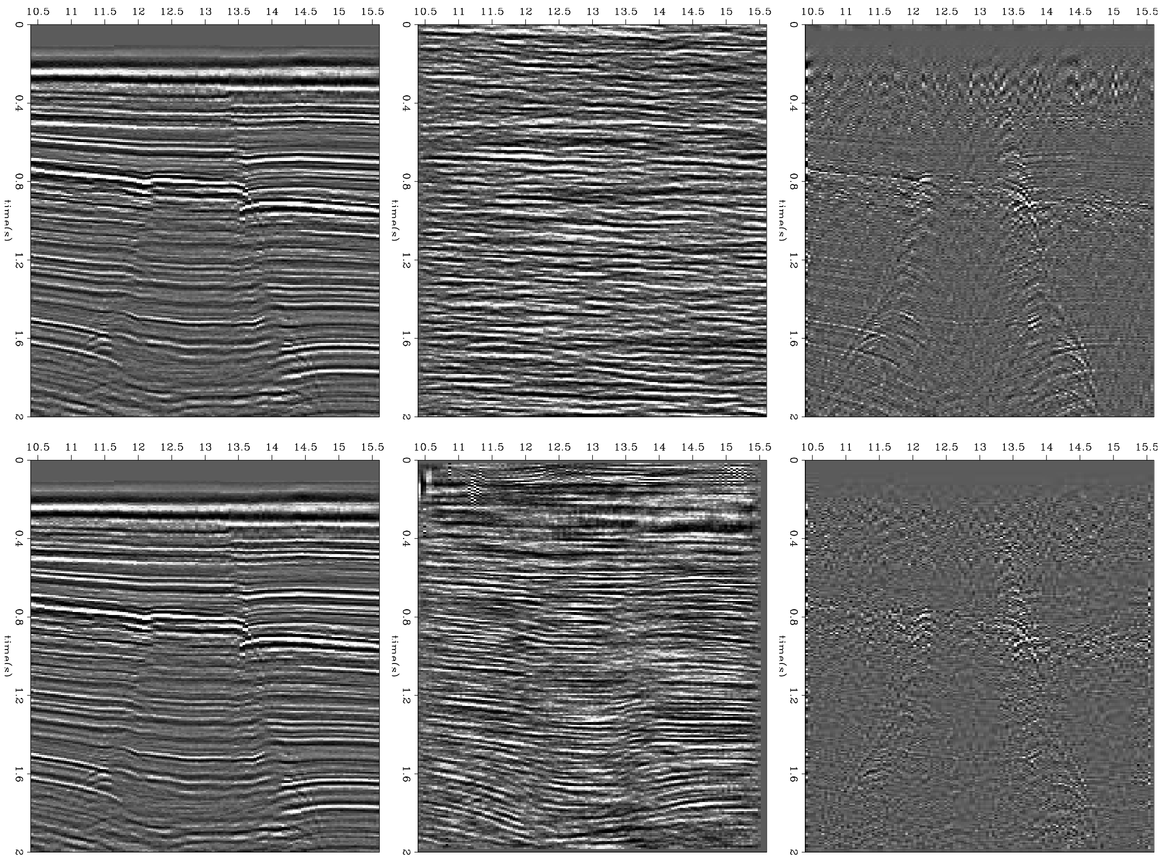
STANFORD EXPLORATION PROJECT

*Ali Almomin, Gustavo Alves, Biondo Biondi, Ettore Biondi, Alejandro Cabrales-Vargas,
Jason Chang, Jon Claerbout, Robert Clapp, Joseph Jennings, Stewart Levin, Yinbin Ma,
Musa Maharramov, Ossian O'Reilly, Shuki Ronen, Kaixi Ruan, and Yi Shen*

Report Number 160, September 2015



Preface

The electronic version of this report[1] makes the included programs and applications available to the reader. The markings [ER], [CR], and [NR] are promises by the author about the reproducibility of each figure result. Reproducibility is a way of organizing computational research that allows both the author and the reader of a publication to verify the reported results. Reproducibility facilitates the transfer of knowledge within SEP and between SEP and its sponsors.

