ecoPETROL

ctyf@ecopetrol.com.co

A GPU IMPLEMENTATION OF THE SECOND ORDER ADJOINT STATE THEORY TO QUANTIFY THE UNCERTAINTY ON FWI RESULTS

UNA IMPLEMENTACIÓN EN GPU DE LA TEORÍA DEL ESTADO ADJUNTO DE SEGUNDO ORDEN PARA LA CUANTIFICACIÓN DE LA INCERTIDUMBRE EN RESULTADOS DE FWI

Abreo, Sergio-Alberto[a]*; Ramírez, Ana-B.[a]; Reyes, Oscar-Mauricio[a].

ABSTRACT

The second order scattering information provided by the Hessian matrix and its inverse plays an important role in both, parametric inversion and uncertainty quantification. On the one hand, for parameter inversion, the Hessian guides the descent direction such that the cost function minimum is reached with less iterations. On the other hand, it provides *a posteriori* information of the probability distribution of the parameters obtained after full waveform inversion, as a function of the *a priori* probability distribution information.

Nevertheless, the computational cost of the Hessian matrix represents the main obstacle in the state-of-the-art for practical use of this matrix from synthetic or real data. The second order adjoint state theory provides a strategy to compute the exact Hessian matrix, reducing its computational cost, because every column of the matrix can be obtained by performing two forward and two backward propagations.

In this paper, we first describe an approach to compute the exact Hessian matrix for the acoustic wave equation with constant density. We then provide an analysis of the use of the Hessian matrix for uncertainty quantification of the full waveform inversion of the velocity model for a synthetic example, using the 2D acoustic and isotropic wave equation operator in time.

RESUMEN

La información de dispersión de segundo orden proporcionada por la matriz Hessiana y su inversa juegan un papel importante en la inversión paramétrica y en la cuantificación de la incertidumbre. Para la inversión de parámetros, el Hessiano guía la dirección de descenso de manera que se alcanza el mínimo de la función de costo en un menor número de iteraciones. Por otro lado, proporciona información *a posteriori* de la distribución de probabilidad de los parámetros obtenidos luego de usar la inversión de onda completa, como una función de la distribución de probabilidad *a priori*.

Sin embargo, el costo computacional de la matriz Hessiana representa el principal obstáculo de este método para su uso práctico sobre datos sintéticos o datos reales. La teoría del estado adjunto de segundo orden proporciona una estrategia para calcular la matriz Hessian exacta, reduciendo su costo computacional, ya que cada columna de la matriz se puede obtener realizando dos propagaciones hacia adelante y dos hacia atrás.

En este artículo, primero mostramos una metodología para calcular la matriz Hessiana exacta usando la ecuación de onda acústica con densidad constante. Luego, proporcionamos un análisis del uso de la matriz Hessiana para la cuantificación de la incertidumbre de la inversión de onda completa en un ejemplo sintético, utilizando como operador la ecuación de onda acústica 2D, isotrópa con densidad constante en el dominio del tiempo.

KEYWORDS / PALABRAS CLAVE

AFFILIATION

[a]CPS Research Group, Depart. Electrical, Electronic and Telecom. Eng- Universidad Industrial de Santander, carrera 27 calle 9 Bucaramanga, Colombia
*email: abreosergio@gmail.com

1 INTRODUCTION

Recently, full waveform inversion (FWI) has become an attractive technique for the estimation of high resolution parameters of the Earth's subsurface. FWI is formulated as a local optimization problem, where first-order or pre-conditioned first-order functions of the misfit between observed and modeled data, are used to find the search direction to the nearest minimum [1],[2] The inverse of the Hessian matrix not only guides the descent direction in the local optimization method, but also allows quantifying the uncertainty of the parameters estimated with FWI. For uncertainty quantification, the inverse of the Hessian matrix is related to the covariance matrix of the estimated set of parameters when Bayesian inference and Gaussian models are used to model the error between the real and estimated parameters [3]In fact, uncertainty of the FWI can be quantified using Gaussian model assumptions when ℓ_2-error norms are used as data-fitting term[4]. The second-order information of the misfit function, known as the Hessian matrix, is usually neglected in FWI implementations because computing this matrix for large scale problems (real inverse problems) is computationally unfeasible. Several methods have been proposed to compute approximations of the Hessian matrices, such as the BFGS and its variants [5], Newton, Gauss-Newton and Levenberg-Marquardt [6].

The main contributions of this work are two-fold. First, we provide the formulation of the adjoint wavefields used to compute the Hessian matrix using the Second Order Adjoint State Method (SOASM) for the acoustic and isotropic case. Special emphasis is placed on computing the adjoint wavefields using parallel programming such that a feasible implementation on GPUs is proposed. Second, we provide the link between the Hessian matrix and the uncertainty quantification of the estimated velocity model obtained with the acoustic and isotropic FWI. This is a function of the uncertainty of an a priori velocity model. The analysis of uncertainty of the estimated parameters are performed for different central frequencies of the source. Thus, we can identify which parameters are better resolved at different frequencies.

The exact Hessian matrix is obtained via the SOASM, which computes Hessian matrix-vector products by using auxiliary variables called *adjoint variables* [6]-[10]. As regards the acoustic and isotropic case, we obtained every column of the Hessian matrix using four wave propagations via finite-differences in time-domain (FDTD). In the FDTD scheme, every spatial point of a pressure wavefield snapshot can be computed independently, which generates an intrinsic parallelism in the numerical computation of the wave propagations. In this work, we exploit the high level of parallelism of the pressure wavefield modelled with FDTD through the use of parallel architectures as GPUs [11], for reducing the computational cost of the Hessian matrix computation.

This paper is organized as follows: the second section presents the adjoint wavefield equations and the pseudo-code of the parallel implementation on a GPU architecture. The third section describes the strategy for quantifying uncertainty of the estimated parameters after FWI. The fourth and fifth sections show the experiment conducted to test the SOASM implementation and the corresponding results with synthetic velocity models, respectively.

2. THEORETICAL FRAME

In this section, we describe the adjoint wavefields that are used to obtain Hessian-vector products. In particular, we formulate the adjoint wavefields for the acoustic and constant density model of a seismic source wave propagation. Furthermore, the pseudo-code of the adjoint wavefields implementation is presented and its computational cost on CPUs and GPUs is also discussed.

Let $H(v)$ be the Hessian matrix as a function of a given velocity model v. The ith column of $H(v)$ can be obtained through the interaction of the wavefields u and α with the adjoint wavefields $u_1^\dagger$ and $u_2^\dagger$ as

$$H(v)w_i = <u_2^\dagger, \frac{\partial(L(u,v)-f)}{\partial v}> + <u_1^\dagger, \frac{\partial(L(\alpha,v)-\Phi_w)}{\partial v}>$$
$$> + \sum_{j=1}^{N_x \times N_z} w_i(j) < u_1^\dagger, \frac{\partial^2(L(u,v)-f)}{\partial v^2} > \tag{1}$$

where w_i is the vector used to select the column of the Hessian matrix and $<a,b>$ is the inner product operator between vectors a and b . The sources of the forward wavefields are f and Φ_w, and $N_x \times N_z$ represents the model size.

COMPUTING THE DERIVATIVE OF THE WAVEFIELD u

The wavefield u produced by a seismic source that propagates through an acoustic and isotropic medium can be modeled in two-dimensions for constant density as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} - \frac{1}{v^2(x,z)}\frac{\partial^2 u}{\partial t^2} = f(x,z) \tag{2}$$

where $v(x,z)$ is the acoustic velocity of the medium as a function of the spatial variables x and z; u denotes the scalar pressure field, t is the time variable, and $f(x,z)$ represents the seismic source. The wavefield u can be numerically obtained by using a finite-difference stencil in time domain (FDTD), with the initial conditions given by $u(x,z,0) = 0$, $\partial u/\partial t(x,z,0) = 0$. The non-natural boundaries (left, right and bottom sides) of the FDTD numerical implementation are handled using Convolutional Perfectly Matched Layer (CPML), proposed by [12].

