Journal of Computational Physics 296 (2015) 58–71

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A frozen Gaussian approximation-based multi-level particle swarm optimization for seismic inversion

Jinglai Li [a,*], Guang Lin [b,c], Xu Yang [d]

[a] Institute of Natural Sciences, Department of Mathematics, and MOE Key Laboratory of Scientific and Engineering Computing, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Department of Mathematics, School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA
[c] Computational Sciences and Mathematics Division, Pacific Northwest National Laboratory, Richland, WA 99352, USA
[d] Department of Mathematics, University of California, Santa Barbara, CA 93106, USA

ARTICLE INFO

Article history:
Received 19 October 2014
Received in revised form 18 April 2015
Accepted 30 April 2015
Available online 6 May 2015

Keywords:
Frozen Gaussian approximation
Full waveform inversion
High-frequency wave
Particle swarm optimization

ABSTRACT

In this paper, we propose a frozen Gaussian approximation (FGA)-based multi-level particle swarm optimization (MLPSO) method for seismic inversion of high-frequency wave data. The method addresses two challenges in it: First, the optimization problem is highly non-convex, which makes hard for gradient-based methods to reach global minima. This is tackled by MLPSO which can escape from undesired local minima. Second, the character of high-frequency of seismic waves requires a large number of grid points in direct computational methods, and thus renders an extremely high computational demand on the simulation of each sample in MLPSO. We overcome this difficulty by three steps: First, we use FGA to compute high-frequency wave propagation based on asymptotic analysis on phase plane; Then we design a constrained full waveform inversion problem to prevent the optimization search getting into regions of velocity where FGA is not accurate; Last, we solve the constrained optimization problem by MLPSO that employs FGA solvers with different fidelity. The performance of the proposed method is demonstrated by a two-dimensional full-waveform inversion example of the smoothed Marmousi model.

1. Introduction

In seismic inversion, full-waveform inversion (FWI) [28,41] is a promising technique to reconstruct subsurface velocity profiles from seismograms. The method becomes increasingly popular thanks to its ability to produce high resolution images of the velocity profile. FWI is usually cast as minimizing the misfit between the collected data and prediction provided by the simulation of seismic wave propagation.

Despite the rather simple formulation, computation of FWI problems is rather challenging. One of the major difficulties lie in that the minimization problem is highly non-convex [37]: most gradient based optimization techniques rely on good initial guesses to reach global minima, and unfortunately such good initial guesses are often not available in practical problems. In fact, usually one first finds an approximate FWI solution with global optimization techniques and then improves it with gradient-based optimization methods. Particularly, stochastic global optimization techniques, such as simulated annealing (SA) [35,39], genetic algorithm (GA) [33,36], and more recently, particle swarm optimization (PSO) [34], have been

* Corresponding author.
E-mail addresses: jinglaili@sjtu.edu.cn (J. Li), lin491@purdue.edu (G. Lin), xuyang@math.ucsb.edu (X. Yang).

http://dx.doi.org/10.1016/j.jcp.2015.04.050
0021-9991/

applied to seismic inversion problems due to their abilities of escaping from undesired local minima. However, stochastic optimization methods usually require a large number of repeated simulations of wave propagation, and each simulation involves waves of high-frequency, whose wavelengths are extremely short compared to the domain size of interest. As a result, direct simulations of high-frequency wave propagation, e.g., finite difference/volume/element methods, can be extremely computationally expensive. Thus performing a global optimization with direct simulations of high-frequency waves is nearly prohibitive.

Alternative approaches to reduce the computational cost are to look for approximate solutions to wave equation based on semiclassical approximation, among which, the ray theory [2,31,6] provides attractive alternatives. In the ray-based approaches, one decomposes wavefields into elementary waveforms which propagate along rays, and reconstructs wavefields based on the dynamic information on rays (e.g., path trajectory, amplitude and phase). Kirchhoff migration [10,21] and Gaussian beam migration [15,16,27,11,12,30] are famous seismic migration methods of this kind. Kirchhoff migration greatly increases the computational speed with an asymptotic error proportional to the ratio of wavelength over the domain size, but it yields unbounded amplitudes at caustics. Gaussian beam migration keeps the merits of ray tracing, but also handles multipathing as well as maintain accuracy at caustics. However, Gaussian beam migration relies on the Taylor expansion around the central ray, hence the error of the approximation increases when the beams become wide; see Example 4.2 in [23] for a numerical study of this case. One needs to tune the width parameter of Gaussian beams in order to get a good resolution, especially when the wave solution spreads over time [3,15,7]. This is practically difficult due to the heterogeneity of the media and the non-linearity of the Riccati equation involved in the beam construction.

Recently, the frozen Gaussian approximation (FGA) method [42] was developed for seismic modeling in complex structures. The method was originally motivated by Herman–Kluk propagator for solving the Schrödinger equation in quantum chemistry [14,19,20], and later generalized to linear strictly hyperbolic systems [23–25]. The main idea of FGA is to use Gaussian functions with fixed widths to approximate the solution of wave equation. These Gaussian functions are also called coherent states in quantum mechanics, which was previously applied in seismic imaging [1,9] but did not have the rigorous treatment of amplitude factors given by FGA. Compared to Gaussian beam migration, FGA can provide a more accurate and robust solution, especially in the situation of wave spreading [23,24]. The main procedure of FGA is described as following. The initial data are decomposed into a sum of Gaussians with fixed (small) widths. Then one propagates each Gaussian function along geometric rays. The amplitudes of these Gaussians are given according to rigorously-derived dynamic equations so that the sum of them produces a good approximation to the solution of wave equation at the final time.

As an asymptotic solution to the wave equation, the accuracy of FGA is derived in [24]. Nevertheless, the actual asymptotic error of FGA depends on the smoothness of velocity profiles and it loses accuracy in certain cases (e.g. discontinuous media), which will lead to erratic inversion results. In practice, it is desirable to restrict the search within the region of smooth velocities where FGA provides an accurate approximation, however such a region cannot be easily identified *a priori*. In this work, we introduce a constraint in the original FWI problem to prevent the solution moving out of the FGA-valid region. This will add proper smoothing effects, and yield a smoothed velocity profile for the original FWI problem. It is also consistent with the original problem when the wave equation is solved exactly.

In principle all the aforementioned global optimization techniques such as SA and GA can be used to solve the constrained optimization problem. Here we choose the PSO method for its reported superior computational efficiency in various applications [13]. Another issue in optimization is the computational cost of FGA. As will be shown in Section 2, the accuracy of FGA depends on the initial approximation error, which in turn can only be reduced by increasing the number of Gaussians in the initial decomposition. Consequently, the cost of computing a highly accurate FGA solution can be rather high (though still much lower than direction simulations). On the other hand, at the early stage of the optimization, high-accuracy solutions may not be necessary, which motivates the idea to start with a less accurate FGA solver and gradually increase its precision as the optimization proceeds. Accordingly, we propose a multi-level particle swarm optimization (MLPSO) algorithm that employs FGA solvers with different fidelity (namely, different number of beams) to further improve the computational efficiency.

In summary, the main contribution of the work is threefold: we propose to use FGA to accelerate the computation of high-frequency FWI problems; we design a constrained optimization problem to prevent the solution getting into the region where FGA is not accurate; we develop an MLPSO algorithm to efficiently solve the resulting optimization problem.

The rest of this paper is organized as follows. In Section 2, we describe the FGA algorithm for the computation high-frequency wave propagation. We introduce the constrained FWI optimization problem in Section 3. In Section 4, we present the MLPSO algorithm to solve the constrained problem. The performance of our method is demonstrated in Section 5 by a two-dimensional example of smoothed Marmousi model. Finally, we make conclusive remarks in Section 6.

2. Frozen Gaussian approximation

We consider a high-frequency acoustic wave equation in d-dimension:

$$\frac{\partial^2 u}{\partial t^2} - \xi(\boldsymbol{x})\,\Delta u = 0, \tag{2.1}$$

with the prescribed initial conditions:

$$u(0,\boldsymbol{x}) = U_0^\varepsilon(\boldsymbol{x}), \tag{2.2a}$$

$$\frac{\partial u}{\partial t} = U_1^\varepsilon(\boldsymbol{x}), \tag{2.2b}$$

where t and $\boldsymbol{x}$ are the time and multidimensional space variables respectively, u is the wavefield, $\xi(\boldsymbol{x}) = c^2(\boldsymbol{x}) > 0$ is the square of a velocity profile, and ε indicates the scale of wavelength. We assume all variables are in dimensionless units, and $\varepsilon \ll 1$ corresponds to the high-frequency (short wavelength) regime.

Next we introduce how to obtain an FGA solution to Eq. (2.1). Note that the presentation of FGA here has been tailored and simplified for the purposes of this work. For more details, such as the asymptotic derivation, the error estimates, the validity at caustics, and generalization to other strictly hyperbolic systems, we refer the readers to [23–25,42].