ER denotes Easily Reproducible and are the results of processing described in the paper. The author claims that you can reproduce such a figure from the programs, parameters, and makefiles included in the electronic document. The data must either be included in the electronic distribution, be easily available to all researchers (e.g., SEG-EAGE data sets), or be available in the SEP data library[2]. We assume you have a UNIX workstation with Fortran, Fortran90, C, C++, X-Windows system and the software downloadable from our website (SEP makerules, SEPlib, and the SEP latex package), or other free software such as SU. Before the publication of the electronic document, someone other than the author tests the author's claim by destroying and rebuilding all ER figures. Some ER figures may not be reproducible by outsiders because they depend on data sets that are too large to distribute, or data that we do not have permission to redistribute but are in the SEP data library.

CR denotes Conditional Reproducibility. The author certifies that the commands are in place to reproduce the figure if certain resources are available. The primary reasons for the CR designation is that the processing requires 20 minutes or more, MPI or CUDA based code, or commercial packages such as Matlab or Mathematica.

NR denotes Non-Reproducible figures. SEP discourages authors from flagging their figures as NR except for figures that are used solely for motivation, comparison, or illustration of the theory, such as: artist drawings, scannings, or figures taken from SEP reports not by the authors or from non-SEP publications.

Our testing is currently limited to LINUX 2.6 (using the Intel compiler), but the code should be portable to other architectures. Reader's suggestions are welcome. More information on reproducing SEP's electronic documents is available online[3].

[1] http://sepwww.stanford.edu/private/docs/sep160

[2] http://sepwww.stanford.edu/public/docs/sepdatalib/toc_html

[3] http://sepwww.stanford.edu/research/redoc/

SEP160-2 — TABLE OF CONTENTS

Imaging and inversion

Modeling and attenuation

Velocity estimation

Case studies

Signal processing

Time-lapse inverse scattering theory

Musa Maharramov and Biondo L. Biondi

ABSTRACT

“Time-lapse inverse scattering theory” that we introduce in this paper focuses on recovering changes in physical models without accurate knowledge of model backgrounds. More specifically, we study the feasibility of recovering low and high-wavenumber components of model perturbation using the traditional Born and Rytov scattering approximations, and establish a connection between the Rytov approximation and phase-only full-waveform inversion (FWI). We provide a theoretical justification for applying regularized simultaneous time-lapse FWI to problems of applied seismology. We demonstrate the method’s sensitivity to realistic production effects in seismic data, and its stability with respect to inaccurate starting models.

INTRODUCTION

This work is dedicated to developing a systematic theory for solving *time-lapse* inversion problems that we subsequently refer to as *time-lapse inverse theory*. The term “time-lapse” relates to separate observations of physical phenomena taken at discrete time intervals. Inverse time-lapse theory concerns itself with estimating or *inverting* changes in the underlying physical models from such discrete observations.

Problems of time-lapse or “4D” seismic imaging and reservoir geomechanics that arise in Petroleum Industry(Johnston, 2013; Biondi et al., 1996) provide important applications for the developed theory, and are the primary target of our work as an exploration geophysicists. However, many fundamental concepts, constructs and ideas presented in this work, as well some mathematical, algorithmic and computational byproducts of this research, are applicable beyond the limits of exploration seismology and reservoir geomechanics. We envisage ubiquitous applications of these results to diverse problems of acoustic and electromagnetic inverse scattering, imaging sciences and large-scale numerical optimization. Exploration geophysicists are the primary and key audience of this work, however, we have attempted to make our narrative accessible to specialists in applied mathematics and mathematical physics. Although a considerable part of this work is dedicated to providing a robust and systematic theoretical background for the proposed inversion techniques, we have structured the material in such a way that a motivated practitioner can go straight to examples and case studies, as well as the nuts and bolts of specific algorithms as required for immediate practical application.

If we think of seismic time-lapse analysis as an estimation of *changes* in subsurface model parameters[1] that occurred between two separate seismic experiments, the “inverse time-lapse theory” can be simply regarded as a subset of the inverse acoustic scattering

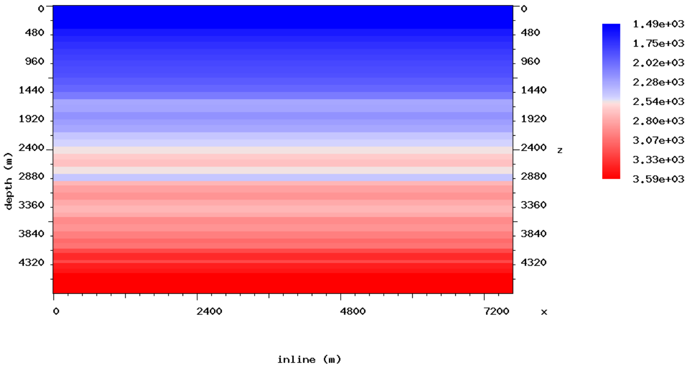
[1] as a result of petroleum production, fluid injection or environmental phenomena

theory. Indeed, classical scattering theory addresses the problem of estimating properties of a "scatterer"—a perturbation in the background model—from incident and scattered wave fields. If production-induced subsurface changes are regarded as a penetrable scatterer, then the 4D analysis simply becomes a problem of inverse scattering theory. If so, why develop a new "time-lapse" inverse theory?

