GEOPHYSICS, VOL. 86, NO. 1 (JANUARY-FEBRUARY 2021); P. R99–R108, 6 FIGS., 2 TABLES.
10.1190/GEO2020-0462.1

Anderson acceleration for seismic inversion

Yunan Yang[1]

ABSTRACT

State-of-the-art seismic imaging techniques treat inversion tasks such as full-waveform inversion (FWI) and least-squares reverse time migration (LSRTM) as partial differential equation-constrained optimization problems. Due to the large-scale nature, gradient-based optimization algorithms are preferred in practice to update the model iteratively. Higher-order methods converge in fewer iterations but often require higher computational costs, more line-search steps, and bigger memory storage. A balance among these aspects has to be considered. We have conducted an evaluation using Anderson acceleration (AA), a popular strategy to speed up the convergence of fixed-point iterations, to accelerate the steepest-descent algorithm, which we innovatively treat as a fixed-point iteration. Independent of the unknown parameter dimensionality, the computational cost of implementing the method can be reduced to an extremely low dimensional least-squares problem. The cost can be further reduced by a low-rank update. We determine the theoretical connections and the differences between AA and other well-known optimization methods such as L-BFGS and the restarted generalized minimal residual method and compare their computational cost and memory requirements. Numerical examples of FWI and LSRTM applied to the Marmousi benchmark demonstrate the acceleration effects of AA. Compared with the steepest-descent method, AA can achieve faster convergence and can provide competitive results with some quasi-Newton methods, making it an attractive optimization strategy for seismic inversion.

INTRODUCTION

The fast growth of computational power popularizes numerous techniques that use full wavefields in seismic imaging (Tarantola and Valette, 1982). In particular, full-waveform inversion (FWI) (Virieux and Operto, 2009) and least-squares reverse time migration (LSRTM) (Dai and Schuster, 2013) aim to reconstruct subsurface properties such as the wave velocity and the material density by minimizing an objective function that measures the discrepancy between synthetic data and observed data. Iterative optimization algorithms are then applied to find the optimal solution (Métivier et al., 2017).

For local optimization, the descent direction depends on the gradient and the Hessian information of the objective function with respect to the model parameters. Theoretically, the step size along the descent direction should be determined by a line search to guarantee a sufficient decrease in the objective function and avoid overshooting. However, the process of a backtracking line search could incur a considerable amount of extra wave modeling. Sometimes, to reduce the computational cost of a line-search method and avoid overshooting, a tiny fixed step size is preferred instead, but this slows down the convergence. Similarly, Newton's method is not widely used in practical seismic inversion due to the cost of calculating and storing the Hessian matrix, despite being known to offer a quadratic convergence rate. For large-scale optimization problems such as seismic inversion, a better rate of convergence often comes at the cost of memory and computing power. In practice, it is often best to balance computing and memory considerations.

In the past two decades, Anderson acceleration (AA) has been widely used in several applied fields for problems that can be solved by a fixed-point iteration. The application of AA includes flow problems (Pollock et al., 2018), solving nonlinear radiation-diffusion equations (An et al., 2017), and wave propagation (Yang et al., 2020). It is closely related to Pulay mixing and direct inversion on the iterative subspace (DIIS) (Pulay, 1980; Kudin et al., 2002), which are prominent methods in self-consistent field theory (Ceniceros and Fredrickson, 2004). AA is also becoming popular in the numerical analysis

Manuscript received by the Editor 30 June 2020; revised manuscript received 10 September 2020; published ahead of production 8 October 2020; published online 07 January 2021.
[1]New York University, Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, New York 10012, USA. E-mail: yunan.yang@nyu.edu (corresponding author).

DOI:10.1190/geo2020-0462.1

community (Walker and Ni, 2011; Toth and Kelley, 2015; Zhang et al., 2018; Pollock and Rebholz, 2019; Evans et al., 2020). The literature on this subject is broad, so we only mention a few papers to show the variety of results obtained by AA. In contrast to Picard iteration (Picard, 1893; Butenko and Pardalos, 2014), which uses only one previous iterate, the method proceeds by linearly recombining a list of previous iterates to approximately minimize the linearized fixed-point residual. AA can be applied directly to accelerate fixed-point operators that arise naturally from solving partial differential equations (PDEs). The method was mainly used in optimization-free scenarios until the past few years. Recently, AA has been showing promising results in accelerating optimization algorithms (Li and Li, 2018; Peng et al., 2018; Fu et al., 2019; Mai and Johansson, 2019) and in machine learning (Geist and Scherrer, 2018).

In this paper, we aim to combine the fast convergence of Newton-type methods with the low cost of only evaluating the gradient. We do this by applying an acceleration strategy introduced by Anderson (1965) to the steepest-descent algorithm. We first reformulate the iterative formula as a fixed-point operator. In contrast to the classic gradient descent, AA produces a new iterate as a linear combination of several previous iterates. The linear coefficients are selected optimally to achieve the best reduction in the linearized fixed-point residual. As an acceleration strategy for the steepest-descent method, AA can achieve competitive convergence speed with respect to methods such as limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) and the nonlinear conjugate gradient descent (nCG) while reducing the computational cost of computing the exact or approximating the (inverse) Hessian matrix. We illustrate the performance of these methods as the optimization algorithm for FWI and LSRTM in the numerical examples.

THEORY

In this section, we first introduce the algorithmic details of AA for fixed-point problems and explain its similarities and differences with Picard iteration. Later, we review some essential background regarding FWI and LSRTM. Throughout the paper, we assume that the forward model is an acoustic wave equation with a constant density.

Anderson acceleration

AA is an acceleration strategy introduced to improve the slow convergence of Picard iterations (Anderson, 1965). We present the details of AA in Algorithm 1. The memory parameter $\mathcal{M}$ determines the number of additional past iterates that need to be stored to compute the next iterate. For example, when $\mathcal{M} = 0$, AA reduces to the Picard iteration because the $(k+1)$th iterate p_{k+1} only depends on p_k and $p_{k+1} = G(p_k)$, where G is the fixed-point operator. For a nonzero $\mathcal{M}$, p_{k+1} is a linear combination of the previous $\mathcal{M}+1$ iterates, together with their evaluation by the fixed-point operator G; see equation 2 for a detailed updating formula. If $\mathcal{M} = +\infty$ and the damping parameter β_k in equation 2 is also chosen optimally at every iteration, AA is essentially equivalent to the generalized minimal residual method (GMRES) when G is a linear fixed-point operator and the fixed-point solution solves the square linear system $Ax = b$ (Toth and Kelley, 2015). The damping parameter β_k, which could vary at different iteration k, controls the balance between the linear combination of the iterates $\{p_{k-\mathcal{M}+i}\}_{i=0}^{\mathcal{M}}$ and the linear combination of their evaluation by the operator $\{G(p_{k-\mathcal{M}+i})\}_{i=0}^{\mathcal{M}}$.

At iteration k, the coefficient vector $\alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_{\mathcal{M}_k}^{(k)})^T$ is determined by minimizing the sum of the weighted fixed-point residuals. The sum of all of the coefficients must total one so that the fixed-point solution p^* is preserved under the updating equation 2 of AA. The Picard iteration fits into the updating equation 2 with the weighting vector $(0, 0, \ldots, 0, 1)^T$ for every k. Because the weighting vector for AA is obtained from the optimization problem (equation 1), AA is always at least as good as Picard iteration as

$$\left\| \sum_{i=0}^{\mathcal{M}_k} \alpha_i^{(k)} f_{k-\mathcal{M}_k+i} \right\|_* \le \|f_k\|_*, \tag{3}$$

where $f_i = G(p_i) - p_i$ is the fixed-point residual of the ith iteration.