2.1. FGA formulation

The ansatz of FGA is taken to be a sum of Gaussian functions with a fixed width,

$$\begin{aligned} u_{\mathrm{FGA}}(t,\boldsymbol{x}) = &\sum_{(\boldsymbol{q},\boldsymbol{p})\in G_+} \frac{a_+\psi_+^\varepsilon}{(2\pi\varepsilon)^{3d/2}} e^{\frac{\mathrm{i}}{\varepsilon}\boldsymbol{P}_+\cdot(\boldsymbol{x}-\boldsymbol{Q}_+)-\frac{1}{2\varepsilon}|\boldsymbol{x}-\boldsymbol{Q}_+|^2} \\ &+ \sum_{(\boldsymbol{q},\boldsymbol{p})\in G_-} \frac{a_-\psi_-^\varepsilon}{(2\pi\varepsilon)^{3d/2}} e^{\frac{\mathrm{i}}{\varepsilon}\boldsymbol{P}_-\cdot(\boldsymbol{x}-\boldsymbol{Q}_-)-\frac{1}{2\varepsilon}|\boldsymbol{x}-\boldsymbol{Q}_-|^2}, \end{aligned} \tag{2.3}$$

where

$$\psi_\pm^\varepsilon(\boldsymbol{q},\boldsymbol{p}) = \int_{\mathbb{R}^d} u_{\pm,0}^\varepsilon(\boldsymbol{y},\boldsymbol{q},\boldsymbol{p}) e^{-\frac{\mathrm{i}}{\varepsilon}\boldsymbol{p}\cdot(\boldsymbol{y}-\boldsymbol{q})-\frac{1}{2\varepsilon}|\boldsymbol{y}-\boldsymbol{q}|^2}\,\mathrm{d}\boldsymbol{y}, \tag{2.4}$$

$$u_{\pm,0}^\varepsilon(\boldsymbol{y},\boldsymbol{q},\boldsymbol{p}) = \frac{1}{2}\left(U_0^\varepsilon(\boldsymbol{y}) \pm \frac{\mathrm{i}\varepsilon}{\xi(\boldsymbol{q})|\boldsymbol{p}|} U_1^\varepsilon(\boldsymbol{y})\right). \tag{2.5}$$

In (2.3), $\mathrm{i} = \sqrt{-1}$ is the imaginary unit, and "+" and "−" indicate the two wave branches, and $G_\pm$ are the sets of $(\boldsymbol{q},\boldsymbol{p})$ pairs. In FGA, the weight function $\psi_\pm$ (2.4) is in a form of FBI transform [26], *time-independent* and computed initially; the *time-dependent* quantities are: Center $\boldsymbol{Q}_\pm$, momentum $\boldsymbol{P}_\pm$ and amplitude $a_\pm$; the width of Gaussian function is fixed at all time.

The evolution of $\boldsymbol{Q}_\pm(t,\boldsymbol{q},\boldsymbol{p})$ and $\boldsymbol{P}_\pm(t,\boldsymbol{q},\boldsymbol{p})$ satisfies the ray tracing equations associated with the Hamiltonian $H_\pm = \pm\sqrt{\xi(\boldsymbol{Q}_\pm)}|\boldsymbol{P}_\pm|$. For simplicity, we will omit the subscripts "±" without any confusion. Then $(\boldsymbol{Q},\boldsymbol{P})$ follows

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{Q}}{\mathrm{d}t} = \partial_{\boldsymbol{P}} H, \\ \dfrac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}t} = -\partial_{\boldsymbol{Q}} H, \end{cases} \tag{2.6}$$

with the initial conditions

$$\boldsymbol{Q}(0,\boldsymbol{q},\boldsymbol{p}) = \boldsymbol{q} \quad \text{and} \quad \boldsymbol{P}(0,\boldsymbol{q},\boldsymbol{p}) = \boldsymbol{p}. \tag{2.7}$$

The amplitude $a(t,\boldsymbol{q},\boldsymbol{p})$ solves

$$\frac{\mathrm{d}a}{\mathrm{d}t} = a\frac{\partial_{\boldsymbol{P}} H\cdot\partial_{\boldsymbol{Q}} H}{H} + \frac{a}{2}\,\mathrm{tr}\left(Z^{-1}\frac{\mathrm{d}Z}{\mathrm{d}t}\right), \tag{2.8}$$

with initial condition $a(0,\boldsymbol{q},\boldsymbol{p}) = 2^{d/2}$. In (2.8), the matrix Z is given by

$$\partial_{\boldsymbol{z}} = \partial_{\boldsymbol{q}} - \mathrm{i}\partial_{\boldsymbol{p}}, \qquad Z = \partial_{\boldsymbol{z}}(\boldsymbol{Q} + \mathrm{i}\boldsymbol{P}). \tag{2.9}$$

Here $\partial_{\boldsymbol{z}}\boldsymbol{Q}$ and $\partial_{\boldsymbol{z}}\boldsymbol{P}$ are defined as matrices, with the (j,k) component denoted as $\partial_{z_j}\boldsymbol{Q}_k$. The matrices $\partial_{\boldsymbol{z}}\boldsymbol{Q}$ and $\partial_{\boldsymbol{z}}\boldsymbol{P}$ can be computed at each time step by either divided difference or the following dynamic ray tracing equations,

$$\frac{\mathrm{d}(\partial_{\boldsymbol{z}}\boldsymbol{Q})}{\mathrm{d}t} = \partial_{\boldsymbol{z}}\boldsymbol{Q}\,\frac{\partial^2 H}{\partial\boldsymbol{Q}\,\partial\boldsymbol{P}} + \partial_{\boldsymbol{z}}\boldsymbol{P}\frac{\partial^2 H}{\partial\boldsymbol{P}^2}, \tag{2.10}$$

$$\frac{\mathrm{d}(\partial_{\boldsymbol{z}}\boldsymbol{P})}{\mathrm{d}t} = -\partial_{\boldsymbol{z}}\boldsymbol{Q}\,\frac{\partial^2 H}{\partial\boldsymbol{Q}^2} - \partial_{\boldsymbol{z}}\boldsymbol{P}\frac{\partial^2 H}{\partial\boldsymbol{P}\partial\boldsymbol{Q}}, \tag{2.11}$$

which can be also written in a componentwise form, with Einstein's index summation convention,

$$\frac{\mathrm{d}(\partial_{\boldsymbol{z}}\boldsymbol{Q})_{jk}}{\mathrm{d}t} = \partial_{\boldsymbol{z}_j}\boldsymbol{Q}_l \frac{\partial^2 H}{\partial \boldsymbol{Q}_l \partial \boldsymbol{P}_k} + \partial_{\boldsymbol{z}_j}\boldsymbol{P}_l \frac{\partial^2 H}{\partial \boldsymbol{P}_l \boldsymbol{P}_k},$$
$$\frac{\mathrm{d}(\partial_{\boldsymbol{z}}\boldsymbol{P})_{jk}}{\mathrm{d}t} = -\partial_{\boldsymbol{z}_j}\boldsymbol{Q}_l \frac{\partial^2 H}{\partial \boldsymbol{Q}_l \boldsymbol{Q}_k} - \partial_{\boldsymbol{z}_j}\boldsymbol{P}_l \frac{\partial^2 H}{\partial \boldsymbol{P}_l \partial \boldsymbol{Q}_k}.$$

Remark that the solution of dynamic ray tracing equations (2.10)–(2.11) in FGA only affects the amplitude a, while it affects both the amplitude and the beam width in Gaussian beam migration.

2.2. Initial wavefield decomposition

To compute (2.3) in FGA, one needs to choose proper sets $G_\pm$ of $(\boldsymbol{q}, \boldsymbol{p})$ pair in (2.3) and compute $\psi_\pm^\varepsilon$ in (2.4) correspondingly, i.e. to decompose the initial wavefield (2.2) into a sum of Gaussian functions. Here we use the local Fast FBI transform introduced in [42] to efficiently compute $\psi_\pm^\varepsilon$.