The answer is quite simple. In practical time-lapse applications of exploration geophysics and beyond, the background model is not known accurately. Moreover, errors in our best estimates of the background model can be of the same magnitude as, or even exceed, the time-lapse effects that we seek to estimate. Is it even possible to estimate time-lapse changes when their magnitudes can be easily masked by the effects of measurement noise or errors in the background model? The main product of this research is a systematic theory of inverting small (and spatially bounded) time-lapse changes from noisy and insufficient observations. The proposed "time-lapse inverse theory" differs from inverse theory by placing emphasis on accurate estimation of *relative model changes* while ignoring errors in the background and perturbed models. We provide a toolkit of robust inversion techniques for accurate inversion of time-lapse changes, and demonstrate them on a synthetic example.

Effective methodologies exist for detecting production-induced reflectivity changes and translating them into impedance changes due to fluid substitution or reservoir compaction (Johnston, 2013; Biondi et al., 1996). However, resolving strains in the overburden from seismic data currently requires extraction of time shifts from cross-equalized surveys and mapping the estimated time-strains into the overburden (Rickett et al., 2007). Therefore, we make the primary emphasis of this paper achieving automated recovery of long-wavelength small-magnitude changes of the subsurface acoustic velocities caused by overburden dilation. Maharramov and Biondi (2014d, 2015a) proposed a method for a simultaneous multi-scale inversion of both low and high-wavenumber production anomalies. We demonstrate a hierarchical approach to multi-scale inversion for the recovery of both long-wavelength blocky overburden anomalies and short-wavelength reservoir effects.

Figure 1: The true baseline model. We chose a flat reflector model to study the sensitivity of FWI of short-offset reflection data to small velocity perturbation in the overburden. [**ER**] musa2/. truebase



EARLIER WORK

Prevalent practice in time-lapse seismic processing relies on picking time displacements and changes in reflectivity amplitudes between migrated baseline and monitor images, and converting them into impedance changes and subsurface deformation (Johnston, 2013).

This approach requires a significant amount of manual interpretation and quality control. One alternative approach uses the high-resolution power of full-waveform inversion (Sirgue et al., 2010) to reconstruct production-induced changes from wide-offset seismic acquisitions (Routh et al., 2012; Zheng et al., 2011; Asnaashari et al., 2012; Raknes et al., 2013; Maharramov and Biondi, 2014b; Yang et al., 2014; Maharramov et al., 2015a). However, while potentially reducing the amount of manual interpretation, time-lapse FWI is sensitive to repeatability issues (Asnaashari et al., 2012), with both coherent and incoherent noise potentially masking important production-induced changes. The joint time-lapse FWI proposed by Maharramov and Biondi (2013, 2014b) addressed repeatability issues by joint inversion of multiple vintages with model-difference regularization based on the L_2-norm and produced improved results when compared to the conventional time-lapse FWI techniques. Maharramov et al. (2015b) extended this joint inversion approach to include edge-preserving total-variation (TV) model-difference regularization. The new method was shown to achieve a dramatic improvement over alternative techniques by significantly reducing oscillatory artifacts in the recovered model difference for synthetic data with repeatability issues. Originally, the method was envisaged for applications to large-offset datasets where FWI is traditionally strong. However, Maharramov and Biondi (2015b); Maharramov et al. (2015a) applied this method in a Gulf of Mexico case study to resolving small $(1 - 2\%)$ production-induced velocity changes associated with overburden dilation. The approach used phase-only FWI of reflection-only data with 5 km maximum offset and target reflectors at about 4 km depth.