Through a change of variable, one can remove the constraints in equation 1 to simplify the optimization step. Consider a new vector $\gamma^{(k)} = (\gamma_0^{(k)}, \ldots, \gamma_{\mathcal{M}_k-1}^{(k)})^T$ defined by the optimal parameter $\alpha^{(k)}$ where

$$\gamma_i^{(k)} = \alpha_0^{(k)} + \ldots + \alpha_i^{(k)}, 0 \le i \le \mathcal{M}_k - 1. \tag{4}$$

Consider the matrix A_k given by

$$A_k = (f_{k-\mathcal{M}_k+1} - f_{k-\mathcal{M}_k}, \ldots, f_k - f_{k-1}), \tag{5}$$

whose column vectors are the differences in the fixed-point residual between two consecutive iterations. The optimization step (equation 1) is equivalent to the following unconstrained optimization problem:

Algorithm 1. The AA strategy.

Input: Given the initial guess p_0 and memory parameter $\mathcal{M} \ge 1$; G is the given fixed-point operator. Set $p_1 = G(p_0)$.

for $k = 0, 1, 2, \ldots$ **do**

Step 1: Set $\mathcal{M}_k = \min(\mathcal{M}, k)$ and matrix $F_k = (f_{k-\mathcal{M}_k}, \ldots, f_k)$, where $f_i = G(p_i) - p_i$ is the fixed-point residual of the ith iterate.

Step 2: Find the optimal weights $\alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_{\mathcal{M}_k}^{(k)})^T$ by the optimization problem

$$\min_{\sum_{i=0}^{\mathcal{M}_k} \alpha_i^{(k)} = 1} \|F_k \alpha^{(k)}\|_*. \tag{1}$$

Step 3: Update the next iterate p_{k+1}

$$p_{k+1} = (1-\beta_k) \sum_{i=0}^{\mathcal{M}_k} \alpha_i^{(k)} p_{k-\mathcal{M}_k+i} + \beta_k \sum_{i=0}^{\mathcal{M}_k} \alpha_i^{(k)} G(p_{k-\mathcal{M}_k+i}). \tag{2}$$

end for

DOI:10.1190/geo2020-0462.1

$$\gamma^{(k)} = \text{argmin}_\gamma \|A_k\gamma - f_k\|_*. \tag{6}$$

There are several variants regarding the choice of the norm $\|\cdot\|_*$ in the optimization step. For example, one can use the ℓ^1, ℓ^2, or the ℓ^∞ norm as the objective function. Alternatively, a weighted ℓ^2 norm may improve the conditioning of the fixed-point operator or enforce the spectral bias toward certain modes of the solution (Yang et al., 2020). The optimal weights may not be the same among different choices of the objective function, and the cost of solving the corresponding optimization problem also can be radically different. For example, linear programming is required to solve the optimization under the ℓ^1 and the ℓ^∞ norms. However, for the ℓ^2 norm, $\gamma^{(k)}$ becomes the least-squares solution to the following linear system:

$$A_k\gamma^{(k)} = f_k, \tag{7}$$

where $f_k = G(p_k) - p_k$ is the fixed-point residual at iteration k. If $\beta_k = 1$ for any k, then we can rewrite the updating formula in terms of $\gamma^{(k)}$ as follows:

$$p_{k+1} = G(p_k) - \sum_{i=0}^{\mathcal{M}_k-1} \gamma_i^{(k)}[G(p_{k-\mathcal{M}_k+i+1}) - G(p_{k-\mathcal{M}_k+i})]. \tag{8}$$

It is computationally efficient to implement AA based on equation 8. The size of A_k is n by $\mathcal{M}_k$, where n is the dimension of the parameter and $\mathcal{M}_k = \min(\mathcal{M}, k) \le \mathcal{M}$. The memory parameter $\mathcal{M}$ of AA is often chosen to be small. A rank-updated QR factorization can further reduce the cost of solving equation 7 (Golub and Van Loan, 2013, section 12.5.1).

FWI and LSRTM

Seismic inversion aims to estimate the distribution of underground material properties. They are large-scale inverse problems that we treat as constrained optimization problems based on the deterministic approach of solving inverse problems.

FWI is a nonlinear inverse technique that uses the entire wavefield information to estimate the earth's properties. Without loss of generality, the PDE constraint of FWI is the following acoustic wave equation with zero initial condition and nonreflecting boundary conditions:

$$\begin{cases} m(\mathbf{x})\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} - u(\mathbf{x},t) = s(\mathbf{x},t), \\ u(\mathbf{x},0) = 0, \\ \frac{\partial u}{\partial t}(\mathbf{x},0) = 0. \end{cases} \tag{9}$$

We set the model parameter $m(\mathbf{x}) = 1/c(\mathbf{x})^2$, where $c(\mathbf{x})$ is the wave velocity, $u(\mathbf{x},t)$ is the forward wavefield, and $s(\mathbf{x},t)$ is the wave source. The velocity parameter m is often the target of reconstruction. Equation 9 is a linear PDE, but it defines a nonlinear operator $\mathcal{F}$ that maps $m(\mathbf{x})$ to $u(\mathbf{x},t)$. In FWI, we translate the inverse problem of finding the model parameter based on the observable seismic data to a constrained optimization problem:

$$m^* = \text{argmin}_m J(m), \qquad J(m) = \frac{1}{2}\|f(m) - g\|_2^2. \tag{10}$$

The least-squares norm is commonly used as the objective function J to calculate the misfit between the synthetic data $f(m) = R\mathcal{F}(m)$ and the observed data g. Here, R is the projection operator that extracts the wavefield u at the receiver locations. There are other choices of objective functions to mitigate the cycle-skipping issues of FWI (Yang et al., 2018).

LSRTM is a migration method designed to improve the image quality generated by RTM. It is formulated as a linear inverse problem based on the Born approximation, a first-order linearization of the map $\mathcal{F}$ (Hudson and Heritage, 1981). From now on, we denote the forward operator of LSRTM, i.e., the Born modeling, as $\mathcal{L} = \delta\mathcal{F}/\delta m$, the functional derivative of $\mathcal{F}$ with respect to m. The linear operator $\mathcal{L}$ maps a small perturbation in the velocity m_r to the scattering wavefield u_r:

$$\begin{cases} m_0\frac{\partial^2 u_r(\mathbf{x},t)}{\partial t^2} - u_r(\mathbf{x},t) = -m_r\frac{\partial^2 u_0(\mathbf{x},t)}{\partial t^2}, \\ u_r(\mathbf{x},0) = 0, \\ \frac{\partial u_r}{\partial t}(\mathbf{x},0) = 0. \end{cases} \tag{11}$$

Here, m_0 is the given background velocity and the background wavefield $u_0 = \mathcal{F}(m_0)$. We seek the reflectivity model by minimizing the least-squares error between the observed data d_r and the predicted scattering wavefield $Lm_r = R\mathcal{L}m_r$,

$$m_r^* = \text{argmin}_{m_r} J(m_r), \qquad J(m_r) = \frac{1}{2}\|Lm_r - d_r\|_2^2. \tag{12}$$

To solve for m^* in equation 10 and m_r^* in equation 12, optimization algorithms heavily rely on the gradient and the Hessian information of the objective function J. In seismic inversions, one can obtain the gradient of a parameter by solving the forward equation and the adjoint equation once, based on the adjoint-state method (Plessix, 2006). The adjoint equation for FWI and LSRTM is as follows:

$$\begin{cases} m\frac{\partial^2 v(\mathbf{x},t)}{\partial t^2} - v(\mathbf{x},t) = -R^*\frac{\partial J}{\partial f}, \\ v(\mathbf{x},T) = 0, \\ \frac{\partial v}{\partial t}(\mathbf{x},T) = 0. \end{cases} \tag{13}$$