To obtain $\psi_\pm$, it is sufficient to know

$$\psi_j^\varepsilon(\boldsymbol{q}, \boldsymbol{p}) = \int_{\mathbb{R}^d} U_j^\varepsilon(\boldsymbol{y}) e^{-\frac{\mathrm{i}}{\varepsilon}\boldsymbol{p}\cdot(\boldsymbol{y}-\boldsymbol{q}) - \frac{1}{2\varepsilon}|\boldsymbol{y}-\boldsymbol{q}|^2}\,\mathrm{d}\boldsymbol{y}, \tag{2.12}$$

for $j = 0, 1$. Rewrite (2.12) by the change of variable $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{q}$,

$$\psi_j^\varepsilon(\boldsymbol{q}, \boldsymbol{p}) = \int_{\mathbb{R}^d} U_j^\varepsilon(\boldsymbol{q}+\boldsymbol{r}) e^{-\frac{\mathrm{i}}{\varepsilon}\boldsymbol{p}\cdot\boldsymbol{r} - \frac{1}{2\varepsilon}|\boldsymbol{r}|^2}\,\mathrm{d}\boldsymbol{r}. \tag{2.13}$$

Define

$$g_{\boldsymbol{q},j}^\varepsilon(\boldsymbol{r}) = U_j^\varepsilon(\boldsymbol{q}+\boldsymbol{r}) \exp(-\tfrac{1}{2\varepsilon}|\boldsymbol{r}|^2), \tag{2.14}$$

then ψ_j^ε is given by the (rescaled) Fourier transform of $g_{\boldsymbol{q},j}^\varepsilon$,

$$\psi_j^\varepsilon(\boldsymbol{q}, \boldsymbol{p}) = \widehat{g_{\boldsymbol{q},j}^\varepsilon}(\boldsymbol{p}/\varepsilon). \tag{2.15}$$

Notice that $g_{\boldsymbol{q},j}^\varepsilon$ contains an exponential function, hence its function value is negligible outside a localized domain centered around zero, for example, a small box

$$B_\varepsilon = [-L/2, L/2]^d \subset \mathbb{R}^d$$

with the length L scaled as $O(\sqrt{\varepsilon})$. Therefore, $\psi_j^\varepsilon(\boldsymbol{q}, \boldsymbol{p})$ can be evaluated efficiently by applying Fast Fourier Transform of $g_{\boldsymbol{q},j}^\varepsilon$ restricted on the small box B_ε.

Once $\psi_\pm^\varepsilon$ is obtained, we do a thresholding to get the sets $G_\pm$ where $\psi_\pm^\varepsilon$ have relatively large values.

2.3. Algorithm

The algorithm of frozen Gaussian approximation consists of three steps [42]:

(1) Initial decomposition: Choose the sets $G_\pm$ of $(\boldsymbol{q}, \boldsymbol{p})$ pair and calculate $\psi_\pm^\varepsilon$ defined in (2.4) from U_0^ε and U_1^ε;
(2) Time propagation: Numerically integrate (2.6) and (2.8) up to the final time T;
(3) Reconstruction: Compute the wavefield at time T by (2.3).

2.4. Error estimate

First we recall the energy estimate for linear hyperbolic systems.

Lemma 2.1. *Given a strictly hyperbolic system*

$$\partial_t v + \sum_{l=1}^d A_l(x)\partial_{x_l} v = f,$$

with the initial condition $v(0, x) = v_0(x)$, where $A_l : \mathbb{R}^d \to \mathbb{R}^{N\times N}$, $1 \le l \le d$ are smooth matrix valued functions. For any $T > 0$, there exists a constant C_T such that

$$\sup_{0\le t\le T} \|v(t,x)\|^2_{L^2(\mathbb{R}^d;\mathbb{C}^N)} \le C_T\left(\|v_0(x)\|^2_{L^2(\mathbb{R}^d;\mathbb{C}^N)} + \int_0^T \|f(s,x)\|^2_{L^2(\mathbb{R}^d;\mathbb{C}^N)}\,\mathrm{d}s\right).$$

Next we define

$$v^\varepsilon = (\partial_t u^\varepsilon, \partial_{\boldsymbol{x}} u^\varepsilon)^{\mathrm{T}}, \quad v^\varepsilon_{\mathrm{FGA}} = (\partial_t u^\varepsilon_{\mathrm{FGA}}, \partial_{\boldsymbol{x}} u^\varepsilon_{\mathrm{FGA}})^{\mathrm{T}}, \quad \tilde{v}^\varepsilon = v^\varepsilon - v^\varepsilon_{\mathrm{FGA}}, \tag{2.16}$$

then (2.1) can be viewed as a linear strictly hyperbolic system for v^ε, and thus Lemma 2.1 together with Proposition 6.2 in [24] implies the following estimate.

Theorem 2.2. *For any $T > 0$, there exists a constant C_T and $\varepsilon_0 > 0$ such that, for any $\varepsilon \in (0, \varepsilon_0]$,*

$$\sup_{0\le t\le T} \|\tilde{v}^\varepsilon(t,\boldsymbol{x})\|^2_{L^2(\mathbb{R}^d)} \le C_T\left(\|\tilde{v}^\varepsilon_0(\boldsymbol{x})\|^2_{L^2(\mathbb{R}^d)} + \varepsilon\right), \tag{2.17}$$

where $\tilde{v}^\varepsilon_0(\boldsymbol{x}) = v^\varepsilon(0,\boldsymbol{x}) - v^\varepsilon_{\mathrm{FGA}}(0,\boldsymbol{x})$ is the initial error made in the initial decomposition.

Remark. 1. Proposition 6.2 in [24] shows that the local error caused by FGA is of $O(\varepsilon)$, i.e.

$$\partial_t^2 u^\varepsilon_{\mathrm{FGA}} - \xi(\boldsymbol{x})\Delta u_{\mathrm{FGA}} = C(\varepsilon), \tag{2.18}$$

where C is a constant depending on the final time T, and the velocity profile ξ and its derivatives $\partial^\alpha \xi$ with the multi-index $|\alpha| \le 3$.

2. Define

$$\begin{aligned} U^\varepsilon_{\mathrm{FGA}}(t,\boldsymbol{x}) = &\int \frac{a_+\psi^\varepsilon_+}{(2\pi\varepsilon)^{3d/2}} e^{\frac{\mathrm{i}}{\varepsilon}\boldsymbol{P}_+\cdot(\boldsymbol{x}-\boldsymbol{Q}_+)-\frac{1}{2\varepsilon}|\boldsymbol{x}-\boldsymbol{Q}_+|^2}\,\mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p} \\ &+ \int \frac{a_-\psi^\varepsilon_-}{(2\pi\varepsilon)^{3d/2}} e^{\frac{\mathrm{i}}{\varepsilon}\boldsymbol{P}_-\cdot(\boldsymbol{x}-\boldsymbol{Q}_-)-\frac{1}{2\varepsilon}|\boldsymbol{x}-\boldsymbol{Q}_-|^2}\,\mathrm{d}\boldsymbol{q}\,\mathrm{d}\boldsymbol{p}, \end{aligned} \tag{2.19}$$

then based on the isometry of the FBI transform, one has

$$U^\varepsilon_{\mathrm{FGA}}(0,\boldsymbol{x}) = U^\varepsilon_0(\boldsymbol{x}), \qquad \partial_t U^\varepsilon_{\mathrm{FGA}}(0,\boldsymbol{x}) = U^\varepsilon_1(\boldsymbol{x}).$$

Since (2.3) is the discretization of (2.19) on $\boldsymbol{q}$ and $\boldsymbol{p}$, the initial error $\tilde{v}^\varepsilon_0(\boldsymbol{x})$ is actually the error of discretizing the integrals in (2.19) by the rectangular rule, and thus the accuracy is controlled by the number of Gaussians in (2.3).

3. Suppose τ is the time step for a pth order numerical integrator of (2.6)–(2.11), then there will be an error of $O(\tau^p)$ appearing on the right-hand-side of (2.17).

3. Constrained FWI optimization problem

Suppose that the wavefield u is related to the predicted data via a detection operator $\mathcal{D}$. The velocity is obtained by solving

$$\min_{\xi\in\Xi} \|d - \mathcal{D}u\|_2\,, \tag{3.1}$$

subject to u given by Eq. (2.1), where d is the observed data and Ξ is the state space of ξ. Note the velocity ξ enters the formulation via Eq. (2.1). In what follows, when not causing any ambiguity, we shall use $u(\xi)$ to indicate the mapping from ξ to $u(t,\mathbf{x})$ and $u_{\mathrm{FGA}}(\xi)$ for the mapping from ξ to $u_{\mathrm{FGA}}(t,\mathbf{x})$. Our goal here is to accelerate the optimization by using FGA. Namely, as an approximation to Eq. (3.1), one solves

$$\min_{\xi\in\Xi} \|d - \mathcal{D}u_{\mathrm{FGA}}(\xi)\|_2, \tag{3.2}$$

subject to u_{FGA} given by (2.3). The accuracy of the FGA solution depends on the velocity profile $\xi(\boldsymbol{x})$ and its derivatives [24], and cannot be estimated *a priori*. Hence it is possible that FGA fails to give an accurate approximate in certain state spaces of velocity, leading to erratic inversion results. To address the issue, we propose to solve the optimization in a subspace Ω of Ξ:

$$\min_{\xi\in\Omega} \|d - \mathcal{D}u_{\mathrm{FGA}}(\xi)\|_2, \tag{3.3}$$

where u_{FGA} approximates u accurately for all $\xi \in \Omega$. A critical question here is how to define and identify such a subspace Ω. To this end, we have the following result:

Proposition 3.1. *Suppose C is a positive constant, and let*

$$\Omega = \{\xi \mid \max_{|\alpha|\le 3} |\partial^\alpha \xi| \le C\}, \tag{3.4}$$

where α is a multi-index. If ξ^* is a minimizer of Eq. (3.3), there exists a constant $M > 0$ such that

$$\|d - \mathcal{D}u(\xi^*)\|_2 < \min_{\xi\in\Omega} \|d - \mathcal{D}u(\xi)\|_2 + M\epsilon.$$

Lemma 3.2. *For all $\xi \in \Omega$, there exists a uniform estimate on the accuracy of FGA, i.e., $\exists$ a constant $M > 0$ independent of ξ such that*

$$\|u(\xi) - u_{\mathrm{FGA}}(\xi)\|_2 \le M\epsilon. \tag{3.5}$$

Proof. The conclusion can be straightforwardly deduced by the accuracy estimate in Theorem 2.2 and the expression of remainder terms provided in Proposition 5.5 in [24] which shows the dependence of the constant C_T on the derivatives of ξ. Note that, for simplicity, we have assumed no initial errors made in FGA, i.e., $\tilde{v}_0^\varepsilon = 0$ in (2.17). □

Proof of Proposition 3.1. Let $\xi^\dagger$ be a minimizer of

$$\min_{\xi\in\Omega} \|d - \mathcal{D}u(\xi)\|_2,$$

and we have

$$\begin{aligned}\|d - \mathcal{D}u_{\mathrm{FGA}}(\xi^*)\|_2 &\le \|d - \mathcal{D}u_{\mathrm{FGA}}(\xi^\dagger)\|_2\\ &\le \|d - \mathcal{D}u(\xi^\dagger)\|_2 + \|\mathcal{D}u(\xi^\dagger) - \mathcal{D}u_{\mathrm{FGA}}(\xi^\dagger)\|_2\\ &\le \min_{\xi\in\Omega} \|d - \mathcal{D}u(\xi)\|_2 + \|\mathcal{D}\|_{\mathrm{op}}\|u(\xi^\dagger) - u_{\mathrm{FGA}}(\xi^\dagger)\|_2.\end{aligned} \tag{3.6}$$

Thus we can get

$$\begin{aligned}\|d - \mathcal{D}u(\xi^*)\|_2 &\le \|d - \mathcal{D}u_{\mathrm{FGA}}(\xi^*)\|_2 + \|\mathcal{D}u(\xi^*) - \mathcal{D}u_{\mathrm{FGA}}(\xi^*)\|_2\\ &\le \min_{\xi\in\Omega} \|d - \mathcal{D}u(\xi)\|_2 + \|\mathcal{D}\|_{\mathrm{op}}\|u(\xi^\dagger) - u_{\mathrm{FGA}}(\xi^\dagger)\|_2 + \|\mathcal{D}\|_{\mathrm{op}}\|u(\xi^*) - u_{\mathrm{FGA}}(\xi^*)\|_2.\end{aligned} \tag{3.7}$$

According to Lemma 3.2, there exists a constant $M' > 0$ such that

$$\|u(\xi) - u_{\mathrm{FGA}}(\xi)\|_2 \le M'\epsilon, \quad \text{for all } \xi \in \Omega,$$

Eq. (3.5) follows immediately by letting $M = 2\|\mathcal{D}\|_{\mathrm{op}}M'$. □

Proposition 3.1 suggests that Eq. (3.2) is a good approximation to Eq. (3.1) for $\xi \in \Omega$ given in (3.4). However, it is still not convenient to identify the appropriate Ω (or equivalently C) *a priori*, and thus we propose the following constrained optimization formula to address this issue in practice, in which Ω is approximated by

$$\Omega' = \{\xi \mid \int_0^T \int \left| \frac{\partial^2 u_{\mathrm{FGA}}}{\partial t^2} - \xi(\mathbf{x})\,\nabla u_{\mathrm{FGA}} \right|^2 d\mathbf{x}dt \le \delta\}, \tag{3.8}$$

where δ is a prescribed constant, and Eq. (3.3) becomes

$$\min_{\xi\in\Omega'} \|d - \mathcal{D}u_{\mathrm{FGA}}(\xi)\|_2^2. \tag{3.9}$$

Obviously the constraint equation (3.8) degenerates as $u_{\mathrm{FGA}} \to u$ (i.e., $\varepsilon \to 0$).

Remark. The choice of the constant δ is highly problem dependent and ideally it should be chosen in a way that the truth is contained in Ω, which certainly is difficult to verify in practice. Thus a practical solution is to try different values and use the one that provides the best results.

4. Multi-level particle swarm optimization algorithm

In this section, we describe the multi-level particle swarm optimization (MLPSO) algorithm to solve the optimization problem (3.9). We start with a brief introduction to the standard particle swarm optimization (PSO) algorithm for unconstrained optimization problems.

4.1. Particle swarm optimization

PSO, first introduced by Kennedy and Eberhart [22], is a stochastic global optimization technique inspired by the swarm behavior of animals. Suppose one wants to find the solution of $\min_\xi f(\xi)$. PSO employs a population of agents called particles that fly through the state space of the problem following some prescribed rules and search for the optimal solution. Each particle is characterized by two variables: the position ξ representing the optimization variables and the velocity v describing how the position is updated. A major feature of the PSO algorithm is that the particles share information during the search. Namely, each particle combines its own best attained solution with the best solution of the whole swarm to

determine its search pattern. More specifically the velocity v and position ξ of a particle at each iteration are updated as [22],

$$v_n^k = c_0\, v_n^k + c_1\, \alpha_n^k \left(p_n^k - \xi_n^k\right) + c_2\, \beta_n^k \left(g - \xi_n^k\right), \tag{4.1a}$$

$$\xi_n^k = \xi_n^k + v_n^k, \tag{4.1b}$$

where $k = 1, \cdots, K$ and $n = 1, \cdots, N$ are the iteration index and the particle index respectively. p_n^k is the best value of the n-th particle at iteration n:

$$p_n^k = \arg\min\{f(\xi_n^1),\ \ldots,\ f(\xi_n^k)\},$$

and g is the swarm's global best positions:

$$g^k = \arg\min\{f(p_1^k),\ \ldots,\ f(p_N^k)\}.$$

The coefficient c_0 is known as the "inertial weight" which is used to provide a good balance between global exploration and local exploitation [5], and in practice, c_0 is usually designed to vary with respect to time. Constants c_1 and c_2 are called the cognitive and social parameters, quantifying a particle's tendency to move toward its own best and the global best solutions respectively. The variables α_n^k and β_n^k are drawn from a uniform distribution $[0, 1]$, which renders the optimization procedure stochastic.

The standard PSO algorithm is originally developed for unconstrained optimization problems, while Eq. (3.9) is a constrained optimization. Therefore we use a trial-and-error PSO algorithm proposed by Hu and Eberhart [17] for constrained optimization problems, while noting that other constraint-handling mechanisms (e.g., [32]) are also possible. The advantage of the algorithm is that the constraints are handled in a very simple manner, and each particle searches the whole state space, but only keeps tracking feasible solutions. Specifically two modifications are made on the standard PSO algorithm: particles were only initialized to feasible positions; only those particles that satisfy the constraints are used for the local and global best positions.

Theoretical analysis of the PSO algorithm is rather difficult, and to the best of our knowledge, only a number of partial results are available (see [29] for a detailed discussion), and the general convergence result of the method is still an open problem. A comprehensive introduction to PSO is not in the scope of the paper and we refer interested readers to [5,29,4] and the references therein for further details such as parameter selection and existing variants.

4.2. Multi-level particle swarm optimization

One can implement the PSO algorithm with Eq. (2.1) solved by FGA; however, this can still be rather expensive, since an accurate FGA solution requires a large number of Gaussians. On the other hand, when the particle swarm is far from the solution, it is not necessary to evaluate the objective function for all the particles with high fidelity. This motivates us to propose an MLPSO algorithm that combines FGA solvers with different levels of fidelity to reduce the computational cost. In particular, we assume that $f_l(\xi)$ and $f_h(\xi)$ are respectively a low-fidelity and a high-fidelity approximation to the objective function $f(\xi)$ in Eq. (3.9). In our problem, f_l is constructed by FGA with a small number of Gaussians while f_h is constructed by FGA with a large number of Gaussians.