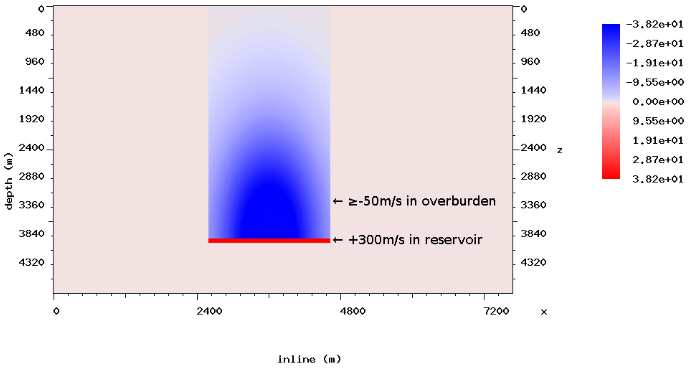


Figure 2: The true model difference is a combination of a positive +300 m/s velocity change in a target reflector at a depth of 3900 m, and a negative velocity change in the overburden above the reflector, peaking at −50 m/s. In this work we investigate the sensitivity of simultaneous time-lapse FWI to small and blocky velocity changes in the overburden. [**ER**] musa2/. truediff

While both synthetic and field data experiments involving joint time-lapse FWI with a model-difference regularization indicate robustness and broad applicability of the proposed technique, a detailed theoretical analysis of the joint inversion method is highly desirable for understanding its strengths and limitations.

THEORY

Assuming known background slowness $s(\mathbf{x}), \mathbf{x} \in \mathbb{R}^3$ and a slowness perturbation $\delta s(\mathbf{x})$, the total wavefield component $u(\mathbf{x})$ for frequency ω satisfies the Helmholtz equation

$$\left[\Delta + \omega^2(s(\mathbf{x}) + \delta s(\mathbf{x}))\right] u(\mathbf{x}) \;=\; -f(\mathbf{x}),\; \mathbf{x} \in D \subset \mathbb{R}^3, \tag{1}$$

where $f(\mathbf{x})$ is the seismic source component for frequency ω. The total wavefield is the sum of incident and scattered wavefields

$$u(\mathbf{x}) \; = \; u_I(\mathbf{x}) \; + \; u_S(\mathbf{x}), \tag{2}$$

where the incident wavefield u_I satisfies the Helmoholtz equation with the unperturbed slowness:

$$\left[\Delta + \omega^2 s(\mathbf{x})\right] u_I(\mathbf{x}) \; = \; -f(\mathbf{x}). \tag{3}$$

Note for well-posedness of (1) and (3) we need to impose an additional condition on the solution, such as the Sommerfeld radiation condition for a homogeneous medium (Colton and Kress, 1998). Physically, such a condition requires that the total field be outgoing at infinity. We will assume that equations (1) and (3) are solved in a domain $D \subset \mathbf{R}^3$, and absorbing boundary conditions (Engquist and Majda, 1977) are applied along the domain boundary, ensuring outgoing propagation of the wavefields.

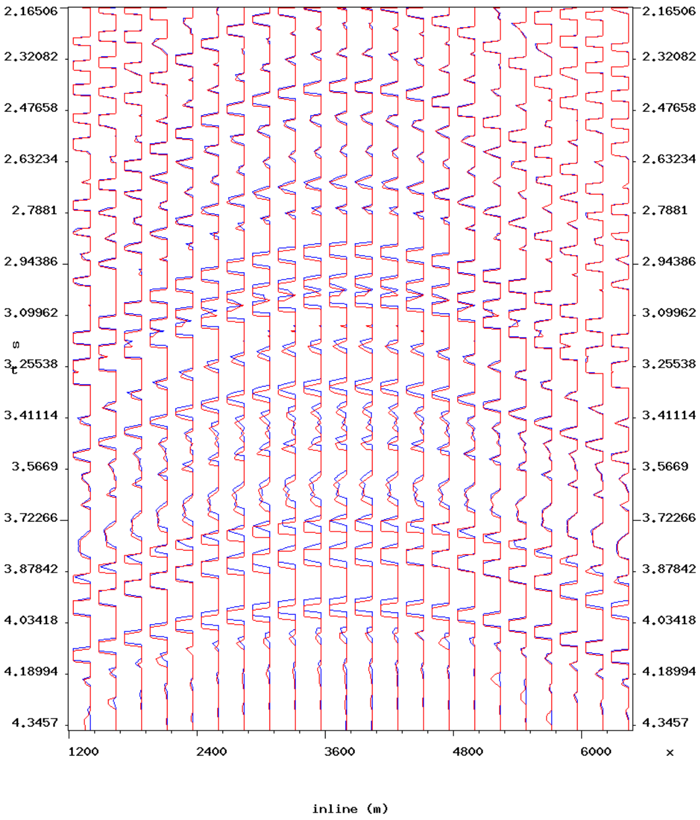


Figure 3: Time shifts observed in common-midpoint gathers centered above the target reservoir (blue is baseline, red is monitor). Travel times of the monitor near-offset reflections traveling through the negative velocity anomaly of Figure 2 are slightly delayed. [**CR**] musa2/. timeshifts