For LSRTM, the m in equation 13 is the background velocity m_0. The term $\partial J/\partial f$ is the Fréchet derivative of the objective function with respect to the synthetic data f, also known as the adjoint source. If J is the least-squares norm, $\partial J/\partial f = f - g$ is the data residual. The functional derivative of the objective function J with respect to the model parameter m is

$$\frac{\partial J}{\partial m} = -\int_0^T \frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} v(\mathbf{x},t)dt, \tag{14}$$

where u and v are the forward and adjoint wavefields, respectively. For FWI, u is the solution to the acoustic wave equation 9, whereas for LSRTM, u is the solution to linearized wave equation 11. An outstanding advantage of the adjoint-state method is that the number of wave simulations to compute the gradient is independent of the size of m. The model parameter then can be updated by a gradient-based optimization algorithm iteratively until meeting the stopping criteria. The Hessian matrix also can be computed based on the adjoint-state method if it is needed for optimization, uncertainty quantification, or resolution analysis.

METHOD

Within the framework of iterative methods, we treat the gradient-descent algorithm as a fixed-point iteration, where the fixed-point solution is the optimal model parameter. Consider an objective function $J(p)$ for the unknown parameter p. If we choose to minimize $J(p)$ by the steepest-descent algorithm, then the $(k+1)$th iterate p_{k+1} is obtained by the kth iterate p_k and the gradient vector for p_k. The step size is chosen to guarantee a sufficient decrease in the objective function. Without loss of generality, we fix the step size as a small positive constant η and obtain the updating formula by the steepest descent:

$$p_{k+1} = p_k - \eta \left.\frac{\partial J}{\partial p}\right|_{p=p_k} = G(p_k). \tag{15}$$

Because the right side of equation 15 only depends on p_k, one can regard the updating formula as a fixed-point operator G applied to p_k. Equation 15 can be considered as the Picard iteration for G. The fixed-point solution p^* should satisfy

$$p^* = G(p^*) \Leftrightarrow \left.\frac{\partial J}{\partial p}\right|_{p=p^*} = 0. \tag{16}$$

Thus, p is the fixed-point solution of G if and only if the gradient of the objective function J is zero at p.

Seismic inverse problems are often ill-posed and suffer from cycle-skipping issues. Typically, the zero-gradient condition is far from sufficient to guarantee the optimality, especially for FWI. There has been extensive literature on tackling the nonconvexity (Engquist and Yang, 2020; Symes, 2020). In this paper, we focus on accelerating the convergence and not addressing the cycle-skipping issues, which is another important research topic by itself. Thus, we assume that the initial guess p_0 in this paper is sufficient so that the optimization problem does not suffer from local minima.

Algorithm 2. The ℓ^2-based AA for the gradient descent.

Input: Given the initial model p_0, the memory parameter $\mathcal{M} \geq 1$, and the fixed-point operator G based on the gradient-descent update (equation 15). Set $p_1 = G(p_0)$.

for $k = 1, 2, \ldots$ until convergence or maximum iteration **do**

Step 1: Set $\bar{p}_{k+1} = G(p_k) = p_k - \eta \mathcal{G}_k$. Update A_k and f_k using the gradient vectors $\{\mathcal{G}_i\}_{i=k-\mathcal{M}_k}^{k}$ where $\mathcal{M}_k = \min(\mathcal{M}, k)$.

Step 2: Find the least-squares solution $\gamma^{(k)}$ to the linear system (equation 7) by using the low-rank QR update.

Step 3: Compute the new iterate $\widetilde{p}_{k+1}$ following equation 8.

Step 4: Apply the backtracking line search for λ such that $J(\lambda \widetilde{p}_{k+1} + (1-\lambda)\bar{p}_{k+1})$ has a sufficient decrease compared to $J(p_k)$.

Step 5: Set the new iterate as

$$p_{k+1} = \lambda \widetilde{p}_{k+1} + (1-\lambda)\bar{p}_{k+1}. \tag{19}$$

end for

Given the fixed-point operator G defined by the steepest-descent algorithm (equation 15), we aim to accelerate the convergence by applying AA. First, we rewrite equation 5 as follows:

$$A_s = f_k = G(p_k) - p_k = -\eta \mathcal{G}_k, \tag{17}$$

$$A_k = -\eta(\mathcal{G}_{k-\mathcal{M}_k+1} - \mathcal{G}_{k-\mathcal{M}_k}, \ldots, \mathcal{G}_k - \mathcal{G}_{k-1}), \tag{18}$$

where $\mathcal{G}_k = \partial J/\partial p|_{p_k}$ is the gradient vector for the iterate p_k. We recall that the core of AA is to solve a linear system (equation 7) for $\gamma^{(k)}$, which gives the optimal coefficients $\alpha^{(k)}$ by a change of variable. Applying AA to accelerate the gradient descent, we remark that the main components of the linear system (equation 7) are constructed only by the gradients $\{\mathcal{G}_i\}_{i=k-\mathcal{M}_k}^{k}$ of the optimization. Thus, the memory requirements of AA are the same as the L-BFGS algorithm.

For typical fixed-point problems, the fixed-point residual is the indicator of convergence. That is, we judge the convergence by comparing the norm of $G(p) - p$. As a unique feature of our application, the fixed-point operator comes from an optimization problem. Thus, the objective function also can be used in AA to improve the convergence further. The traditional AA described in Algorithm 1 does not have such a step related to the objective function. Therefore, by combining the fixed-point residual and the objective function, we describe a new workflow of AA for seismic inversion in Algorithm 2. The final p_{k+1} is a linear combination of the output by the gradient descent $\bar{p}_{k+1}$ and the optimized new iterate by AA $\widetilde{p}_{k+1}$. A backtracking line search following the Wolfe condition is applied to determine the weighting between $\bar{p}_{k+1}$ and $\widetilde{p}_{k+1}$ such that we can achieve the best decrease in the objective function J; see Algorithm 2 for more details.

NUMERICAL EXAMPLE

In this section, we present several inversion tests for FWI and LSRTM and compare the performance of AA with other optimization algorithms such as L-BFGS and the steepest-descent method.

Full-waveform inversion

We aim to reconstruct the Marmousi velocity model that is 3 km in depth and 9 km in width (Figure 1a) from a smoothed initial guess as shown in Figure 1b. There are 11 equally spaced sources at 150 m below the air-water interface. The source is a Ricker wavelet centered at 15 Hz, and 4 s is the total recording time. There is no cycle skipping with the chosen initial model. The most time-consuming component of FWI is seismic modeling, which is essential for gradient calculation. Thus, instead of counting the number of iterations, we use the number of gradient evaluations to measure the performance.