A common idea in designing multi-fidelity algorithm is to compute all the candidate solutions with the low fidelity model, and then choose a certain fraction of the solutions and re-compute them with the high fidelity model. The key issue here is to find which solutions should be computed with the high fidelity model. In this problem, it is easy to see that the particles with smaller objective function values are more important than those with larger values and should be evaluated with the high fidelity model. Thus in each iteration, one first evaluates the objective function of all the particles with the low fidelity model $f_l(\xi)$, and then re-evaluate a certain percentage of the "best" particles (namely those with the smallest objective function values) with the high fidelity model $f_h(\xi)$. More importantly, as the swarm moves closer to the solution, a larger fraction of particles should be evaluated with the high-fidelity model and eventually all particles should be computed with the high fidelity model. Namely the percentage of particles evaluated with f_h should increase with iteration number. To this end, we define a non-decreasing function $\rho = \rho(k)$ where $0 \le \rho(k) \le 1$ and $\rho(K) = 1$ to compute the percentage. The function $\rho(k)$ is critical for the performance of the algorithm and should be chosen problem dependently. We describe the complete MLPSO procedure in Algorithm 1. Finally we note that since we require that the percentage function $\rho(K) = 1$, i.e., all the solutions are computed with the high fidelity model in the end, the convergence analysis of the MLPSO algorithms is the same as that of the standard single-level PSO.

5. Numerical example

The performance of FGA-based MLPSO is demonstrated by a two-dimensional FWI problem. The true velocity profile is taken to be the well-known smoothed Marmousi model [40] in Fig. 1 [42]. The computational domain is a $[0, 4] \times [0, 3]$ rectangle. The initial wave is generated from a point source located at the center of the domain shown in Fig. 2. The wavefield data used for the inversion is simulated by propagating the source to $T = 0.2$ using the highly accurate lowrank symbol approximation method [8] on a 1000×750 mesh. We plot the wavefield data in Fig. 3.

Input: Low-fidelity model $f_l(\xi)$, high-fidelity model $f_h(\xi)$, constraint $c(\xi) < 0$, swarm size N, maximum iteration number K, the percentage function $\rho(k)$, the inertia weight $c_0(k)$, the cognitive parameter c_1 and the social parameter c_2;
Output: Best solution g;
Initialization: Randomly generate the position of N particles in the feasible space (i.e., all particles satisfy the constraints): $\{\xi_n^1\}_{n=1}^N$ and let $pbest_n = \xi_n^1$;
for $k = 1$ **to** K **do**
  For each particle, evaluate the objective function with the low-fidelity model: $y_n^k = f_l(\xi_n^k)$ for $n = 1 \dots N$;
  Compute $\rho = \rho(k)$ and let y_ρ be the ρ-th quantile of $\{y_n^k\}_{n=1}^N$;
  for $n = 1$ **to** N **do**
    if $y_n^k < y_\rho$ **then**
      Re-calculate y_n^k with the high-fidelity model: $y_n^k = f_h(\xi_n^k)$;
    end
  end
  for $n = 1$ **to** N **do**
    if $y_n^k < pbest_n$ *and* $c(\xi_n^k) < 0$ **then**
      $pbest_n = y_n^k$, $p_n = \xi_n^k$
    end
  end
  Let $g = p_{n^*}$ where $pbest_{n^*} = \min\{pbest_n\}_{n=1}^N$;
  if $k < K$ **then**
    Update the position of each particle ξ_n^{k+1} using Eqs. (4.1);
  end
end

Algorithm 1: Multi-level particle swarm optimization algorithm.

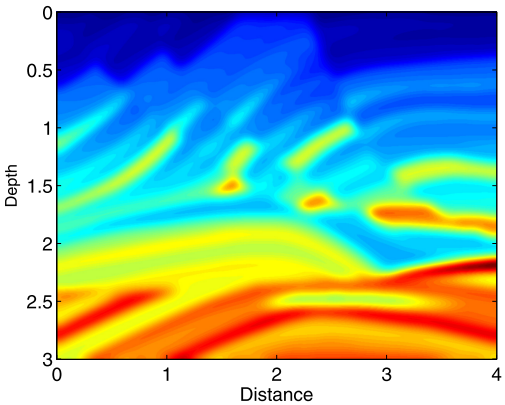

Fig. 1. The smoothed Marmousi velocity model.

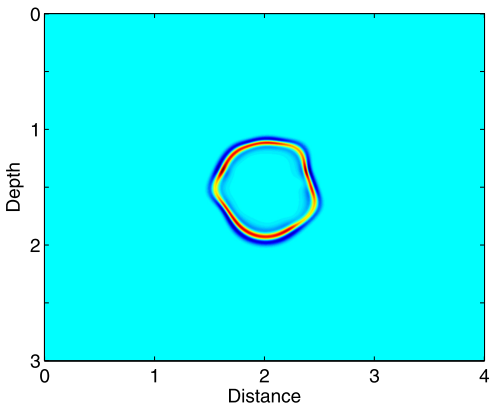

Fig. 2. The initial condition.

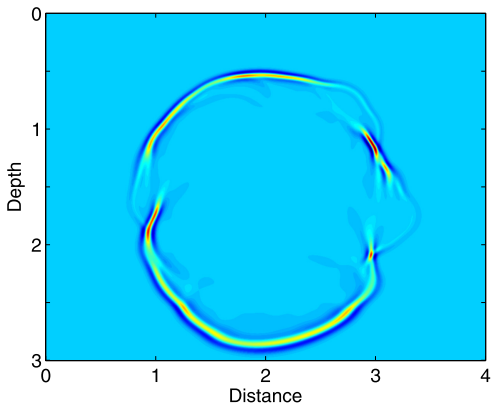

Fig. 3. The wavefield data collected at $T = 0.2$.

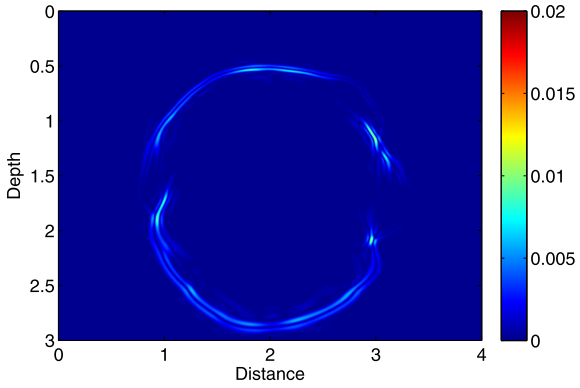

Fig. 4. The errors in FGA with 300 Gaussians.

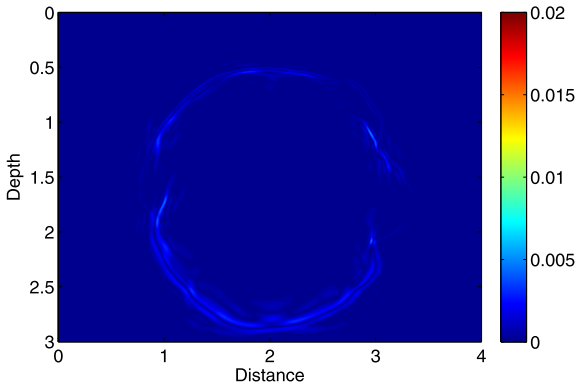

Fig. 5. The errors in FGA with 3000 Gaussians.

In MLPSO, we use two FGA solvers: The high-fidelity model uses 3000 Gaussians and the low-fidelity one uses 300 Gaussians, which implies the low-fidelity model is ten times faster than the high-fidelity model. In both models, the time integration is performed with a fourth-order Runge–Kutta method with sufficiently small time step 0.001 so that the numerical error due to time integration can be neglected. To validate the FGA solvers, we compute the wavefield by FGA for the true velocity model and compare it with the data. In Fig. 4, we show the difference between the data and the result computed with 300 Gaussians and in Fig. 5 we show that with 3000 Gaussians. The results show that the FGA with 3000 beams produces a sufficiently good approximation of the true solution. On the other hand, the error in FGA with 300 beams is rather large; nevertheless, as we can see later, it is still useful in MLPSO. We also show the computational cost for the two FGA models in Table 1. It should also be noted that in what follows the inversion is performed with a different solver (i.e. FGA) from that is used to generate the data, and so no inverse crime is committed.

We assume that the velocity model can be written as a linear combination of a set of features:

Table 1
The number of high fidelity model and low fidelity model evaluations for all four choices of percentage function.