For time-lapse problems we consider slowness perturbations $\delta s(\mathbf{x})$ with support wholly contained in the interior of D. If $G(\mathbf{x}, \mathbf{y})$ is Green's function for the unperturbed Helmholtz equation (3) in D and absorbing boundary conditions, then equation (1) is equivalent to the Lippmann-Schwinger integral equation

$$u_S(\mathbf{x}, \mathbf{y}) \; = -\,\omega^2 \int_D G(\mathbf{x}, \mathbf{y}) \delta s(\mathbf{y}) \left[u_I(\mathbf{y}) + u_S(\mathbf{y})\right] d\mathbf{y}, \tag{4}$$

or, equivalently,

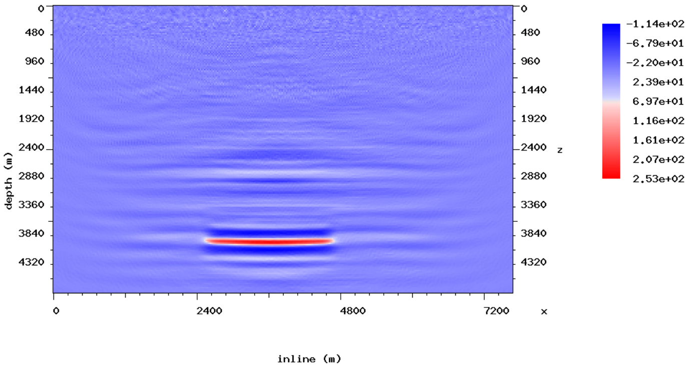
$$u_S(\mathbf{x},\mathbf{y}) \; = -\,\omega^2 \int_{\operatorname{supp}\delta s} G(\mathbf{x},\mathbf{y})\delta s(\mathbf{y})\left[u_I(\mathbf{y}) + u_S(\mathbf{y})\right] d\mathbf{y}. \tag{5}$$

The incident wavefield $u_I(\mathbf{x})$ in (4),(5) is assumed known.

Our method does not rely on solving (5). However, we will briefly discuss potential advantages of solving Lippmann-Schwinger instead of the Helmholtz equation. Three advantages of solving (5) for time-lapse problems instead of solving (1) with absorbing boundary conditions are immediately evident. First, it suffices to solve the Lippmann-Schwinger equation in the domain $\operatorname{supp}\delta s$ that in practical applications is much smaller that D (e.g., compaction effects are limited to overburden above producing reservoirs). After discretization, (5) becomes a system of linear equations with a dense modeling operator, and dimension of the model space is determined by the size of $\operatorname{supp}\delta s$ over the computational grid. Second, solution to (5) automatically satisfies absorbing boundary conditions along ∂D because the unperturbed Green's function $G(\mathbf{x},\mathbf{y})$ already satisfies those conditions. Third, once the scattered field is computed inside the support of δs, equation (5) can be used to compute its values outside the perturbation—e.g., at surface receivers.

However, discretization of (5)[2] is a dense linear system, and its numerical properties are highly dependent on the spectral content (smoothness) of δs (Duan and Rokhlin, 2009). "Sparsifying" preconditioners for (5) are an area of active research (see Ying (2015) for homogeneous backgrounds) and merit an investigation as a potentially useful technique for forward modeling of scattered wavefields for spatially bounded perturbations. Another challenge of using (5) is that it explicitly contains Green's function for problem (3). However, spatial boundedness of one of the arguments allows practical application of precomputed Green's functions (Etgen, 2012). In (5) both source and receiver arguments belong to the support of perturbation δs, making use of precomputed Green's functions feasible for compact targets. Computation of the scattered wavefield $u_S(\mathbf{x})$ outside of $\operatorname{supp}\delta s$ can be computationally equally efficient as the wavefield is required only at surface receiver locations.