Figure 2 shows the FWI results using AA ($\mathcal{M} = 20$), L-BFGS ($\mathcal{M} = 20$), nCG, and the steepest descent after 1000 gradient evaluations. The same backtracking line search following the Armijo rule and the curvature condition is

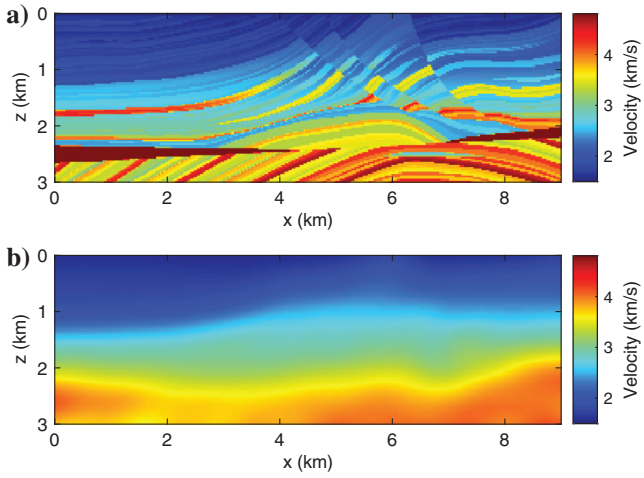

Figure 1. (a) Marmousi true velocity and (b) Marmousi initial velocity for FWI.

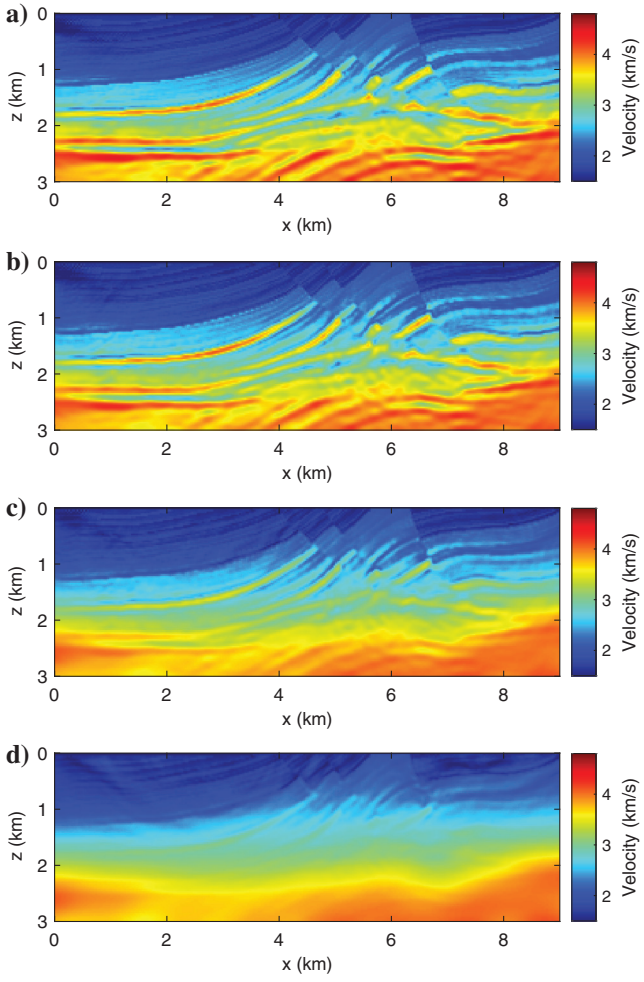

Figure 2. (a) FWI using AA after 1000 gradient evaluations, (b) FWI using L-BFGS after 1000 gradient evaluations, (c) FWI using nonlinear CG after 1000 gradient evaluations, and (d) FWI using the steepest-descent method after 1000 gradient evaluations.

applied to all of the methods. The convergence history for the ℓ^2 objective function and the norm of the gradient is shown in Figure 3. The results by AA and L-BFGS illustrate better resolution than that by nCG. The steepest-descent method converges slowly. In both plots, AA demonstrates a faster convergence rate than L-BFGS and nCG. Known as quasi-Newton methods, L-BFGS and CG converge in fewer iterations than AA. However, more gradient evaluations are spent on the backtracking line search, which increases the overall CPU time. The drastic improvement in the convergence rate between AA and the steepest-descent method shows the benefits of this simple strategy by linearly recombining previous iterates. Considering the low cost of implementation, AA can be an attractive optimization technique for FWI. More analysis between AA and L-BFGS is presented in the next section.

Inversion with noise

We present another FWI example that, unlike the previous set of tests, introduces mean-zero noise to the observed data to make the inversion test more representative of results expected on real data. The noise follows a uniform distribution, and the signal-to-noise ratio (S/N) is 0.55 dB. We plot one trace of the clean data and the noisy data in Figure 4a for illustration. All of the other settings remain the same as the noise-free example. Figure 4b is the inversion result using AA to accelerate the steepest-descent algorithm, whereas Figure 4d is the inversion result without acceleration. We also perform a test for the L-BFGS algorithm (see Figure 4c). All experiments are stopped after 1000 gradient calculations as previously. Compared with the noise-free results in the previous section, one can observe artificial oscillatory features in all of the images resulting from noise overfitting. However, the noise footprints are equally strong for the inversion using L-BFGS and the one using AA. Although the reconstruction by the steepest-descent method seems to be less noisy, it also recovers fewer features of the

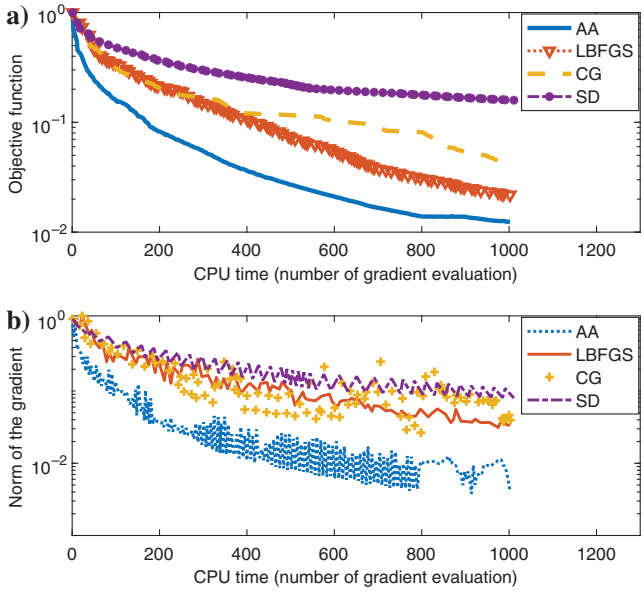

Figure 3. (a) FWI convergence history of the objective function and (b) FWI convergence history of the gradient norm, in terms of computational time (measured by the number of FWI gradient evaluations).

true Marmousi model. This is due to its slower convergence compared with AA and L-BFGS. Typically, one can expect that the artifacts in the reconstructed model will be proportional to the noise S/N. To mitigate the noise effects, one can change the objective function from the ℓ^2 norm to the W_2 metric (Yang et al., 2018), which has proven to be more robust with respect to noise. One can also add regularization terms to the objective function, which is a common strategy to improve the stability of the inverse problem.

Least-squares reverse time migration

Our third example is to apply AA to LSRTM. We still use the Marmousi benchmark (Figure 1a) for illustration. The smooth background velocity is shown in Figure 5a. We locate 80 equally spaced wave sources (Ricker wavelet centered at 25 Hz) at 100 m below the air-water interface. The entire workflow is similar to the FWI experiment except for a different forward problem and a different target. The size of the velocity model is 151×461, and the spatial spacing is 20 m. The total recording time is 4 s. Figure 5b and 5c shows the true reflectivity model and one iteration of the RTM image, respectively. RTM provides a crude subsurface image with unbalanced illumination. After Laplacian filtering (Zhang and Sun, 2009), the migration artifacts are reduced, but the amplitude of the image is still incorrect, as seen by comparing the color bar of Figure 5c and 5d with the truth (Figure 5b). LSRTM aims to refine the image obtained by conventional RTM toward the true reflectivity. Therefore, we use the image obtained by conventional RTM (Figure 5c) as the initial guess for inversion tests under the following four optimization methods: restarted GMRES, AA, L-BFGS, and steepest descent.

