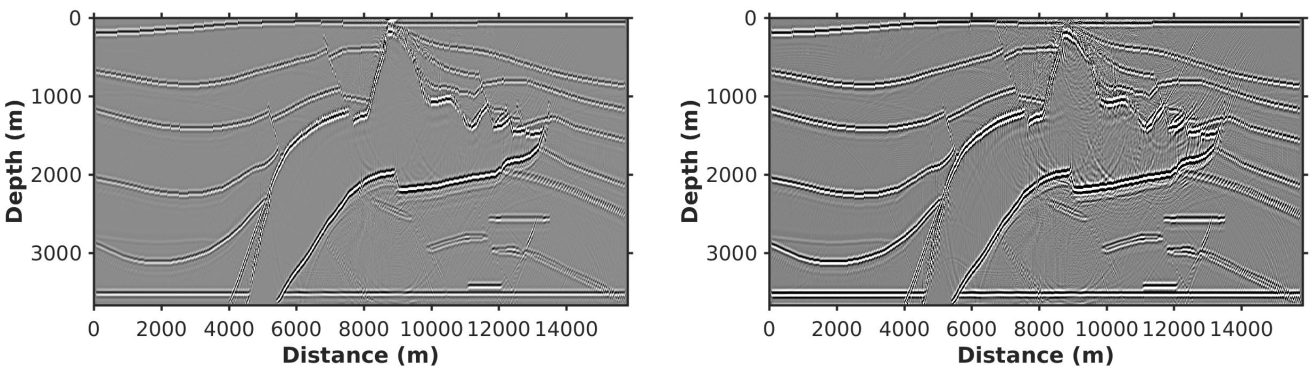
Xu, L., A. Stanton, and M. Sacchi, 2016, Elastic least-squares reverse time migration, *in* SEG Technical Program Expanded Abstracts 2016, 2289–2293, Society of Exploration Geophysicists.

Yao, G. and H. Jakubowicz, 2015, Least-squares reverse-time migration in a matrix-based formulation: Geophysical Prospecting.



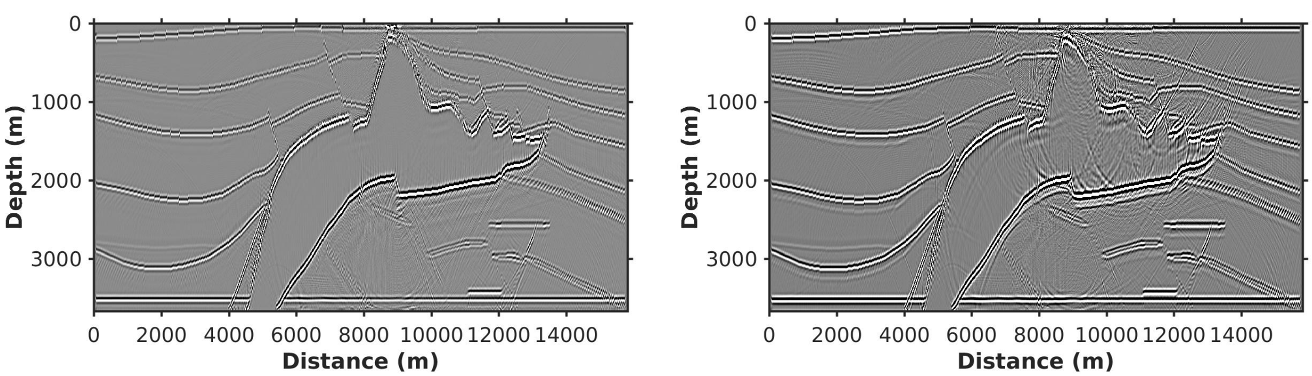
(a) (b)

Figure 3: Migrated image with (a) adjoint operator and (b) pseudo-adjoint operator.