Equation 2 can be re-written in a general form as

$$L(u,v) = f \tag{3}$$

where L represents the acoustic wave equation. The derivative of Equation 3 with respect to the model parameters is given by

$$\frac{\partial(L(u,v)-f)}{\partial v} = \frac{-2}{v^3(x,z)}\frac{\partial^2 u}{\partial t^2} \tag{4}$$

Once the auxiliary field $\partial^2 u/\partial t^2$ has been computed, the solution for Equation 4 is easily obtained by multiplying the auxiliary field by the scale factor $-2/v^3(x,z)$. This is shown in steps 1 and 2 of Algorithm 1.

Table 1. Algorithm 1 SOASM for Hessian Matrix Computation.

STEPS
Compute L(u,v)=f.
1: $\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial z^2} - \dfrac{1}{v^2(x,z)}\dfrac{\partial^2 u}{\partial t^2} = f(x,z)$
Compute $\dfrac{\partial(L(u,v)-f)}{\partial v}$
2: $\dfrac{-2}{v^3(x,z)}\dfrac{\partial^2 u}{\partial t^2}$
Compute $L(u_1^\dagger,v)=f^\dagger$
3: $\dfrac{1}{v^2(x,z)}\dfrac{\partial^2 u_1^\dagger}{\partial t^2} = \dfrac{\partial^2 u_1^\dagger}{\partial x^2}+\dfrac{\partial^2 u_1^\dagger}{\partial z^2}+\text{flipud }(d_{mod}-d_{obs})$
Compute $\dfrac{\partial(L(u_1^\dagger,v)^\dagger+R^\dagger(d_{mod}-d_{obs}))}{\partial v}$
4: $\dfrac{-2}{v^3(x,z)}\dfrac{\partial^2 u_1^\dagger}{\partial t^2}$
Compute $\Phi_w=-\sum_{j=1}^k w_i(j)\dfrac{\partial(L(u,v)-f)}{\partial v}$
5: $\Phi_w=-\sum_{j=1}^{N_x \times N_z} w_i(j)\left(-\dfrac{2}{v^3(x,z)}\dfrac{\partial^2 u}{\partial t^2}\right)$
Compute $L(\alpha,v)=\Phi_w$
6: $\dfrac{1}{v^2(x,z)}\dfrac{\partial^2\alpha}{\partial t^2}=\dfrac{\partial^2\alpha}{\partial x^2}+\dfrac{\partial^2\alpha}{\partial z^2}-\sum_{j=1}^{N_x \times N_z} w_i(j)\left(-\dfrac{2}{v^3(x,z)}\dfrac{\partial^2 u}{\partial t^2}\right)$
Compute $\dfrac{\partial(L(a,v)-\Phi_w)}{\partial v}$
7: $\dfrac{-2}{v^3(x,z)}\dfrac{\partial^2\alpha}{\partial t^2}+\sum_{j=1}^{N_x \times N_z} w_i(j)\left(\dfrac{6}{v^4(x,z)}\dfrac{\partial^2 u}{\partial t^2}\right)$
Compute $L(u_2^\dagger,v)=-\sum_{j=1}^k w_i(j)\left(-\dfrac{2}{v^3(x,z)}\dfrac{\partial^2 u_1^\dagger}{\partial t^2}\right)-R^\dagger(R\alpha)$
8: $\dfrac{1}{v^2(x,z)}\dfrac{\partial^2 u_2^\dagger}{\partial t^2}=\dfrac{\partial^2 u_2^\dagger}{\partial x^2}+\dfrac{\partial^2 u_2^\dagger}{\partial z^2}-\sum_{j=1}^{N_x \times N_z} w_i(j)\left(-\dfrac{2}{v^3(x,z)}\dfrac{\partial^2 u_1^\dagger}{\partial t^2}\right)-R^\dagger(R\alpha)$
Compute $\dfrac{\partial^2(L(u,v)-f)}{\partial v^2}$
9: $\dfrac{6}{v^4(x,z)}\dfrac{\partial^2 u}{\partial t^2}$
Compute $H(v)w_i$
10: $H(v)w_i=\langle\text{Step8},\text{Step2}\rangle+\langle\text{Step3},\text{Step7}\rangle+\sum_{j=1}^{N_x \times N_z} w_i(j)\langle\text{Step3},\text{Step9}\rangle$

COMPUTING THE ADJOINT WAVEFIELD $u_1^\dagger$

Let the adjoint wavefield $u_1^\dagger$, in operator form, be given by

$$L(u^\dagger,v)=f^\dagger \qquad (5)$$

where $f^\dagger$ is defined in Equation 7. The derivative of previous operator with respect to the model parameters v is then

$$\frac{\partial(L(u_1^\dagger,v)-f^\dagger)}{\partial v}=\frac{-2}{v^3(x,z)}\frac{\partial^2 u_1^\dagger}{\partial t^2}. \qquad (6)$$

As the acoustic and isotropic wave equation (Equation 2) is used for computing the forward wavefield u, its adjoint wavefield can

be computed using the same wave equation, because the system is self-adjoint. However, the propagation direction changes and the adjoint wavefield $u_1^\dagger$ represents a back-propagation, meaning that the final conditions of the problem are defined as $u_1^\dagger(x,z,T)=0$ and $(\partial u_1^\dagger)/\partial t(x,z,T)=0$, for a final time T.

In order to compute the adjoint wavefield $u_1^\dagger$, the receivers become sources (adjoint source) and the real source disappears as a consequence of the least square minimization of the cost function. The mathematical expression for the new set of sources is given by

$$f^\dagger = -R^\dagger(d_{mod}-d_{obs}) \qquad (7)$$

where d_{obs} represents the measured pressure wavefield at the surface level (seismic traces), d_{mod} are the modeled seismic traces and, $R^\dagger(\cdot)$ represents the operator to flip the vector of residuals. Note that the residuals, used as sources, are injected from the final sample to the first sample. This is shown in steps 3 and 4 of Algorithm 1.

COMPUTING THE DERIVATIVE OF α

Before obtaining $\partial(L(\alpha,v)-\Phi_w)/\partial v$, the auxiliar source Φ_w, and the wavefield α should be defined. Let Φ_w be an auxiliary source, defined by

$$\Phi_w = -\sum_{j=1}^{k} w_i(j)\frac{\partial(L(u,v)-f)}{\partial v},$$
$$\Phi_w = -\sum_{j=1}^{N_x\times N_z} w_i(j)\left(-\frac{2}{v^3(x,z)}\frac{\partial^2 u}{\partial t^2}\right) \qquad (8)$$

where N_x is the number of grid points in the x direction, N_z is the number of points in the z direction, and $w_i(j)$ is a perturbation vector having zero-value in all but the jth coordinate. A non-zero value determines the location for the perturbation, which corresponds to one column vector of the Hessian matrix. If more than one perturbation is selected (i.e., more than one element in the vector $w_i(j)$ is different from zero) then the Hessian-vector-products compute a column that represents a linear combination of the selected Hessian matrix columns. The auxiliary wavefield α is a forward wavefield computed when the source is Φ_w, and the initial conditions are $\alpha(x,z,0)=0$ and $\partial\alpha(x,z,0)/\partial t=0$.

The derivative of the wavefield α with respect to the model parameters v, is given by

$$\frac{\partial(L(\alpha,v)-\Phi_w)}{\partial v}=$$
$$\frac{-2}{v^3(x,z)}\frac{\partial^2\alpha}{\partial t^2}+\sum_{j=1}^{N_x\times N_z} w_i(j)\left(\frac{6}{v^4(x,z)}\frac{\partial^2 u}{\partial t^2}\right) \qquad (9)$$

Equation 9 represents the interaction between the forward field α and the perturbation of the geophysical model. This is shown in steps 5, 6 and 7 of Algorithm 1.