Because GMRES is good at finding the solution for square linear systems, we reformulate the linear inverse problem that LSRTM aims to solve,

$$Lm_r = d_r. \tag{20}$$

We multiply both sides of equation 20 by the Born operator L and obtain

$$L^T L m_r = L^T d_r. \tag{21}$$

Note the right side of equation 21 is nothing new but the migrated image after one step of RTM, which we denote as m_{RTM}. Because $L^T L$ is a symmetric square matrix, we obtain a square linear system

$$A^L m_r = m_{\text{RTM}}, \tag{22}$$

where $A^L = L^T L$. Thus, we can use GMRES to find the solution of equation 22, which is also the solution of the original problem in equation 20. We use a restarted GMRES with memory parameter $\mathcal{M} = 3$. Therefore, at most three previous iterates are stored in memory when building up the Krylov space at each iteration. We further discuss the motivation and compare GMRES with AA in the next section. The final solution using the restarted GMRES is shown in Figure 6a after 20 iterations.

The inverse problem or large-scale linear system that GMRES solves is an optimization-free formulation (see equation 22). Next, we return to the optimization formulation of the linear inverse problem (equation 12) and we use gradient-based methods to find the optimal m_r^*. Again, AA is applied to the steepest-descent algorithm following Algorithm 2. Figure 6 shows the inversion results after 20 iterations. AA obtains an equally good image with the one by L-BFGS (see Figure 6b and 6c). The memory parameter for both methods is $\mathcal{M} = 3$. It indicates that at most three iterates and their gradient vectors are stored in memory to compute the next iteration. AA also demonstrates noticeable improvement in LSRTM compared to the steepest-descent method, as shown in Figure 6d. All four methods improve the unbalanced illumination in the original RTM image (Figure 5c) and are closer to the true reflectivity (Figure 5b) compared to the filtered RTM image (Figure 5d).

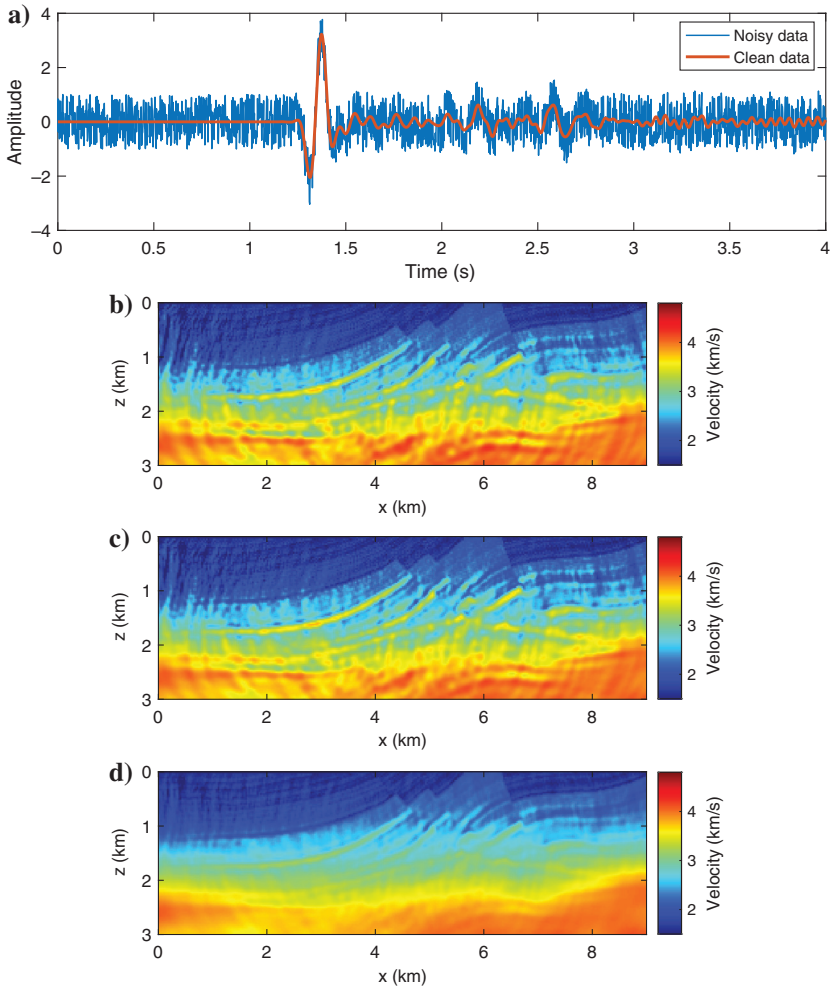

Figure 4. (a) A comparison between one trace of noisy and clean signals from the observed data, (b) noise inversion using AA after 1000 gradient evaluations, (c) noise inversion using L-BFGS after 1000 gradient evaluations, and (d) noise inversion using the steepest descent after 1000 gradient evaluations.

DISCUSSION

AA is a strategy proposed to speed up iterative schemes, particularly fixed-point problems. In this paper, we modify the classic algorithm and apply it to accelerate iterative optimization algorithms, such as the gradient-descent method. AA has intrinsic connections with GMRES and L-BFGS when solving specific types of problems. Therefore, we devote this section to detailed discussions and analysis regarding their differences and the potential benefits of AA. The ultimate goal is not to promote one method over another but to better understand their roles in optimization problems.

Anderson acceleration and GMRES

GMRES is known as an iterative method to solve the square nonsymmetric linear system (Saad, 2003)

$$Ax = b, \quad A \in \mathbb{C}^{n\times n}. \tag{23}$$

We define the kth Krylov subspace for this problem as

$$K_k = K_k(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}, \tag{24}$$

where $r_0 = Ax_0 - b$ and x_0 is the initial guess. GMRES approximates the exact solution of $Ax = b$ by choosing the kth iterate x_k in the Krylov subspace K_k such that

$$x_k = \text{argmin}_{x\in K_k}\|Ax_k - b\|_2, \tag{25}$$

which minimizes the Euclidean norm of the residual $r_k = Ax_k - b$. Because the current Krylov subspace is contained in the next subspace,

$$K_k = \text{span}\{x_0, x_1, \ldots, x_{k-1}\} \subseteq K_{k+1} = \text{span}\{x_0, x_1, \ldots, x_{k-1}, x_k\}, \tag{26}$$

the residual r_k monotonically decreases as the number of iterations increases. There have been numerous variations and extensions of the method over the decades (Saad, 2003).

In practice, it is not feasible to store all of the previous iterates due to the limited machine memory. Instead, after $\mathcal{M}$ iterations, one can treat the iterate $x_{\mathcal{M}}$ as the new initial guess x_0, and construct the new sequence of the Krylov subspace following equation 24. The method is well-known as restarted GMRES and is often denoted as GMRES($\mathcal{M}$). The restarted GMRES may suffer from stagnation in convergence because the restarted subspace is often close to the earlier subspace for matrix A with certain structures (Embree, 2003). Different from restarted GMRES, AA saves memory by replacing the oldest iteration x_0 by the latest iterate $x_{\mathcal{M}}$ if both methods share the same memory parameter $\mathcal{M}$ and are applied to solve the square