Percentage function	ρ_1	ρ_2	$\rho = 0$	$\rho = 1$
Low fidelity model evaluations	20,000	20,000	20,000	0
High fidelity model evaluations	7473	9422	0	20,000
Computer time (hrs)	26	32	6	56
Best function value	0.0147	0.0147	0.0541	0.0147

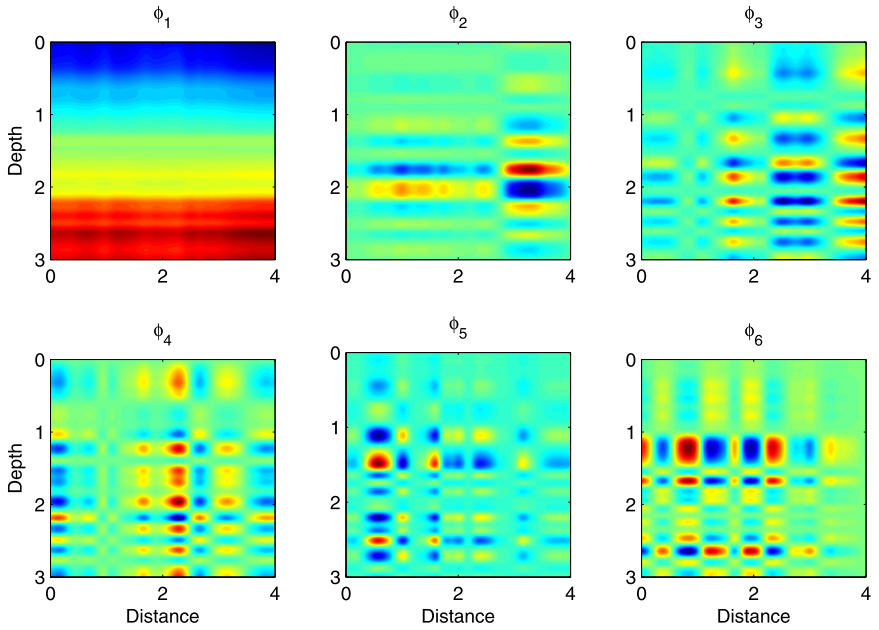

Fig. 6. The top 6 principle components of the true velocity profile.

$$\xi(\mathbf{x}) = \sum_{j=0}^{J} z_j \phi_j(\mathbf{x}),$$

where $\phi_1, \phi_2, \ldots$ are features and $z_1, z_2, \ldots$ are coefficients. In practice the features are available from prior knowledge (e.g. existing rough information about rock structures and velocity profiles), and here we generated the features by applying a singular value decomposition (SVD) to the true velocity (which is represented as a matrix) and taking the top $J = 20$ components. As an example, we show the top 6 features in Fig. 6. Thus we reduce it to a twenty-dimensional problem and now all we need to do is to determine the coefficients $\{z_j\}_{j=1}^{20}$. In reality it is unlikely to have such accurate features that almost exactly resemble the true velocity model. Here we use these synthetic features to minimize its impact on the inversion and so we can be focused on the performance of our inversion algorithm.

In this example we choose the constraint parameter $\delta = 10^{-2}$, and use the MLPSO method to solve the constrained FWI problem. Specifically we choose swarm size $N = 100$, maximum number of iteration $K = 200$, cognitive parameter $c_1 = 2.8$ and social parameter $c_2 = 1.3$. The values of the cognitive and social parameters are chosen as recommended in [18]. Following [38], the inertial weight is chosen to be time dependent, linearly decreasing from 0.9 to 0.4:

$$c_0(k) = 0.9 - \frac{k}{400}.$$

As stated in Section 4, a critical issue in the algorithm is to determine the percentage function ρ. Obviously a fast increasing $\rho(k)$ may produce more accurate results but it is more computationally intensive, while a slowly increasing function is less computationally intensive but its results can be less accurate. To further analyze the impact of the percentage function, we use several different choices of $\rho(k)$ in this example and compare their performance. Namely we choose $\rho_1 = (k/K)^4$, $\rho_2 = k/K$, and for comparison purposes, we also perform the standard PSO with only the low fidelity model being employed, corresponding to $\rho_3 = 0$ in MLPSO, and with only the high fidelity model being employed, corresponding to $\rho_4 = 1$ in MLPSO. The four different percentage choices are plotted against the iteration number in Fig. 7 and we can see that ρ_2 increases much faster than ρ_1.

The total computational cost for the four different choices of percentage function is compared in Table 1. One can see that ρ_1, ρ_2 and $\rho = 1$ yield comparable optimization results while ρ_1 used the least number of high-fidelity model

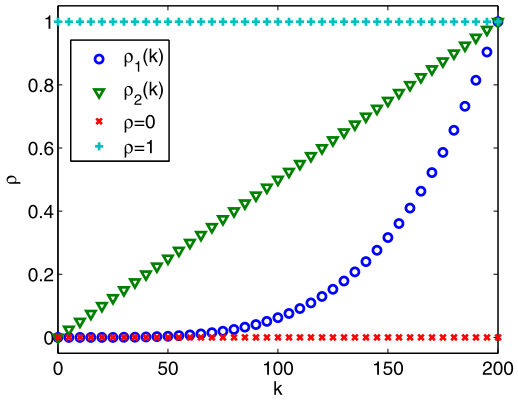

Fig. 7. The four different percentage functions $\rho(k)$ used in the example.

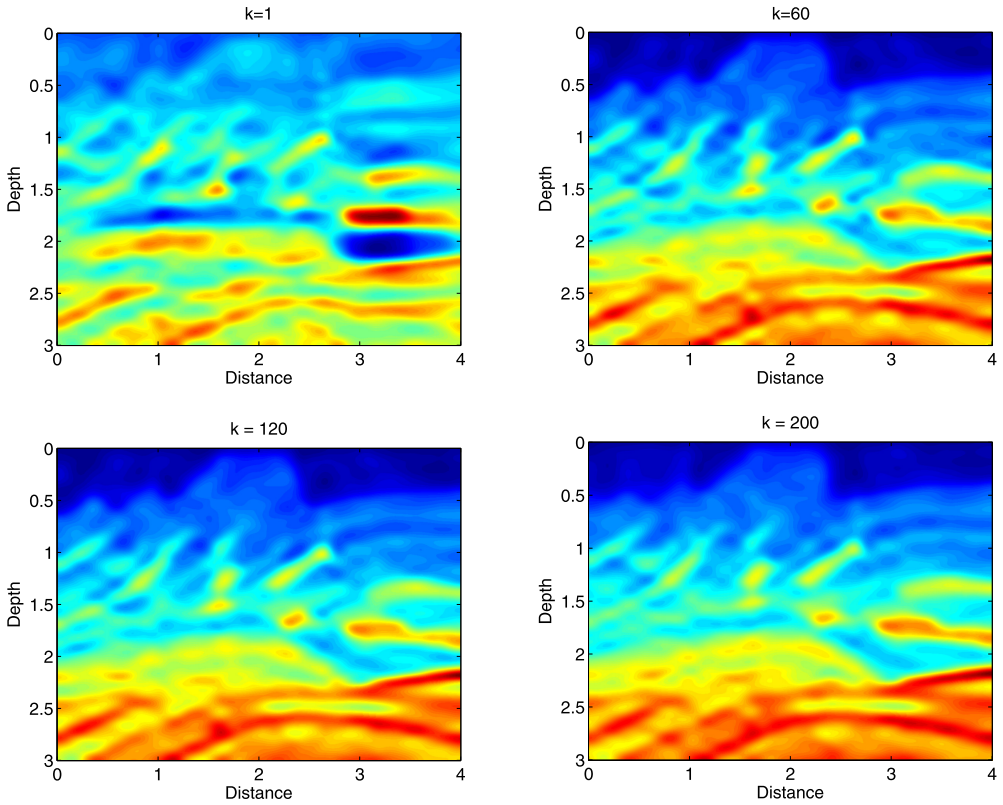

Fig. 8. The results of the multi-level PSO with $\rho_1(k)$. The best velocity found at iteration $k = 1$, $k = 60$, $k = 120$ and $k = 200$.

evaluations. We then compare the computational results of them. In Figs. 8–11, we show the best velocity profile found at the iteration $k = 1$, 60, 120, and 200, for $\rho_1, \ldots, \rho_4$ respectively. In Fig. 10 one can see that when the low fidelity model is only employed, the final results differ significantly from the true velocity profile, indicating that the low fidelity model by itself is not sufficient to produce reliability inversion results. When the high fidelity model is only employed, as shown in Fig. 11, the inversion results agree well with the true velocity. Once again, we see that MLPSO with both ρ_1 and ρ_2 produces comparable results to the high fidelity model, while ρ_1 is less computationally expensive than ρ_2.

Finally, to exam the method's ability to find the global minimal and understand how it depends on the particle size in each iteration, we performed 20 numerical tests each with a randomly chosen initial guess. And in each test, we use 20, 40 and 100 particles per iteration. We have found that, with 20 particles, none of the results converges to the true solution. With 40 particles, 14 tests correctly find the true solution, and with 100 particles, all the 20 test results converge to the true solution, which indicates that whether the method's ability to find the global solution highly depends on the particle size and if possible one should use a rather large particle size to increase the possibility of finding the global solution.