COMPUTING THE ADJOINT WAVEFIELD $u_2^\dagger$

The adjoint wavefield $u_2^\dagger$ is computed using as source

$$-\sum_{j=1}^{N_x \times N_z} w_i(j)\left(-\frac{2}{v^3(x,z)}\frac{\partial^2 u_1^\dagger}{\partial t^2}\right) - R^\dagger(R\alpha) \tag{10}$$

where $R\alpha$ is the pressure field at the surface, obtained with the auxiliary field α. This means that the adjoint wavefield $u_2^\dagger$ uses the number of receivers and the perturbations of the geophysical model as sources. Also, the final conditions are $u_2^\dagger$ (x,z,T)=0 and $((\partial u_2^\dagger)/\partial t)$ (x,z,T)=0. Note that $(\partial^2 u_2^\dagger)/(\partial t^2)$ was obtained previously.

The computation of the adjoint wavefield $u_2^\dagger$ is shown in step 8 of Algorithm 1.

COMPUTING THE SECOND DERIVATE OF THE WAVEFIELD u

The second derivative of the forward wavefield u for the acoustic and isotropic case is given by

$$\frac{\partial^2 (L(u,v)-f)}{\partial v^2} = \frac{6}{v^4(x,z)}\frac{\partial^2 u}{\partial t^2} \tag{11}$$

This is shown in step 9 of Algorithm 1.

Finally, as the terms in Equation 1 are known and the columns of the Hessian matrix can be obtained. A physical interpretation of each kernel in Equation 1 was presented by Fichtner [10].

IMPLEMENTATION OF THE SOASM ON GPUS

The computational complexity of the wave propagation modeling is in the order of $\mathcal{O}(N^3)$ for a squared synthetic model with N^2 elements and N time steps. However, if the same task is performed on a GPU then the execution time is reduced, as if the number of operations were $\mathcal{O}(N)$, due to the parallelism level at each time step [11].

Now, according to the SOASM algorithm, the modeling of four wavefields ($u, \alpha, u_1^\dagger$ and $u_2^\dagger$) is needed to compute one column of the Hessian matrix. Furthermore, as a model of N^2 elements produces a Hessian matrix of N^2 columns, then the full Hessian matrix has a computational cost of $\mathcal{O}(4N^5)$ for the SOASM. Nonetheless, when the SOASM is implemented on a GPU, the execution time is reduced as if the number of operations were $\mathcal{O}(4N^3)$. The reduction of two orders of magnitude in the execution time of the full Hessian matrix makes it feasible for uncertainty quantification.

UNCERTAINTY QUANTIFICATION

This section describes the link between the posterior covariance and the Hessian matrix using Bayesian inference. The posterior probability function of the model v given the observations d_{obs} can be expressed as

$$\rho(v|d_{obs}) \propto \rho(v)\rho(d_{obs}|v) \tag{12}$$

where $\rho(v)$ is the *a priori* density function of the model parameters and $\rho(v|d_{obs})$ is the likelihood function that relates the observations with the given model parameters. If the likelihood and the *a priori* functions follow Gaussian multivariate distributions, and the relation between the observations and the model parameters is linear, then the posterior probability function $\rho(v|d_{obs})$ is also Gaussian [3], and it is given by

$$\rho(v|d_{obs}) \sim \mathcal{N}(\tilde{v}, \sum_{post}) \tag{13}$$

where $\tilde{v}$ is the maximum likelihood solution of the model parameters and Σ_{post} is the posterior covariance. The posterior covariance matrix can be obtained as a function of the Hessian matrix of the cost function H, and the a priori covariance matrix Σ_{prior}, as follows

$$\Sigma_{post} = (H + \Sigma_{prior}^{-1})^{-1} \tag{14}$$

The a *priori* covariance matrix is chosen such that the correlations between model parameters are included. Assuming that the model parameters are uncorrelated and identically distributed, the a *priori* covariance matrix is given by

$$\sum_{prior} = \sigma_{prior}^2 \times I \tag{15}$$

where I is the identity matrix, and the elements of the diagonal are the variance σ_{prior}^2. Even though this is not a realistic model for the prior correlation of the model parameters, because usually parameters located at neighboring positions belong to the same class, the computation of the inverse Σ_{prior}^{-1} is simple. Despite the assumption of uncorrelated model parameters in the prior density function of the model parameters, it is possible that the posterior covariance matrix has elements off-diagonal that are nonzero. The posterior correlation between the parameters $\tilde{v}_i$ and $\tilde{v}_j$, means that they were not independently resolved by the observed data during the full waveform inversion.

3. EXPERIMENTAL DEVELOPMENT

In this section, we describe the experiment that has been formulated to evaluate the Hessian matrix computation for the uncertainty quantification of the velocity model obtained after multi-scale FWI. We also evaluate the computational performance of obtaining the Hessian matrix via SOASM using GPUs.

EXPERIMENT DESCRIPTION

The synthetic true velocity model used to obtain the observed data d_{obs} is depicted in **Figure 1**. The velocity model is of size $N_x \times N_z$=211×68, with a spatial grid of 25 m ($\Delta x = \Delta z = 25$ m). The background velocity is 2000 m/s and the diffracting squared area has a velocity of 2500 m/s. The diffracting element is a square of size 9×9 centered at the position x = 2650 m and z = 850 m. The observed and modeled data were produced using a set of 171 receivers placed in line at a depth of 125 m every 25 m, starting from the position 525 m to the position 4775 m. Each receiver recorded 3.5 s at a time step of 4 ms (Δt=4e -3s).
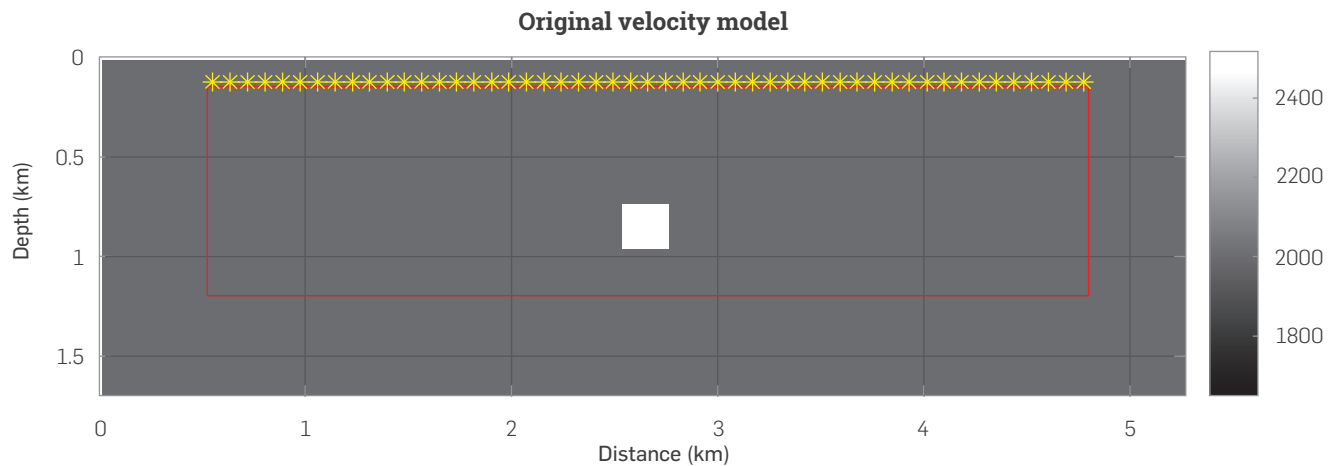
Original velocity model

Different number of sources were located at z=125 m and all of them have the source wavelet defined by

$$g_s = -2\pi f_0(t - t_0)\exp(-2\pi f_0(t - t_0)^2) \qquad (16)$$

where f_0 Hz is the central frequency, and t_0 is a time delay parameter. All the parameters of the numerical implementation were set to satisfy the Courant's stability criterion with a $CPML_{area}$=20 (grid points). Please refer to [12] for the definition of the variables used in the CPML.