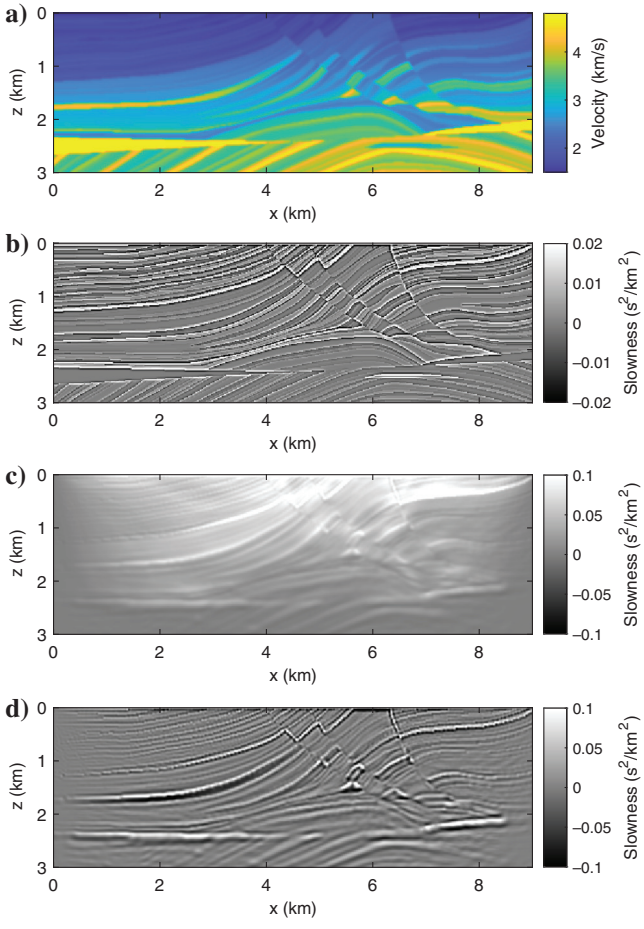

Figure 5. (a) The smooth background velocity m_0 for RTM and LSRTM, (b) the true reflectivity, (c) the original RTM result, and (d) the RTM image after Laplacian filtering.

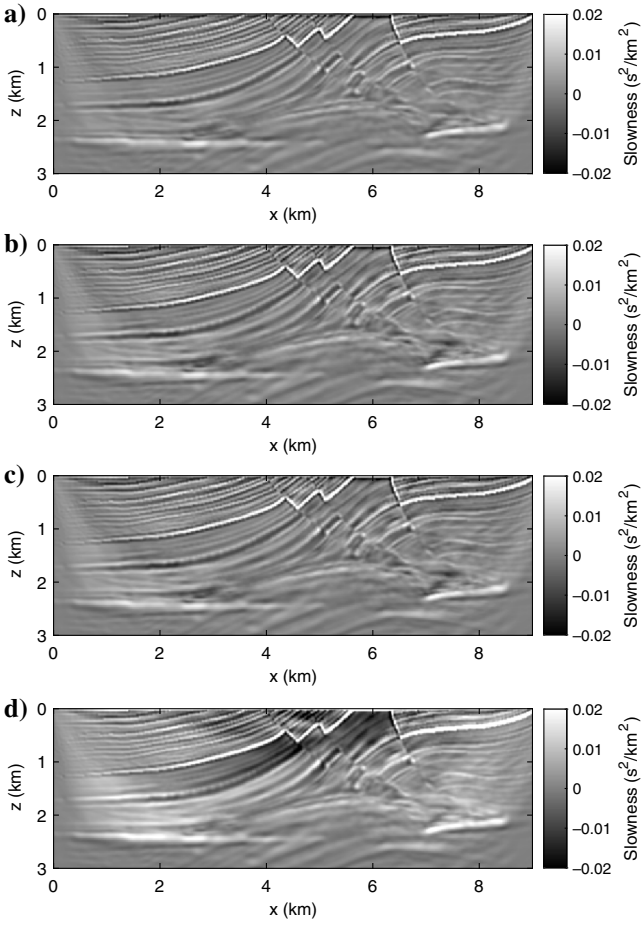

Figure 6. (a) LSRTM using restarted GMRES, (b) LSRTM using AA, (c) LSRTM using L-BFGS, and (d) LSRTM using the steepest descent.

linear system (equation 23). The main differences between the two methods are illustrated as follows:

$$\mathrm{GMRES}(\mathcal{M}): \underbrace{\mathrm{span}\{x_0, x_1, \ldots, x_{\mathcal{M}-1}\}}_{\text{the } \mathcal{M}\text{-th subspace}} \rightarrow \underbrace{\mathrm{span}\{x_{\mathcal{M}}\}}_{\text{the next subspace}}, \tag{27}$$

$$\mathrm{AA}(\mathcal{M}): \underbrace{\mathrm{span}\{x_0, x_1, \ldots, x_{\mathcal{M}-1}\}}_{\text{the } \mathcal{M}\text{-th subspace}} \rightarrow \underbrace{\mathrm{span}\{x_1, x_2, \ldots, x_{\mathcal{M}}\}}_{\text{the next subspace}}. \tag{28}$$

GMRES have been used in geophysical applications as a method of solving *forward* problems, which are also linear PDEs (Erlangga and Herrmann, 2008; Calandra et al., 2012). One should note that the direct connections between AA and GMRES only apply when the fixed-point operator G is based on the *square linear problem*, as shown in equation 23. To date, there has been no proven equivalence between AA applied to nonlinear operators and nonlinear GMRES. There have been numerical comparisons between AA of depth $\mathcal{M}$ and GMRES($\mathcal{M}$) in the literature (Pratapa et al., 2016; Yang et al., 2020). Empirically, AA with memory parameter $\mathcal{M}$ is observed to be more efficient than GMRES($\mathcal{M}$) for certain nonlinear problems or linear problems with a nonpositive definite matrix.

In FWI, we solve a nonlinear problem $F(x) = d$, where F is the forward wave operator and d is the observed data. Building Krylov spaces for such large-scale applications, which is theoretically an infinite-dimensional inverse problem, can be extremely costly. For LSRTM, we are solving a linear problem $Lm_r = d_r$, where L is the Born operator and d_r is the observed scattering data. However, after discretization, matrix L has more rows than columns. GMRES is not the best method for solving the linear system directly, and LSQR could be a better alternative. Therefore, we reformulate the problem of LSRTM so that it is suitable for GMRES. The results are presented in Figure 6. We remark that using AA for LSRTM accelerates the steepest-descent algorithm, but GMRES for LSRTM, as it is done in this paper, is an optimization-free implementation.

Anderson acceleration and L-BFGS

A standard optimization method used in geophysics is L-BFGS, where "L" indicates a variant of the BFGS algorithm with limited memory. We also have used this method for inversion tasks in the previous section. It belongs to the class of quasi-Newton methods, which is preferred when the full Newton's method is too time-consuming to apply. The Hessian matrix of a quasi-Newton method does not need to be computed explicitly at every iteration. Instead, an approximation B_k, which satisfies the following inverse secant condition, is used instead of the true inverse Hessian at the kth iteration:

$$B_k(J(p_k) - J(p_{k-1})) = p_k - p_{k-1}. \tag{29}$$

Here, p_{k-1} and p_k are two consecutive iterates and J is the objective function that we aim to minimize. Quasi-Newton methods differ among each other in how to update B_k, the approximation to the inverse Hessian matrix. The BFGS algorithm follows two principles: (1) satisfy the inverse secant condition in equation 29 and (2) be as close as possible to the approximation at the previous iteration. The latter is translated as

$$B_k = \underset{B \in \mathbb{C}^{n\times n}}{\mathrm{argmin}} \|B - B_{k-1}\|_F^2, \tag{30}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm, i.e., $\|A\|_F^2 = \sum_{i,j=1}^{n} |A_{ij}|^2$. These conditions lead to explicit update schemes for BFGS and L-BFGS as a result of the famous Sherman-Morrison-Woodbury formula (Nocedal and Wright, 2006). After replacing the inverse Hessian matrix in Newton's method with the approximation B_k, the next iterate of the optimization problem is

$$p_{k+1} = p_k - B_k f_k = p_k + \eta B_k \mathcal{G}_k, \tag{31}$$

where $\mathcal{G}_k$ denotes the gradient of the objective function $J(p)$ at $p = p_k$.