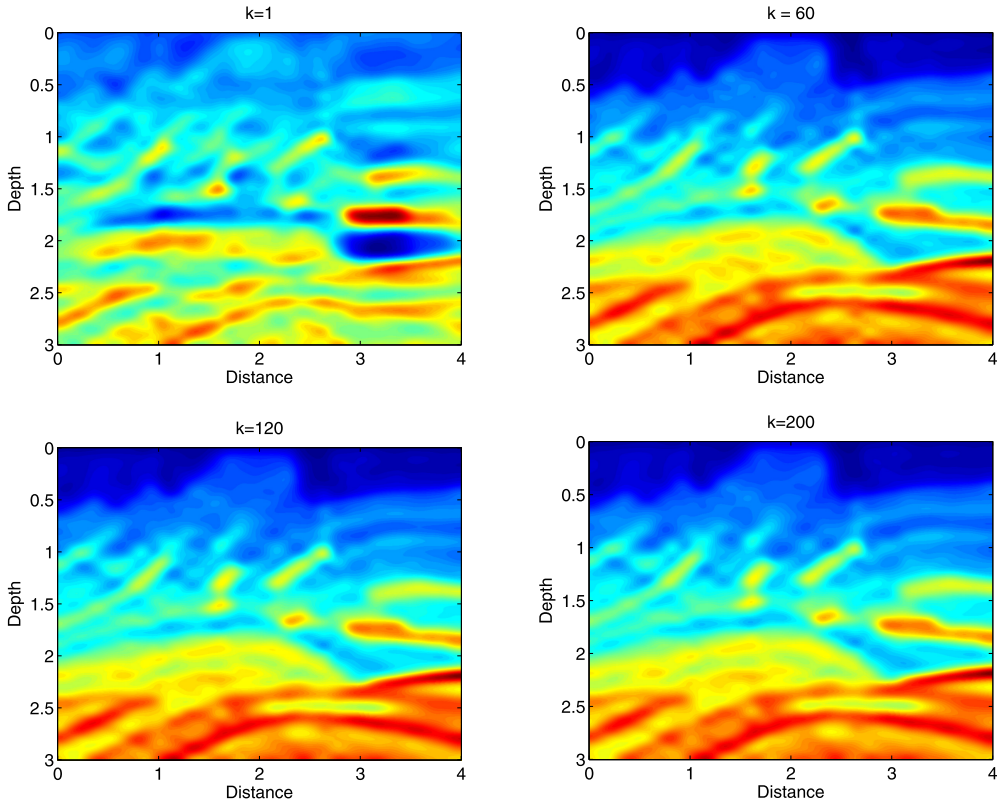

Fig. 9. The results of the multi-level PSO with $\rho_2(k)$. The best velocity found by using at iteration $k = 1$, $k = 60$, $k = 120$ and $k = 200$.

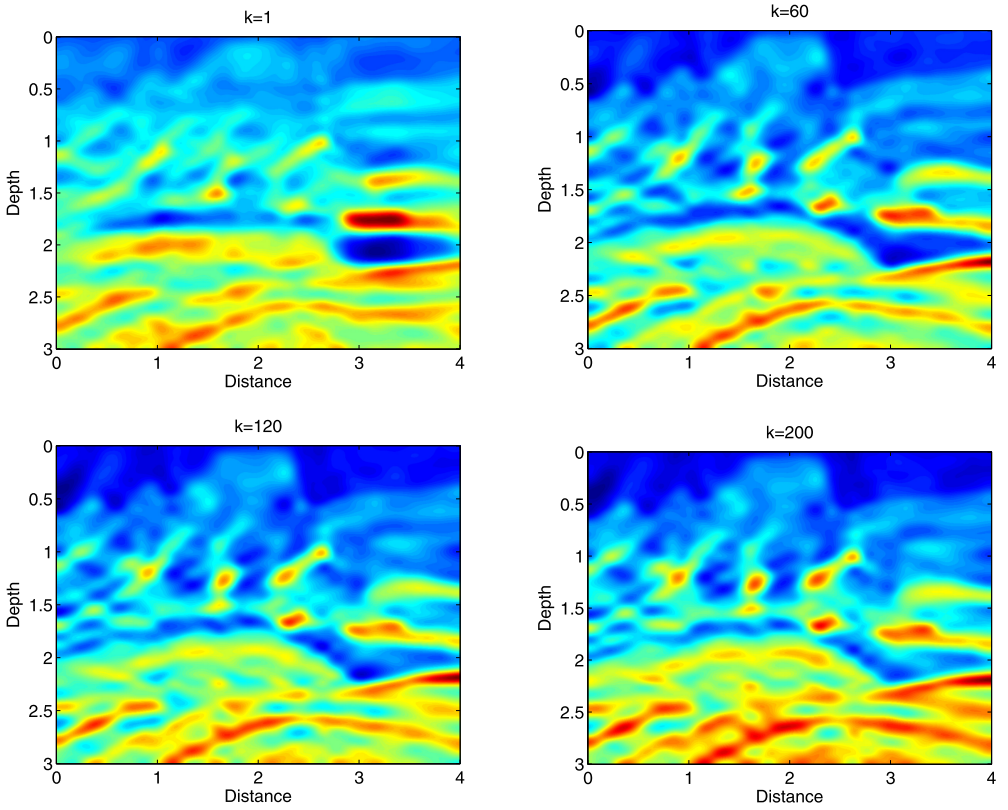

Fig. 10. The results of a standard PSO with the low-fidelity model ($\rho = 0$). The best velocity found by using at iteration $k = 1$, $k = 60$, $k = 120$ and $k = 200$.

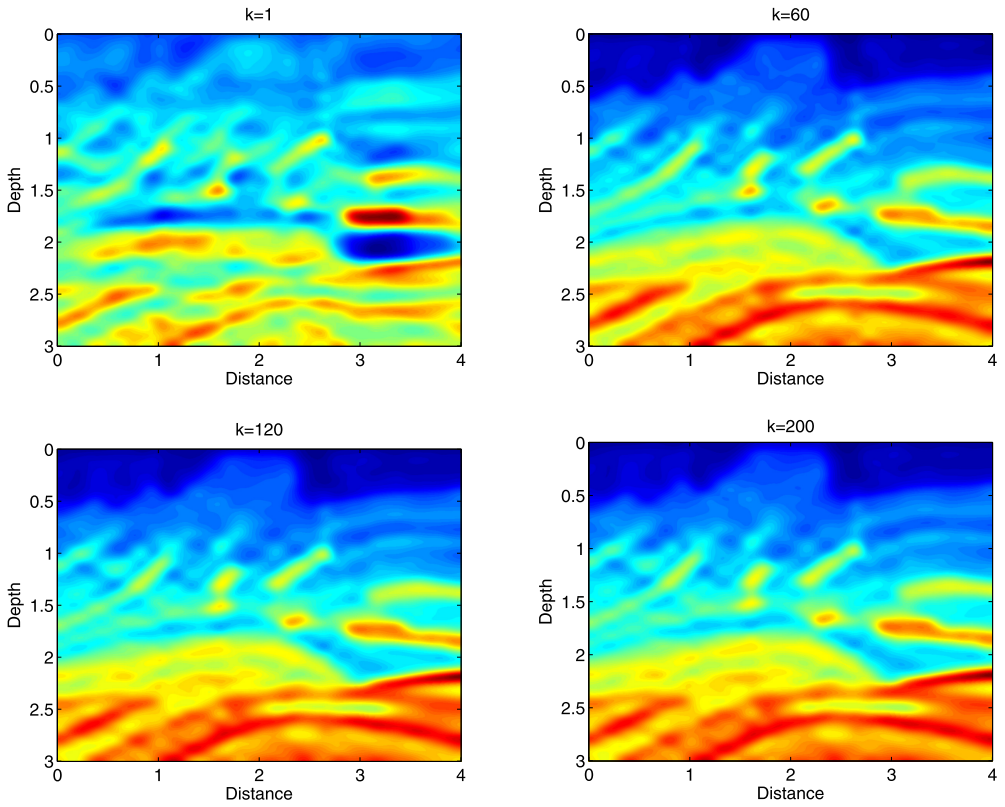

Fig. 11. The results of a standard PSO with the high-fidelity model ($\rho = 1$). The best velocity found by using at iteration $k = 1$, $k = 60$, $k = 120$ and $k = 200$.

6. Conclusions

In summary, we develop a frozen Gaussian approximation (FGA)-based multi-level particle swarm optimization (MLPSO) method for seismic inversion of high-frequency wave data. We use FGA to compute asymptotic solutions to the wave equation, and design a constrained full waveform inversion (FWI) optimization problem to prevent the optimization search moving out of the region where FGA does not produce an accurate solution. The constrained FWI problem is consistent with the original FWI when the wave equation is solved exactly. We propose a particle swarm optimization algorithm with multi-fidelity models of FGA to efficiently solve the global optimization problem. Numerical results are shown to verify the FGA-based MLPSO method and indicate the efficiency and robustness of the method.