The velocity model used as starting point for a multi-scale FWI approach [13] is the same as **Figure 1** but without the diffracting square (white square), where the central frequencies of the source wavelet were set to 3 Hz, 6 Hz and 9 Hz. The implementation uses a second and eighth order stencil of FDTD for approximating the time and spatial derivatives, respectively. We compute the gradient through the inner product between the derivative of the forward modeling and the residuals back propagation as it is proposed by Plessix [14] in Equation 36. Besides, the gradient computation uses CUDA-C for GPUs [15]-[16]. For the first FWI iteration, we used linear interpolation to compute the step forward, and for the subsequent 39 iterations, we used L-BFGS [17]. The resulting velocity model of the first inversion (f_0=3Hz) becomes the starting model for the second inversion (f_0=6 Hz). This process is repeated for the last frequency (f_0=9 Hz). The sources were placed at a depth of S_z=125 m and their spatial location were obtained using the following equation

$$S_x = S_R \cdot \lfloor a + \frac{(b - a)}{(n - 1)} \cdot k \rfloor \quad k = \{0, 1, 2, \cdots, n-1\} \quad (17)$$

where a is the grid position of the first source (a=21), b the grid position of the last source (b=191), n the number of sources (n=1,51), S_R the spatial resolution (S_R=25 m) and $\lfloor \rfloor$ represents the nearest integer operator.

The estimated velocities for the multi-scale process for one and fifty-one sources are depicted in **Figure 2a** and **Figure 2b**, respectively. In **Figure 2** can be noticed that the estimated velocity for the diffracting square is closer to the true velocity as the number of sources increases. In fact, the FWI method is able to estimate the velocities inside the diffracting area with at least 51 sources. However, the zone below the diffracting area cannot be updated correctly during the inversion due to the acquisition geometry.

The Hessian matrix associated to the velocity model estimated with the FWI multi-scale approach for one source is shown in **Figure 3**. Note that the Hessian matrix, obtained via SOASM, differs from the one given by Virieux (see **Figure 2a** in [1]) because it is not symmetric. The lack of symmetry is a consequence of using superficial sources and receivers only. To obtain a symmetric Hessian matrix, an array of sources and receivers should be located enclosing the area of interest [1], but this is not feasible in real seismic acquisitions. Also, the resulting Hessian matrix is not positive definite, which can be explained by the fact that in a surface acquisition, we do not have sufficient observations to build a high-dimensional Hessian (or covariance) matrix from a reduced number of measurements. This can also be evidenced through the magnitude of the eigenvalues of the Hessian matrix, which rapidly decay to zero (see **Figure 4**).

UNCERTAINTY QUANTIFICATION

The posterior covariance matrix can be obtained using Equation 14 for the estimated velocity models with a multi scale FWI method. The Hessian matrix H is obtained via SOASM and the *a priori* covariance is obtained assuming uncorrelated pixels, as expressed in Equation 15.

For the estimated models using 1 and 51 sources, shown in **Figure 2**, the uncertainty of the model can be quantified using three different Hessian matrices, each associated to a central wavelet
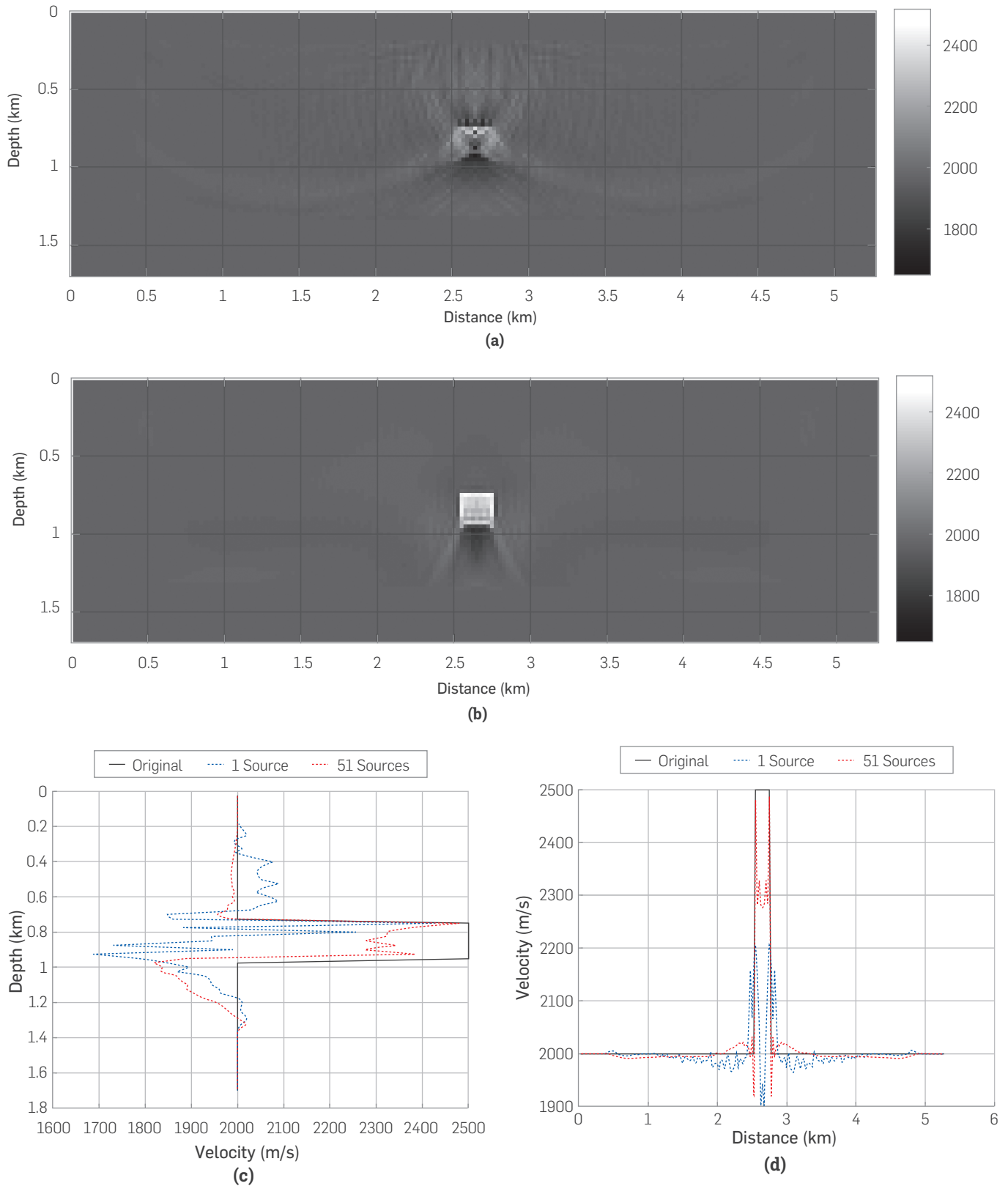
Figure 2. Multi-scale FWI results using the central frequencies f_0=3, 6 and 9 Hz with 40 iterations per frequency step. Results for (a) 1 source and (b) 51 sources with the vertical log (c) taken at x=2.65 Km and the horizontal log (d) placed at z=0.85 Km of depth.
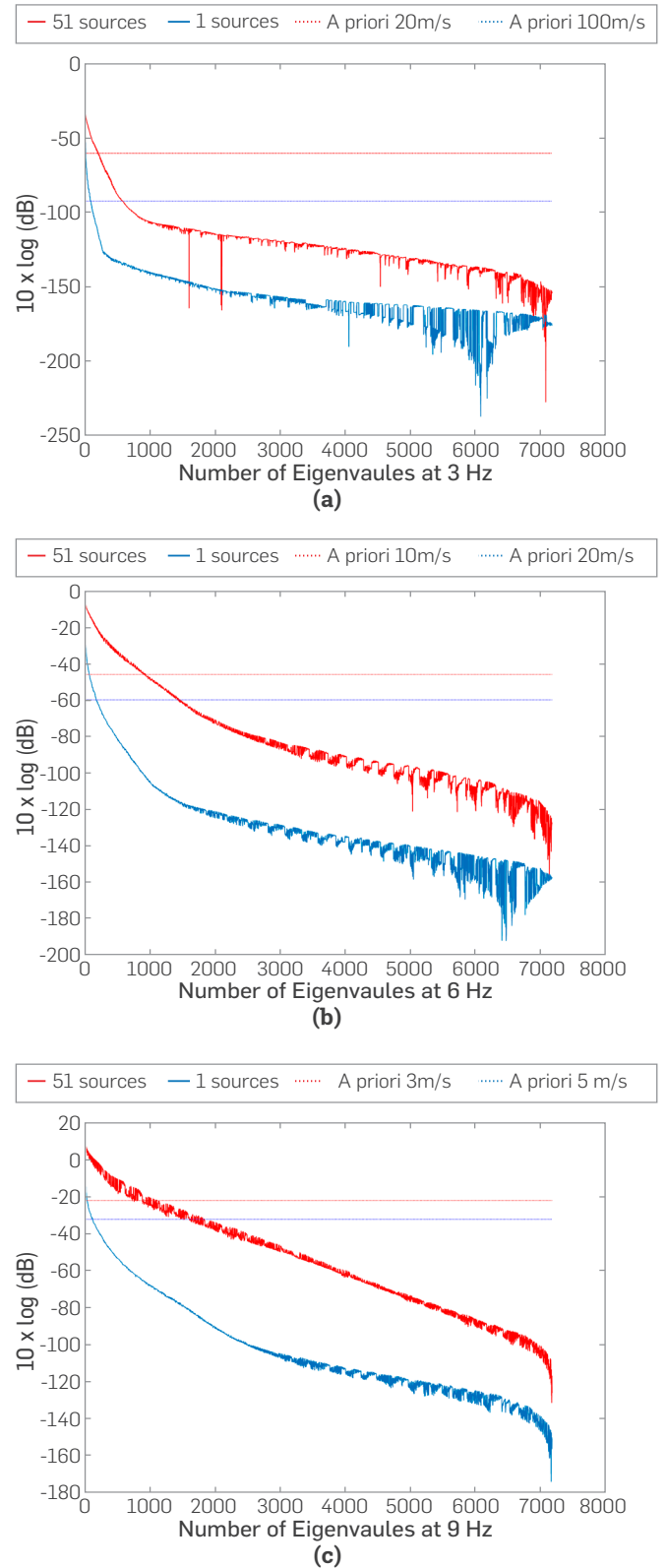
One source 3 Hz
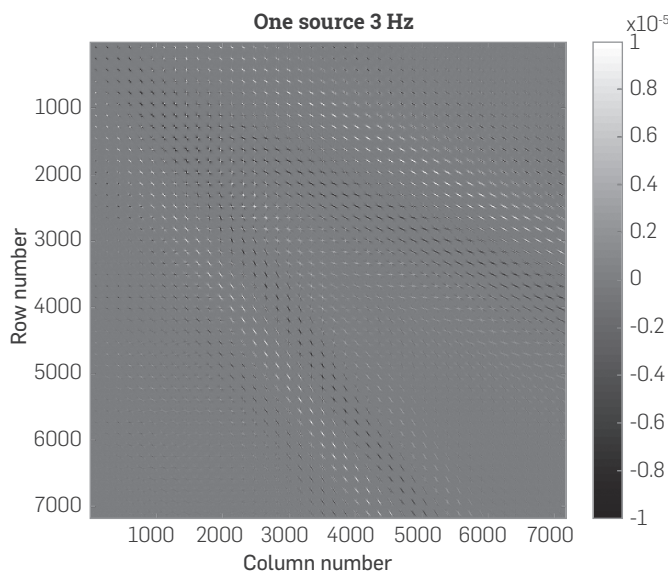
Figure 3. Hessian matrix for the velocity model estimated with multi-scale FWI using 1 source and 171 receivers (see Figure 2a). This Hessian matrix describes the interaction of the 7182 elements inside the red square (see Figure 1).