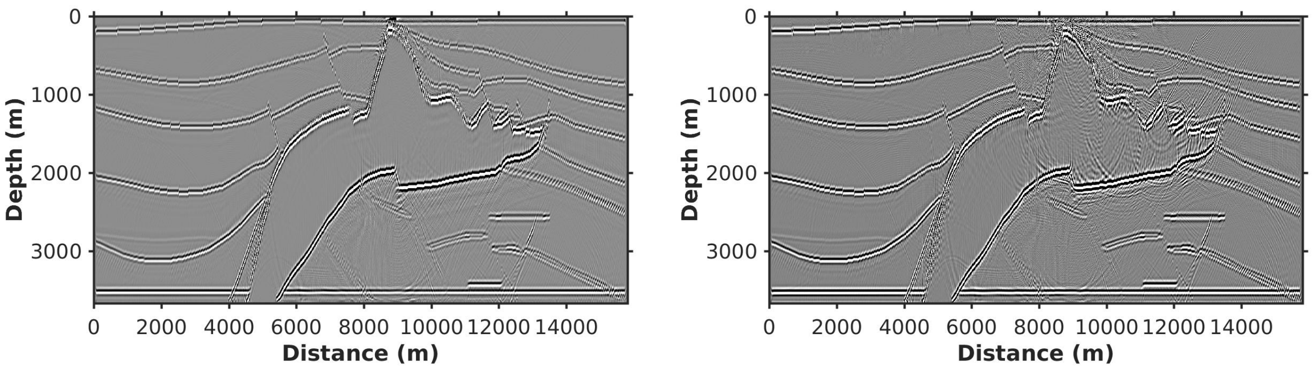
(a) (b)

Figure 4: Migrated image after 10 LS iterations based on the SD method with (a) adjoint operator and (b) pseudo-adjoint operator.



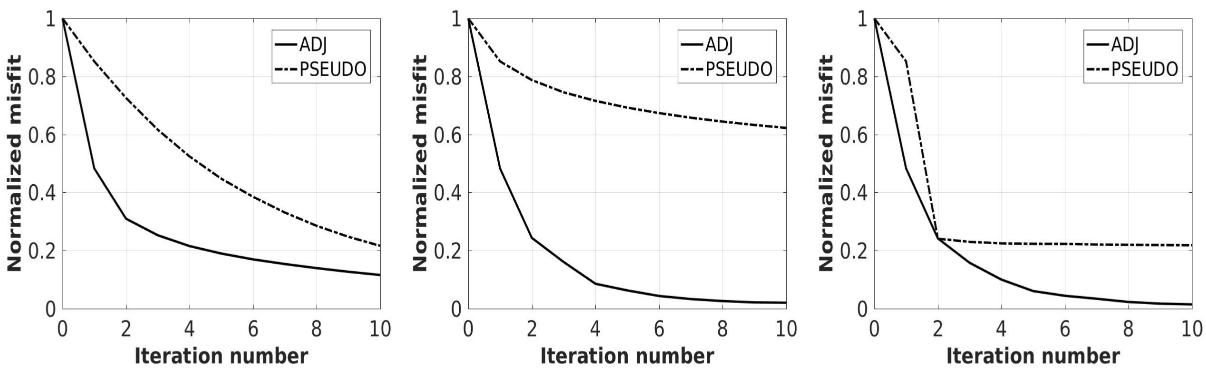
(a) (b)

Figure 5: Migrated image after 10 LS iterations based on the CG method with (a) adjoint operator and (b) pseudo-adjoint operator.



(a) (b)

Figure 6: Migrated image after 10 LS iterations based on the L-BFGS method with (a) adjoint operator and (b) pseudo-adjoint operator.



(a) (b) (c)

Figure 7: Convergence curves for (a) SD (b) CG and (c) L-BFGS methods.

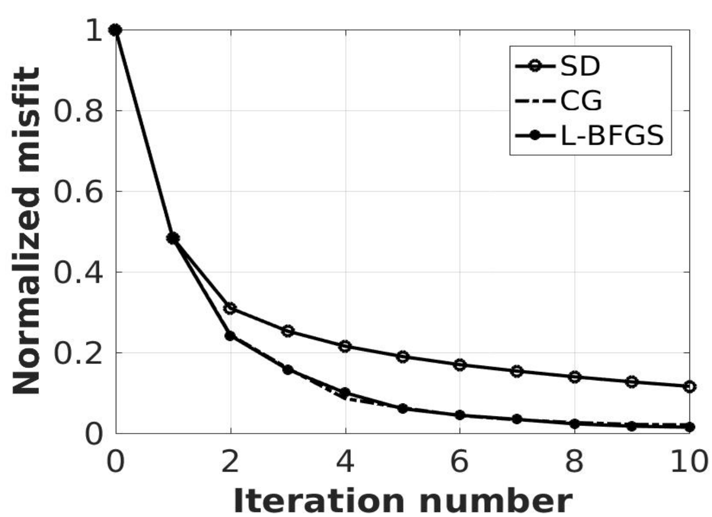


Figure 8: Convergence curve for LSM based on SD (open circles line), CG (dashed line) and L-BFGS (filled circles line) methods.