The proposed MLPSO method only finds an approximate solution to the inversion problem due to the constraint of optimization on the FGA-valid regions. As mentioned in the introduction, if a more accurate solution is desired, one can use the results of MLPSO as an ideal initial guess for a gradient-based method (e.g., Born approximation) with direct simulations of the wave equation.

There are some possible extensions of the work that we plan to investigate in the future. First, for a fixed wavelength, the accuracy of FGA is ultimately limited by the asymptotic error no matter how much computational effort is input. However, a more accurate solver can be obtained by combining FGA and direct simulation methods. Namely, we can construct a surrogate model based on FGA and then use direct simulations to improve it. Secondly, in many practical cases, finding a solution is not the only goal of the inverse problems. It is also important to quantify the uncertainty in the solution. In this respect, the Bayesian methods are particularly useful. In the Bayesian setting, a large number of simulations of the wave equation are required to reconstruct the posterior distribution of the velocity. We believe that the FGA method can be used to develop computationally efficient algorithms for the Bayesian inverse problems as well.

Acknowledgements

J. Li was partially supported by the National Science Foundation of China under grant number 11301337. G. Lin was supported by the Applied Mathematics Program within the DOE's Office of Advanced Scientific Computing Research as part of the Collaboratory on Mathematics for Mesoscopic Modeling of Materials. Pacific Northwest National Laboratory (PNNL) is operated by Battelle for the DOE under Contract DE-AC05-76RL01830. X. Yang was partially supported by the NSF grants DMS-1418936, and DMS-1107291: NSF Research Network in Mathematical Sciences "KI-Net: Kinetic description of emerging

challenges in multiscale problems of natural science", and the Regents Junior Faculty Fellowship of University of California, Santa Barbara.

References

[1] U. Albertin, D. Yingst, H. Jaramillo, Comparing common-offset Maslov, Gaussian beam, and coherent state migrations, in: 71st Ann. Internat. Mtg., Soc. of Expl. Geophys., 2001, pp. 913–916.
[2] V. Cerveny, Seismic Ray Theory, Cambridge University Press, Cambridge, 2001.
[3] V. Cerveny, M.M. Popov, I. Psencik, Computation of wave fields in inhomogeneous media – Gaussian beam approach, Geophys. J. R. Astron. Soc. 70 (1982) 109–128.
[4] Maurice Clerc, Particle Swarm Optimization, vol. 93, John Wiley & Sons, 2010.
[5] Russell C. Eberhart, Yuhui Shi, Particle swarm optimization: developments, applications and resources, in: Proceedings of the 2001 Congress on Evolutionary Computation, 2001, vol. 1, IEEE, 2001, pp. 81–86.
[6] B. Engquist, O. Runborg, Computational high frequency wave propagation, Acta Numer. 12 (2003) 181–266.
[7] Sergey Fomel, Nick Tanushev, et al., Time-domain seismic imaging using beams, in: 2009 SEG Annual Meeting, Society of Exploration Geophysicists, 2009.
[8] Sergey Fomel, Lexing Ying, Xiaolei Song, Seismic wave extrapolation using lowrank symbol approximation, Geophys. Prospect. 61 (3) (2013) 526–536.
[9] D. Foster, R. Wu, C. Mosher, Coherent-state solutions of the wave equation, in: 72nd Ann. Internat. Mtg., Soc. of Expl. Geophys., 2002, pp. 1348–1351.
[10] S.H. Gray, Efficient traveltime calculations for Kirchhoff migration, Geophysics 51 (1986) 1685–1688.
[11] S.H. Gray, Gaussian beam migration of common-shot records, Geophysics 70 (2005) S71–S77.
[12] Samuel H. Gray, Norman Bleistein, True-amplitude Gaussian-beam migration, Geophysics 74 (2) (2009) S11–S23.
[13] Rania Hassan, Babak Cohanim, Olivier De Weck, Gerhard Venter, A comparison of particle swarm optimization and the genetic algorithm, in: Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference, 2005, pp. 18–21.
[14] M.F. Herman, E. Kluk, A semiclassical justification for the use of non-spreading wavepackets in dynamics calculations, Chem. Phys. 91 (1984) 27–34.
[15] N.R. Hill, Gaussian beam migration, Geophysics 55 (1990) 1416–1428.
[16] N.R. Hill, Prestack Gaussian-beam depth migration, Geophysics 66 (2001) 1240–1250.
[17] Xiaohui Hu, Russell Eberhart, Solving constrained nonlinear optimization problems with particle swarm optimization, in: 6th World Multiconference on Systemics, Cybernetics and Informatics, 2002, pp. 203–206.
[18] Vijay Kalivarapu, Jung-Leng Foo, Eliot Winer, Improving solution characteristics of particle swarm optimization using digital pheromones, Struct. Multidiscip. Optim. 37 (4) (2009) 415–427.
[19] K. Kay, Integral expressions for the semi-classical time-dependent propagator, J. Chem. Phys. 100 (1994) 4377–4392.
[20] K. Kay, The Herman–Kluk approximation: derivation and semiclassical corrections, Chem. Phys. 322 (2006) 3–12.
[21] T.H. Keho, W.B. Beydoun, Paraxial ray Kirchhoff migration, Geophysics 53 (1988) 1540–1546.
[22] James Kennedy, Russell Eberhart, et al., Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, Perth, Australia, 1995, pp. 1942–1948.
[23] J. Lu, X. Yang, Frozen Gaussian approximation for high frequency wave propagation, Commun. Math. Sci. 9 (2011) 663–683.
[24] J. Lu, X. Yang, Convergence of frozen Gaussian approximation for high frequency wave propagation, Commun. Pure Appl. Math. 65 (2012) 759–789.
[25] J. Lu, X. Yang, Frozen Gaussian approximation for general linear strictly hyperbolic systems: formulation and Eulerian methods, Multiscale Model. Simul. 10 (2) (2012) 451–472.
[26] A. Martinez, An Introduction to Semiclassical and Microlocal Analysis, Springer-Verlag, New York, 2002.
[27] R. Nowack, M. Sen, P. Stoffa, Gaussian beam migration for sparse common-shot and common-receiver data, in: 73rd Ann. Internat. Mtg., Soc. of Expl. Geophys., 2003, pp. 1114–1117.
[28] Rene-Edouard Plessix, Introduction: towards a full waveform inversion, Geophys. Prospect. 56 (6) (2008) 761–763.
[29] Riccardo Poli, James Kennedy, Tim Blackwell, Particle swarm optimization, Swarm Intell. 1 (1) (2007) 33–57.
[30] M.M. Popov, N.M. Semtchenok, P.M. Popov, A.R. Verdel, Depth migration by the Gaussian beam summation method, Geophysics 75 (2010) S81–S93.
[31] M.M. Popov, Ray Theory and Gaussian Beam Method for Geophysicists, EDUFBA, 2002.
[32] Gregorio Toscano Pulido, Carlos A. Coello Coello, A constraint-handling mechanism for particle swarm optimization, in: Congress on Evolutionary Computation, 2004. CEC2004, vol. 2, IEEE, 2004, pp. 1396–1403.
[33] Malcolm Sambridge, Guy Drijkoningen, Genetic algorithms in seismic waveform inversion, Geophys. J. Int. 109 (2) (1992) 323–342.
[34] Puneet Saraswat, Mrinal K. Sen, et al., Simultaneous stochastic inversion of prestack seismic data using hybrid evolutionary algorithm, in: 2010 SEG Annual Meeting, Society of Exploration Geophysicists, 2010.
[35] Mrinal K. Sen, Paul L. Stoffa, Nonlinear one-dimensional seismic waveform inversion using simulated annealing, Geophysics 56 (10) (1991) 1624–1638.
[36] Mrinal K. Sen, Paul L. Stoffa, Rapid sampling of model space using genetic algorithms: examples from seismic waveform inversion, Geophys. J. Int. 108 (1) (1992) 281–292.
[37] Mrinal K. Sen, Paul L. Stoffa, Global Optimization Methods in Geophysical Inversion, Elsevier, 1995.
[38] Yuhui Shi, Russell C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99, vol. 3, IEEE, 1999.
[39] Khiem T. Tran, Dennis R. Hiltunen, Two-dimensional inversion of full waveforms using simulated annealing, J. Geotech. Geoenviron. Eng. 138 (9) (2011) 1075–1090.
[40] R. Versteeg, The Marmousi experience: velocity model determination on a synthetic complex data set, Lead. Edge 13 (09) (1994) 927–936.
[41] Jean Virieux, Stephane Operto, An overview of full-waveform inversion in exploration geophysics, Geophysics 74 (6) (2009) WCC1–WCC26.
[42] X. Yang, J. Lu, S. Fomel, et al., Seismic modeling using the frozen Gaussian approximation, in: 2013 SEG Annual Meeting, Society of Exploration Geophysicists, 2013.