frequency (3, 6 and 9 Hz). The posterior covariance matrices Σ_{post} are obtained by setting the a *priori* standard deviation for all pixels in σ_a=3, 5, 10, 20 and, 100 m/s. **Figure 5** and **Figure 6** depict the standard deviation of each pixel in the model, for one and fifty-one sources, which correspond to the square diagonal elements of the posterior covariance matrix Σ_{post}. Note in **Figure 5** and **Figure 6** that the posterior variance decreases in comparison to the prior variance, in the area that is properly illuminated by the sources. For only one source, the illumination angle is smaller than for 51 sources, and thus only a small portion of the model is properly updated during the inversion process. Also, note that for the smaller frequency (3 Hz), the uncertainty can only be quantified coarsely, whereas for the larger frequency (9 Hz), the uncertainty can be quantified at a level of pixel.
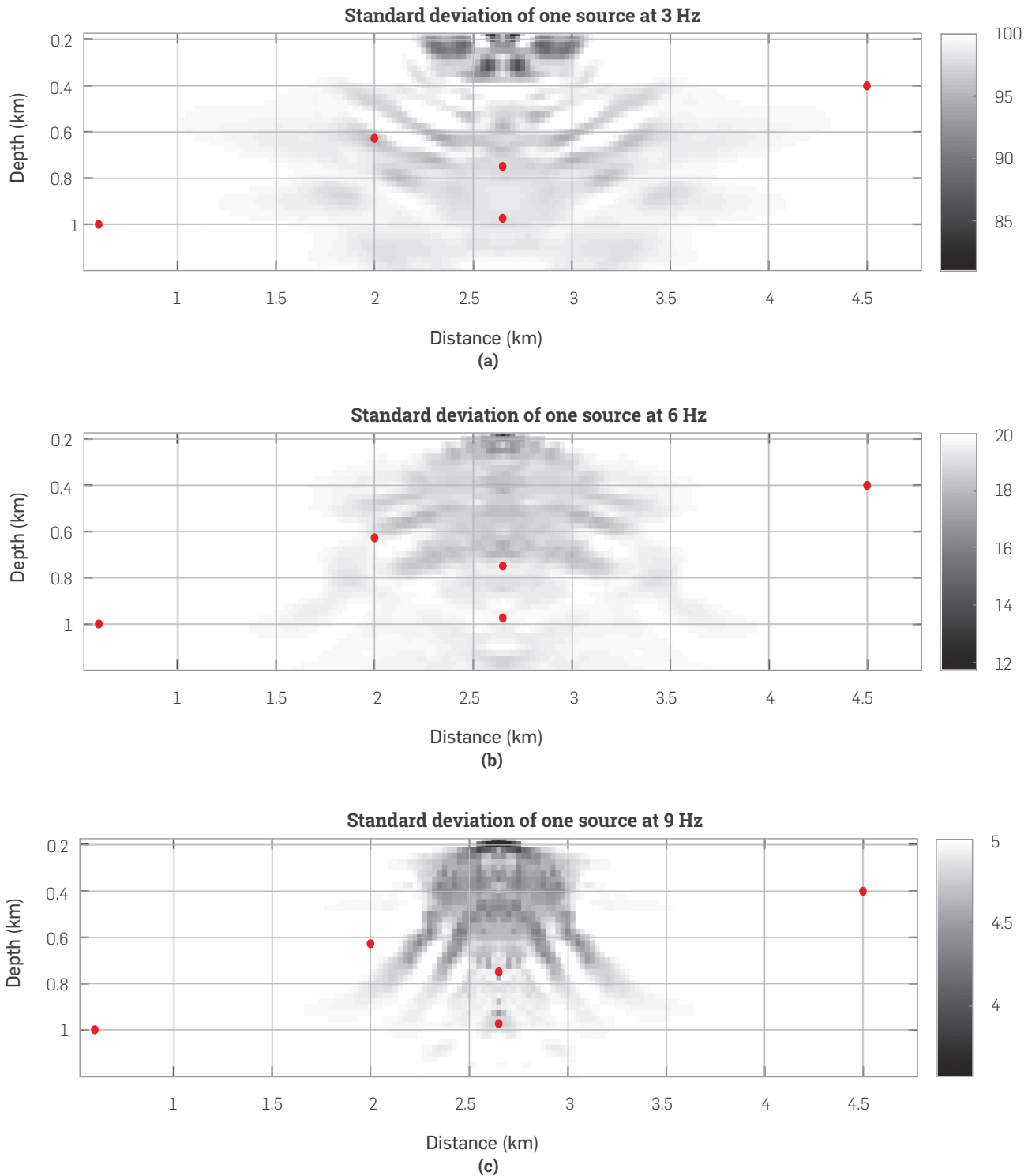
In addition, the uncertainty of the prior velocity model is reduced due to the "new" information brought by the observations in the inversion process. The comparison between posterior uncertainties and prior uncertainties can also be quantified. The uncertainty reduction of the ith pixel in the model, after the multi-scale FWI is measured as

$$VR_i = \frac{\sigma_{prior}^2 - \sigma_{p_i}^2}{\sigma_{prior}^2} \qquad (18)$$

where σ_{pi}^2 is the posterior variance of the ith pixel in the model. A value of VR_i=0 or negative means that no-new information was brought from the observed data on the ith pixel. A positive value of VR_i means that the inversion method added new information on the ith pixel and thus the uncertainty of the posterior variance σ_{pi}^2 is smaller than the a priori variance.