Whereas the BFGS algorithm is a type of secant method, it is worth addressing that AA is equivalent to a *multisecant* method (Fang and Saad, 2009). If one chooses the ℓ^2-based AA regarding equation 1 as we do in this paper, then the update formula (equation 2) with $\beta_k = 1$ can be rewritten as

$$p_{k+1} = p_k - S_k f_k = p_k + \eta S_k \mathcal{G}_k, \tag{32}$$

where $S_k \in \mathbb{C}^{n\times n}$ is the solution to the following constrained optimization problem:

$$S_k = \underset{S \in \mathbb{C}^{n\times n}}{\mathrm{argmin}} \|S + I\|_F^2, \tag{33}$$

subject to the multisecant condition

$$S_k D_k = P_k. \tag{34}$$

Denoting $\Delta p_i = p_{i+1} - p_i$, $\Delta f_i = f_{i+1} - f_i$, matrices P_k and D_k are defined, respectively, as

$$\begin{aligned} P_k &= [\Delta p_{k-\mathcal{M}}, \ldots, \Delta p_{k-1}] \in \mathbb{C}^{n\times\mathcal{M}}, \\ D_k &= [\Delta f_{k-\mathcal{M}}, \ldots, \Delta f_{k-1}] \in \mathbb{C}^{n\times\mathcal{M}}. \end{aligned} \tag{35}$$

We have used the fact that $f_k = G(p_k) - p_k = -\eta \mathcal{G}_k$ in equation 32.

Once we have rewritten the AA algorithm, it is not hard to recognize the similarities between the update formula of AA and that of the L-BFGS method (see equations 31 and 32). The key difference is that matrices S_k and B_k are constructed under different principles, although both can be regarded as approximations to the inverse Hessian matrix in the full Newton's method. For L-BFGS, B_k satisfies the secant condition (equation 29), while minimizing equation 30. For AA, S_k satisfies the *multisecant* condition (equation 34) while minimizing equation 33. Both methods are faster than the steepest-descent algorithm and have been proven to have superlinear convergence.

The original BFGS algorithm stores a dense n-by-n approximation to the inverse Hessian matrix, where n is the number of variables. Besides, each BFGS iteration has a cost of $\mathcal{O}(n^2)$ arithmetic operations. The idea of L-BFGS is to restrict the use of all iterations in the history to the latest $\mathcal{M}$ iterates; $\mathcal{M}$ is a parameter of the

L-BFGS algorithm that can be chosen a priori. Because the earlier iterates often carry little information about the curvature of the current iterate, the change from BFGS to L-BFGS is expected to have minimal effects on the convergence rate. The L-BFGS method has a linear memory requirement in terms of the number of variables. It is particularly well-suited for large-scale optimization problems such as FWI and LSRTM.

Similar to AA, L-BFGS maintains a history of the past $\mathcal{M}$ iterates and their gradients. Again, $\mathcal{M}$ is often chosen to be small. The connections are illustrated by the following diagram where $\mathcal{G}_k$ denotes the gradient of the objective function at $p = p_k$:

$$\begin{aligned}\text{L-BFGS}(\mathcal{M}):&\underbrace{\{p_0,\ \ldots,\ p_{\mathcal{M}-1},\mathcal{G}_0,\ \ldots,\ \mathcal{G}_{\mathcal{M}-1}\}}_{\text{construct } B_{\mathcal{M}-1}}\\ \rightarrow&\underbrace{\{p_1,\ \ldots,\ p_{\mathcal{M}},\mathcal{G}_1,\ \ldots,\ \mathcal{G}_{\mathcal{M}}\}}_{\text{construct } B_{\mathcal{M}}},\end{aligned}\tag{36}$$

$$\begin{aligned}\text{AA}(\mathcal{M}):&\underbrace{\{p_0,\ \ldots,\ p_{\mathcal{M}-1},\mathcal{G}_0,\ \ldots,\ \mathcal{G}_{\mathcal{M}-1}\}}_{\text{construct } S_{\mathcal{M}-1}}\\ \rightarrow&\underbrace{\{p_1,\ \ldots,\ p_{\mathcal{M}},\mathcal{G}_1,\ \ldots,\ \mathcal{G}_{\mathcal{M}}\}}_{\text{construct } S_{\mathcal{M}}}.\end{aligned}\tag{37}$$

We remark that the relationship above is valid only when AA is used to accelerate the steepest-descent algorithm as what we propose in this paper. The most expensive part of the L-BFGS algorithm is the inverse Hessian update, and the most costly step of AA is to compute the optimal coefficients. In terms of computational cost, the low-rank QR update for AA, the inverse Hessian update for L-BFGS, and the restarted GMRES all take $\mathcal{O}(n\mathcal{M})$ floating-point operations (flops), if implemented optimally. The minimum memory requirements for all three methods are also $\mathcal{O}(n\mathcal{M})$, considering that the memory parameter for all of the methods is $\mathcal{M}$ and the size of the unknown is n.

Performance comparison

We have seen the theoretical connections among AA, the restarted GMRES, and the L-BFGS algorithm in the previous two subsections. Although under the same memory parameter $\mathcal{M}$, all of the methods have the same order of computational cost and memory requirements, AA could be advantageous in the following two aspects.

First, when the iteration number is bigger than the memory parameter $\mathcal{M}$, AA always uses the last $\mathcal{M}$ iterates to construct the solution for the next iteration. On the other hand, the restarted GMRES nullifies all of the $\mathcal{M}$ iterates and restarts from zero once the restart window is reached. Thus, AA can use more information in the optimization than the restarted GMRES in most iterations. The advantage is reflected in our numerical examples for LSRTM (see Figure 6a and 6b).

One can only compare AA and GMRES when both methods are used to find the solution x for problems in the form of $Ax = b$, where A is a square matrix. The flexibility of GMRES reduces for rectangle matrices and highly nonlinear problems, whereas AA can be applied to all types of linear and nonlinear problems as long as they are written as fixed-point operators. One study has shown that the fixed-point operator does not need to be a contraction for AA to converge, although it is necessary for Picard iteration (Pollock and Rebholz, 2019).

Second, unlike the restarted GMRES, the L-BFGS algorithm uses all of the last $\mathcal{M}$ iterates to compute the next iteration. Although AA and L-BFGS exploit all of the information available in storage, the two methods approximate the inverse Hessian matrix in slightly different ways: the former is a *multisecant* method, whereas the latter is a secant method. We observe in Figure 2a and 2b that FWI using AA spends less time searching for an appropriate step size than the inversion using the L-BFGS method on average for each iteration. It helps AA achieve better inversion results than L-BFGS under the same number of total gradient evaluations. The inverse Hessian matrix approximated by AA satisfies the secant condition not only for the latest iteration but also for the previous $\mathcal{M}$ iterations. This implicitly enforces the connections among the neighbor iterates to avoid unstable descent directions far from the curvature of the basin of attraction.

Although it is expected that the bigger the memory parameter $\mathcal{M}$ for AA, the better the performance, empirical studies have shown that a relatively small $\mathcal{M}$, commonly ranging from 3 to 20, is often good enough to speed up the convergence of the fixed-point iteration without a significant toll on the machine memory and the computational cost. In the experiments, the choice of the memory parameter for AA could follow similar principles as choosing the memory depth for L-BFGS and the restarted window for GMRES.