Information that complements the FWI results. The Hessian matrix

Figure 4. Eigenvalues of the Hessian Matrices computed at (a) 3 Hz, (b) 6~Hz and (c) 9~Hz for one source and fifty-one sources. These eigenvalues describe the interaction of the 7182 elements inside the red square (see Figure 1).

Standard deviation of one source at 3 Hz

(a)

Standard deviation of one source at 6 Hz

(b)

Standard deviation of one source at 9 Hz

(c)

Figure 5. Posterior standard deviations, computed using Equation (14) at (a) 3 Hz, (b) 6 Hz and, (c) 9 Hz, for the velocity model estimated with 1 source (see Figure 2a). The Hessian matrices are computed using the SOASM and the a priori covariances are obtained using Equation 15 with the a priori standard deviations: (a) 100 m/s, (b) 20 m/s and, (c) 5 m/s. The red dots are placed at five points with different illumination.
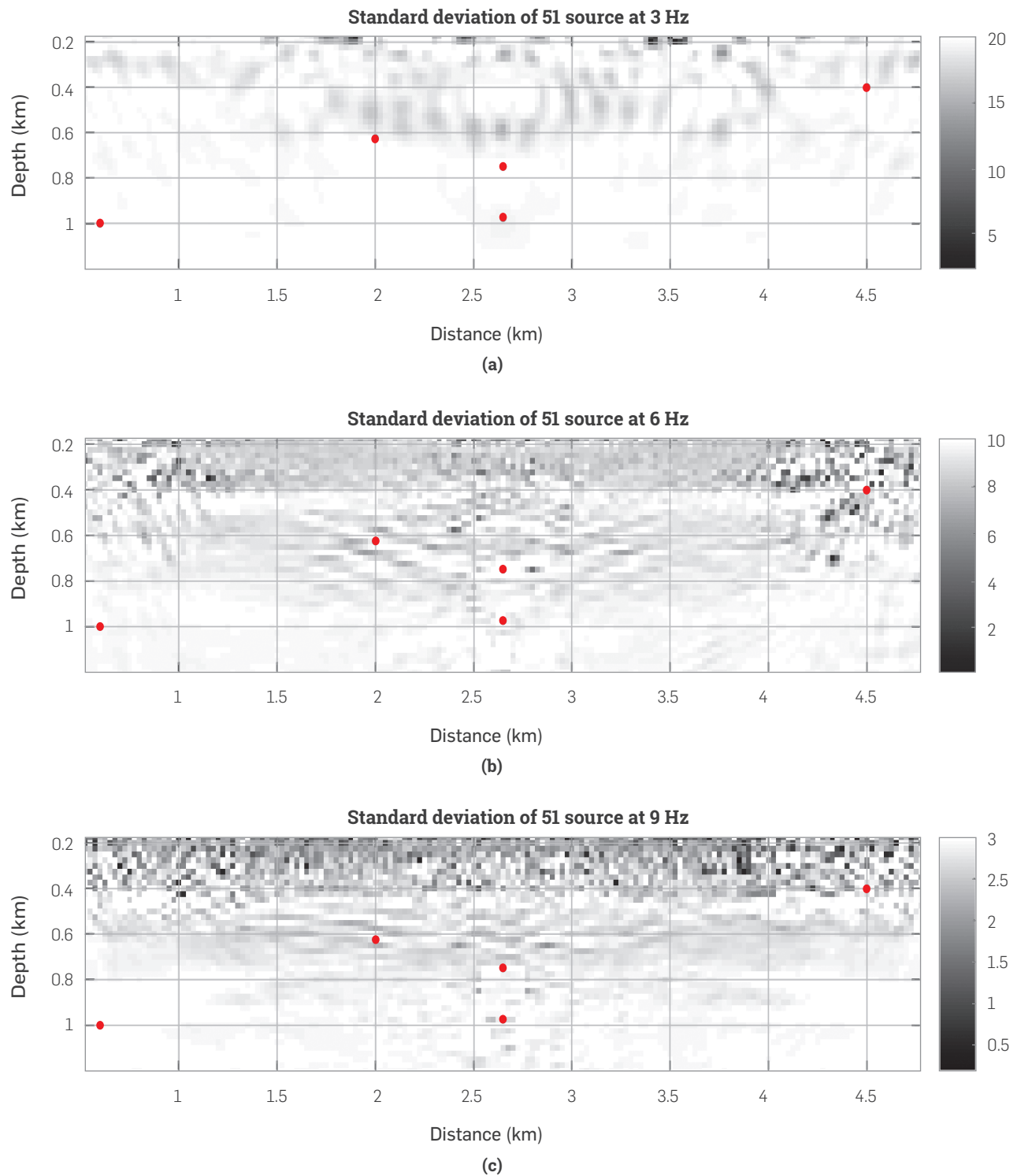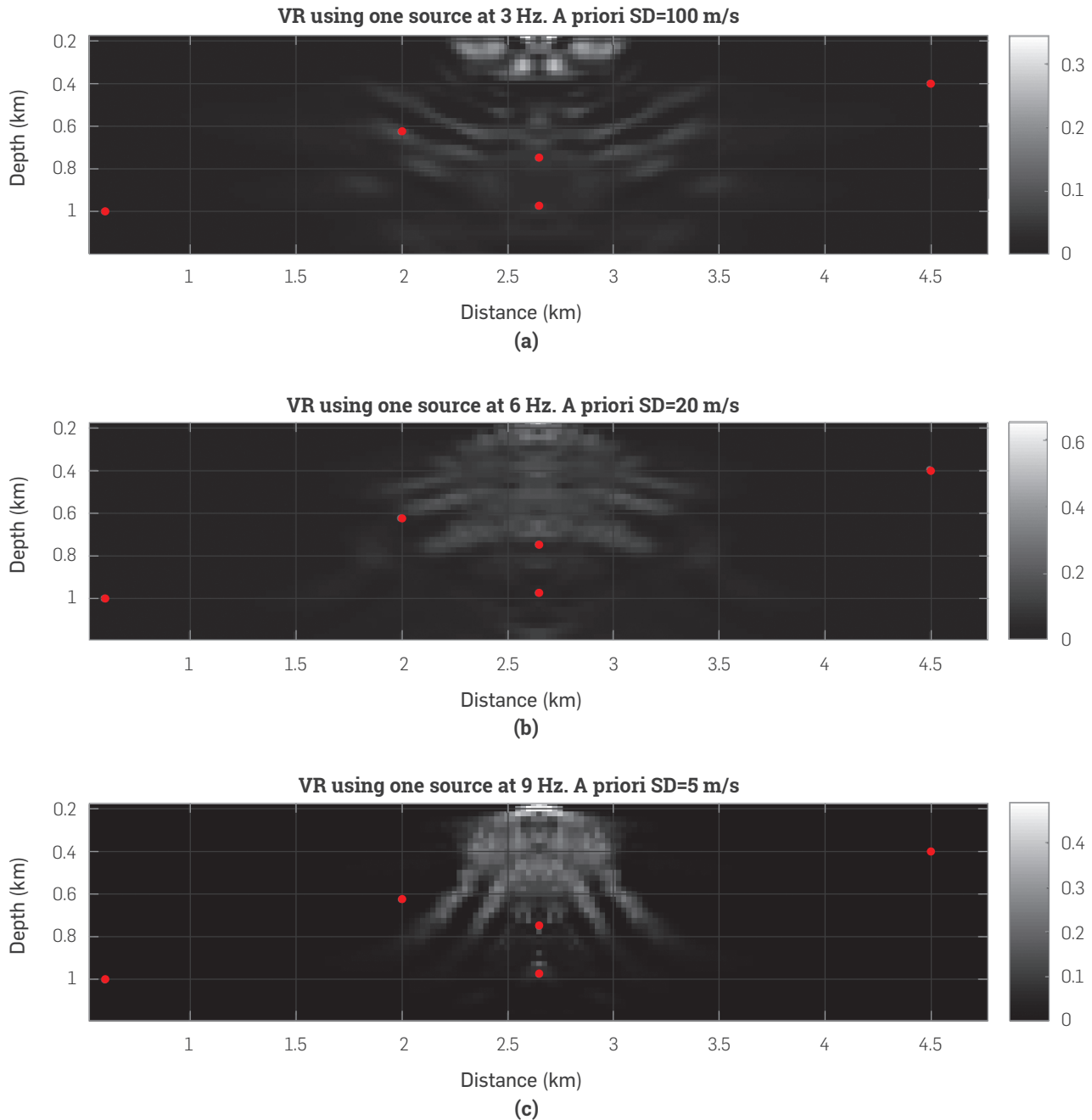
Figure 6. Posterior standard deviations, computed using Equation 14 at (a) 3 Hz, (b) 6 Hz and, (c) 9 Hz, for the velocity model estimated with 51 sources (see Figure 2b). The Hessian matrices are computed using the SOASM and the a priori covariances are obtained using Equation 15 with the a priori standard deviations: (a) 20 m/s, (b) 10 m/s and, (c) 3 m/s. The red dots are placed at five points with different illumination.