CONCLUSION

AA for seismic inversion treats the method of steepest descent as a fixed-point operator. It speeds up the convergence by linearly combining a list of the previous iterates in an optimized way. The computational cost of implementing AA mainly comes from the 1D optimization for the weights. It is thus easy to add AA to existing optimization algorithms. As shown in this paper, AA outperforms the steepest-descent method and also can be considered an alternative to L-BFGS. AA is equivalent to a *multisecant* method, whereas the L-BFGS algorithm is derived by the secant method. Being computationally attractive, AA is an efficient optimization algorithm for FWI and LSRTM.

ACKNOWLEDGMENTS

The author thanks A. Richardson and other anonymous reviewers for constructive suggestions for the paper. This material is based upon work supported by the National Science Foundation under award number DMS-1913129.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be obtained by contacting the corresponding author.

REFERENCES

An, H., X. Jia, and H. F. Walker, 2017, Anderson acceleration and application to the three-temperature energy equations: Journal of Computational Physics, **347**, 1–19, doi: 10.1016/j.jcp.2017.06.031.

Anderson, D. G., 1965, Iterative procedures for nonlinear integral equations: Journal of the ACM, **12**, 547–560, doi: 10.1145/321296.321305.

Butenko, S., and P. M. Pardalos, 2014, Numerical methods and optimization: An introduction: CRC Press.

Calandra, H., S. Gratton, J. Langou, X. Pinel, and X. Vasseur, 2012, Flexible variants of block restarted GMRES methods with application to geophysics: SIAM Journal on Scientific Computing, **34**, A714–A736, doi: 10.1137/10082364X.

Ceniceros, H., and G. Fredrickson, 2004, Numerical solution of polymer self-consistent field theory: Multiscale Modeling & Simulation, **2**, 452–474, doi: 10.1137/030601338.
Dai, W., and G. T. Schuster, 2013, Plane-wave least-squares reverse-time migration: Geophysics, **78**, no. 4, S165–S177, doi: 10.1190/geo2012-0377.1.
Embree, M., 2003, The tortoise and the hare restart GMRES: SIAM Review, **45**, 259–266, doi: 10.1137/S003614450139961.
Engquist, B., and Y. Yang, 2020, Optimal transport based seismic inversion: Beyond cycle skipping: arXiv preprint arXiv:2002.00031.
Erlangga, Y. A., and F. J. Herrmann, 2008, An iterative multilevel method for computing wavefields in frequency-domain seismic inversion: 78th Annual International Meeting, SEG, Expanded Abstracts, 1956–1960, doi: 10.1190/1.3059279.
Evans, C., S. Pollock, L. G. Rebholz, and M. Xiao, 2020, A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically): SIAM Journal on Numerical Analysis, **58**, 788–810, doi: 10.1137/19M1245384.
Fang, H.-R., and Y. Saad, 2009, Two classes of multisecant methods for nonlinear acceleration: Numerical Linear Algebra with Applications, **16**, 197–221, doi: 10.1002/nla.617.
Fu, A., J. Zhang, and S. Boyd, 2019, Anderson accelerated Douglas–Rachford splitting: arXiv preprint arXiv:1908.11482.
Geist, M., and B. Scherrer, 2018, Anderson acceleration for reinforcement learning: arXiv preprint arXiv:1809.09501.
Golub, G., and C. Van Loan, 2013, Matrix computations: Johns Hopkins University Press.
Hudson, J., and J. Heritage, 1981, The use of the Born approximation in seismic scattering problems: Geophysical Journal International, **66**, 221–240, doi: 10.1111/j.1365-246X.1981.tb05954.x.
Kudin, K. N., G. E. Scuseria, and E. Cances, 2002, A black-box self-consistent field convergence algorithm: One step closer: The Journal of Chemical Physics, **116**, 8255–8261, doi: 10.1063/1.1470195.
Li, Z., and J. Li, 2018, A fast Anderson-Chebyshev mixing method for nonlinear optimization: arXiv preprint arXiv:1809.02341.
Mai, V. V., and M. Johansson, 2019, Anderson acceleration of proximal gradient methods: arXiv preprint arXiv:1910.08590.
Métivier, L., R. Brossier, S. Operto, and J. Virieux, 2017, Full waveform inversion and the truncated Newton method: SIAM Review, **59**, 153–195, doi: 10.1137/16M1093239.
Nocedal, J., and S. Wright, 2006, Numerical optimization: Springer Science & Business Media.
Peng, Y., B. Deng, J. Zhang, F. Geng, W. Qin, and L. Liu, 2018, Anderson acceleration for geometry optimization and physics simulation: ACM Transactions on Graphics, **37**, 1–14, doi: 10.1145/3197517.3201290.
Picard, E., 1893, Sur l'application des méthodes d'approximations successives à l'étude de certaines équations différentielles ordinaires: Journal de Mathématiques Pures et Appliquées, **9**, 217–272.
Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503, doi: 10.1111/j.1365-246X.2006.02978.x.
Pollock, S., and L. Rebholz, 2019, Anderson acceleration for contractive and noncontractive operators: arXiv preprint arXiv:1909.04638.
Pollock, S., L. G. Rebholz, and M. Xiao, 2018, Anderson-accelerated convergence of Picard iterations for incompressible Navier–Stokes equations: SIAM Journal on Numerical Analysis, **57**, 615–637, doi: 10.1137/18M1206151.
Pratapa, P. P., P. Suryanarayana, and J. E. Pask, 2016, Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems: Journal of Computational Physics, **306**, 43–54, doi: 10.1016/j.jcp.2015.11.018.
Pulay, P., 1980, Convergence acceleration of iterative sequences. The case of SCF iteration: Chemical Physics Letters, **73**, 393–398, doi: 10.1016/0009-2614(80)80396-4.
Saad, Y., 2003, Iterative methods for sparse linear systems: SIAM.
Symes, W. W., 2020, Full waveform inversion by source extension: Why it works: arXiv preprint arXiv:2003.12538.
Tarantola, A., and B. Valette, 1982, Generalized nonlinear inverse problems solved using the least squares criterion: Reviews of Geophysics, **20**, 219–232, doi: 10.1029/RG020i002p00219.
Toth, A., and C. Kelley, 2015, Convergence analysis for Anderson acceleration: SIAM Journal on Numerical Analysis, **53**, 805–819, doi: 10.1137/130919398.
Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, no. 6, WCC1–WCC26, doi: 10.1190/1.3238367.
Walker, H. F., and P. Ni, 2011, Anderson acceleration for fixed-point iterations: SIAM Journal on Numerical Analysis, **49**, 1715–1735, doi: 10.1137/10078356X.
Yang, Y., B. Engquist, J. Sun, and B. F. Hamfeldt, 2018, Application of optimal transport and the quadratic Wasserstein metric to full-waveform inversion: Geophysics, **83**, no. 1, R43–R62, doi: 10.1190/geo2016-0663.1.
Yang, Y., A. Townsend, and D. Appelö, 2020, Anderson acceleration using the H^{-s} norm: arXiv preprint arXiv:2002.03694.
Zhang, J., B. O'Donoghue, and S. Boyd, 2018, Globally convergent type-I Anderson acceleration for non-smooth fixed-point iterations: arXiv preprint arXiv:1808.03971.
Zhang, Y., and J. Sun, 2009, Practical issues in reverse time migration: True amplitude gathers, noise removal and harmonic source encoding: First Break, **27**, 53–59, doi: 10.3997/1365-2397.2009002.

A biography and photograph of the author are not available.