is mainly affected by the illumination area produced by sources and receivers, the source **Figure 7** and **Figure 8** illustrate the variance reduction. The uncertainty reduction given in **Figure 7a** shows that only a few parameters in the model have been "correctly" resolved. This occurs since the posterior covariance is obtained using a low frequency wavelet as source, which quantifies the uncertainty of a group of neighboring pixels instead of the uncertainty of one single pixel at the time. **Figure 7b** and **Figure 7c** show the uncertainty reduction using sources with higher frequencies. Particularly, **Figure 7c** shows the reduction in the uncertainty for every single parameter in the model, and a higher reduction is obtained (as expected) in the area underneath the source.

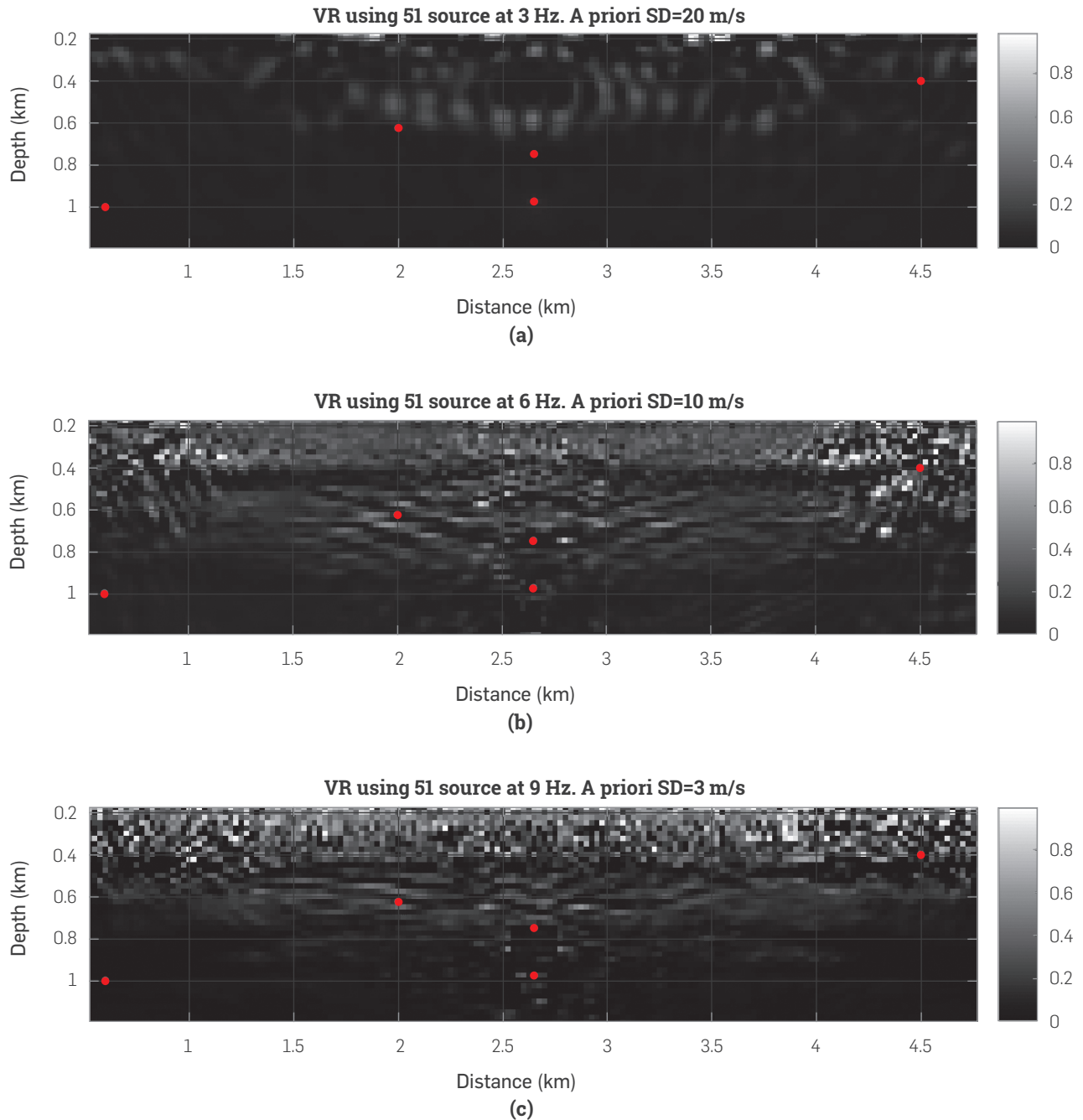The *a priori* standard deviations were selected to define the lowest

Figure 7. Variance reduction computed with Equation 18 for the posterior standard deviations illustrates in Figure 5. For these results the a priori variances were set to (a) 10000 [m/s]² for 3 Hz, (b) 400 [m/s]² for 6 Hz and (c) 25 [m/s]² for 9 Hz. The red dots are placed at five points with different illumination.

eigenvalues of the Hessian matrices accepted to compute the posterior covariance matrices. As the eigenvalues behavior vary when both the number of sources and the source frequency change then each experiment uses its own *a priori* standard deviation (see **Figure 4**). If the *a priori* standard deviation is set to a very small value then the information provided by the Hessian matrix is almost neglected for the posterior covariance. On the other hand, a large *a priori* standard deviation value produces a posterior

covariance matrix that can be affected by the small eigenvalues of the Hessian matrix generating undesired effects (e.g. complex, negative or extremely high variance values). For the six tests, the *a priori* variances were selected to obtain as much information as possible from the Hessian matrices avoiding the undesired effects of the small eigenvalues.

Five pixels in the velocity model were selected to identify areas

(a)

(b)

(c)

Figure 8. Variance reduction computed with Equation 18 for the posterior standard deviations illustrates in Figure 6. For these results the a priori variances were set to (a) 400 [m/s]² for 3 Hz, (b) 100 [m/s]² for 6 Hz and (c) 9 [m/s]² for 9 Hz. The red dots are placed at five points with different illumination.

with low illumination (P_1=(0.6,1) km and P_2=(4.5,0.4) km), with high illumination and under a high velocity contrast area (P_3=(2.65,0.975) km) and with high illumination over a high velocity area (P_4=(2,0.625) km and P_5=(2.65,0.75) km). The true velocity value for P_1,P_2,P_3 and P_4 is 2000 m/s (outside the diffracting square) and for P_5 is 2500 m/s (inside the diffracting square). The pixels are marked in red color in **Figure 5**, **Figure 6**, **Figure 7**, and **Figure 8**. For all the tests P_1, P_2 and P_3 were never updated as a consequence of the lack of illumination or the interference produced by the diffracting square. The standard deviation for P_4 and P_5 are affected by the source frequency, the number of sources and $\sigma_{P_{rior}}$ as **Figure 5** and **Figure 6** illustrates.

COMPUTATIONAL PERFORMANCE

The cost of computing the full Hessian matrix using FDTD, in terms of execution time and memory resources, is high since four wavefield propagations need to be obtained for each column. In particular, we evaluate the performance of computing a single column when the FDTD approximation uses a second order stencil for time derivative and an eighth order stencil for the spatial derivatives. Two implementations are evaluated: (1) A serial ANSI-C implementation on an Intel(R) Xeon(R) E5-2609 and (2) a CUDA-C implementation on a Nvidia(R) GPU K40c. The execution times of one single column of the Hessian matrix are presented in **Table 1**. Besides, **Table 2** summarizes the execution times and RAM requirements by the CUDA-C implementation to compute the full Hessian matrix. Note in **Table 2** that increasing the number of sources does not increase the RAM required, but instead the computational cost increases linearly with the number of sources. The execution times of the full Hessian matrix for the serial ANSI-C implementation are not presented in **Table 2**, as these are not computationally feasible.

Table 2. Execution times required for computing one-column of the Hessian matrix in CPU (Intel(R) Xeon(R) E5-2609) and GPU (Tesla K40c).

Architecture	Language	Exec. Time [s]
Intel(R) Xeon(R) E5-2609	ANSI-C	14.78
GPU K40c	CUDA-C	0.38

Table 3. Execution times and RAM size used to compute the full matrix **H** (7182×7182 elements) for different sources using a GPU Tesla K40c.

Source number	RAM (MB)	Exec. Time [HH:MM:SS]
1	206.3	00:46:00
51	206.3	39:06:00
101	206.3	77:26:00

4. RESULTS ANALYSIS

The Hessian matrix provides valuable information for resolution, uncertainty analysis and radiation patterns identification [18]. Therefore, computational strategies are needed to make feasible the use of the Hessian matrix. In this work, it has been demonstrated that GPU architectures make feasible the implementation of the second-order adjoint method because the execution time is reduced in two orders of magnitude.

This work presents a parallel implementation of the Hessian matrix-vector products using GPU architectures. The implementation uses the FDTD approximation to compute the wavefield propagations required by the SOASM.

We presented a practical method to compute the standard deviation over the final results (σ_p) from the a priori variance (σ_a^2) and H_{SOASM}. Also, a variance reduction (VR) is computed to define the areas with less uncertainty. It is shown that VR growths when the source frequency and the number of sources increase. However, a statistical analysis fails for P_1, P_2 and P_3 because they are not correctly illuminated.

According to **Table 2**, the computational cost of obtaining the Hessian matrix has a linear behavior when the number of sources change. However, the main concern about computing the Hessian matrix arises when the model size increases given that the computational cost for the 2D implementation is $\mathcal{O}(4N^5)$. To overcome this problem, a cluster of GPUs that divides the load over all the nodes can be used. Also, the computation of the Hessian matrix only for the area of interest decreases the computational cost.

The eigenvalues of a matrix are useful to identify, classify and compare set of matrices through characteristics as the condition number, the rank and the trace [19]. In the theoretical case, if the velocity model has been correctly resolved then the rank of the Hessian matrix evaluated at that point equals the number of model dimensions (i.e., $N_x \times N_z$) and the condition number equals one because all the eigenvalues have a magnitude close to one.

In this case, the theoretical behavior of the Hessian matrix near the solution is used as a reference to define whether the velocity models are improving after the FWI. **Figure 4** illustrates the eigenvalue magnitudes of the Hessian matrices computed from the FWI results at 3 Hz, 6 Hz and 9 Hz for one and fifty-one sources including the 7011 dimensions (171×41). The eigenvalues of all the Hessian matrices are non-zero and their magnitudes are above -250 dB. This produces full rank matrices (i.e., rank=$N_x \times N_z$=7182) for all the cases. Besides, if the number of sources increases from one to fifty-one then the eigenvalue magnitudes also increase. But, in all cases the results are still far from the theoretical behavior of the Hessian matrix because the condition numbers are much greater than one due to the ratio of the largest singular value to the smallest singular value is also much greater than one.

CONCLUSIONS

Hessian matrices are key for decision-making processes, as they provide new frequency used for the Hessian kernels computation, the resolution of the estimated velocity model and the numerical error of the implementation. Therefore, a deepest sensitivity analysis must be done on how the above parameters affect the Hessian matrix. Finally, our implementation makes it feasible to obtain the Hessian matrix for small models. Nonetheless, other computational strategies must be explored to reduce both RAM requirements and execution time, before the SOASM can be successfully applied on real data.

ACKNOWLEDGEMENTS

REFERENCES

[1] Virieux, J. and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26.

[2] Wu, Z. and Alkhalifah, T. (2016). The optimized gradient method for full waveform inversion and its spectral implementation. *Geophysical Journal International*, 205(3):1823–1831.

[3] Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation.* siam.

[4] Zhun, H., Li, S., Fomel, S., Stadler, G., and Ghattas, O. (2016). A bayesian approach to estimate uncertainty for full-waveform inversion using a priori information from depth migration. *Geophysics*, 81(5):307–323.

[5] Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782.

[6] Nocedal, J. and Wright, S. (2006). *Numerical optimization.* Springer Science & Business Media.

[7] Métivier, L., Brossier, R., Operto, S., Virieux, J., et al. (2012). Second-order adjoint state methods for full waveform inversion. In *EAGE 2012-74th European Association of Geoscientists and Engineers Conference and Exhibition.*

[8] Métivier, L., Brossier, R., Virieux, J., and Operto, S. (2013). Full waveform inversion and the truncated newton method. *SIAM Journal on Scientific Computing*, 35(2):B401–B437.

[9] Fichtner, A. (2010). *Full seismic waveform modelling and inversion.* Springer Science & Business Media.

[10] Fichtner, A. and Trampert, J. (2011). Hessian kernels of seismic data functionals based upon adjoint techniques. *Geophysical Journal International*, 185(2):775–798.

[11] Abreo, S., Ramirez, A., Reyes, O., Abreo, D., and Gonzalez, H. (2015). A practical implementation of acoustic full waveform inversion on graphical processing units. *CT&F - Ciencia, Tecnología y Futuro*, 6(2)(2):5–19.

[12] Pasalic, D., McGarry, R., et al. (2010). Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations. In *2010 SEG Annual Meeting.* Society of Exploration Geophysicists.

[13] Bunks, C., Saleck, F. M., Zaleski, S., and Chavent, G. (1995). Multiscale seismic waveform inversion. *Geophysics*, 60(5):1457–1473.

[14] 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(2):495–503.

[15] Mao, J., Wu, R.-S., Wang, B., et al. (2012). Multiscale full waveform inversion using gpu. In 2012 *SEG Annual Meeting.* Society of Exploration Geophysicists.

[16] Wang, B., Gao, J., Zhang, H., Zhao, W., et al. (2011). Cuda-based acceleration of full waveform inversion on gpu. In 2011 *SEG Annual Meeting.* Society of Exploration Geophysicists.

[17] Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

[18] Operto, S., Gholami, Y., Prieux, V., Ribodetti, A., Brossier, R., Metivier, L., and Virieux, J. (2013). A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice. *The Leading Edge*, 32(9):1040–1054.

[19] Martiartua, N. K., Boehma, C., Vinarda, N., Balicb, I. J., and Fichtner, A. (2017). Optimal experimental design to position transducers in ultrasound breast imaging. In *SPIE Medical Imaging*, pages 101390M–101390M. International Society for Optics and Photonics.