Figure 4: The parallel difference method (Maharramov and Biondi, 2014c) fails to resolve the long-wavelength velocity changes of Figure 2 changes in the overburden, and produces negative short-wavelength artifacts around the target reflector. [**CR**] musa2/. pardiff



Assuming that $\delta s = O(\epsilon)$ where ϵ is a characteristic magnitude of model perturbation, and formally representing the scattered wavefield as a series

$$u_S(\mathbf{x}) \;=\; u_S^{(1)}(\mathbf{x}) \;+\; u_S^{(2)}(\mathbf{x}) \;+\; \ldots, \tag{6}$$

[2] using quadratures similar to Duan and Rokhlin (2009) for handling singularities at $\mathbf{x} = \mathbf{y}$

where

$$u_S^{(i)}(\mathbf{x}) \;=\; O(\epsilon^i), \tag{7}$$

we obtain

$$\begin{aligned} u_S^{(1)}(\mathbf{x}) \;&=\; -\,\omega^2 \int_{\mathrm{supp}\,\delta s} G(\mathbf{x},\mathbf{y})\delta s(\mathbf{y}) u_I(\mathbf{y}) d\mathbf{y}, \\ u_S^{(i+1)}(\mathbf{x}) \;&=\; -\,\omega^2 \int_{\mathrm{supp}\,\delta s} G(\mathbf{x},\mathbf{y})\delta s(\mathbf{y}) u_S^{(i)}(\mathbf{y}) d\mathbf{y},\; i=0,1,\dots \end{aligned} \tag{8}$$

From (8) we immediately see limitations of the Born series (6) in relating the diffracted wavefield $u_S(\mathbf{x})$ to $\delta s(\mathbf{x})$ for long-wavelength small-magnitude model perturbations. Indeed, assuming without a loss of generality, homogeneous background $s(\mathbf{x}) = s_0$ and constant and finite δs, we have

$$G(\mathbf{x},\mathbf{y}) \;=\; \frac{\exp(i\omega s_0|\mathbf{x}-\mathbf{y}|)}{4\pi|\mathbf{x}-\mathbf{y}|}. \tag{9}$$

Incident plane wave propagating along axis x^1 is given by

$$u_I(\mathbf{x}) \;=\; \exp(i\omega s_0 x^1). \tag{10}$$

For a sufficiently small $\mathrm{diam}\,(\mathrm{supp}\,\delta s) \;\ll\; x^1$ the denominator of (9) is asymptotically a constant factor if $\mathbf{y} \in \mathrm{supp}\,\delta s$, and from (8) we obtain

$$\begin{aligned} u_S^{(1)}(x^1,0,0) \;&\approx\; -\,\frac{\delta s\cdot\omega^2}{4\pi x^1}\int_{\mathrm{supp}\delta s\cap\mathbb{R}^1} \exp(i\omega s_0(x^1-y^1))\exp(i\omega s_0 y^1)dy^1 \\ &=\; -\,\frac{L\cdot\delta s\cdot\omega^2}{4\pi x^1}\exp(i\omega s_0 x^1), \end{aligned} \tag{11}$$

where

$$L \;=\; |\mathrm{supp}\delta s \cap \mathbb{R}^1(x^1)|, \tag{12}$$

is the length of model perturbation along axis $\mathbb{R}^1(x^1)$. Equation (11) means that the first-order Born scattering under our assumptions only affects the amplitude but not the phase of the scattered wavefield. Indeed, phase changes accumulate in (6) through the effect of the denominator in (9), requiring exponentially many terms to account for a phase delay or advance in the scattered (transmitted) wavefield. However, transmission through a constant perturbation δs of length L would cause a phase change proportional to $L\delta s$, therefore any technique based on truncated Born scattering would be suboptimal for relating large-wavelength, or “blocky”, velocity perturbations to measured scattered wavefields. This is a well-known limitation of the diffraction tomography (Wu and Toksoz, 1987) that is inherited by full-waveform inversion using L_2 misfit (Fichtner, 2011). On the other hand, Born series is a very good scattering approximation for small-wavelength, large-amplitude perturbations as, again, demonstrated by equation (11) (compare with Slaney et al. (1984)).

Rytov scattering series (Ishimaru, 1999) based on asymptotic phase expansion of the scattered wavefield linearly relates phase changes and magnitudes of the slowness change in first approximation, as does the initial approximation of full-waveform inversion of phase differences (Fichtner, 2011). Moreover, in time-lapse problems of inverting long-wavelength small-magnitude model perturbations, Rytov inverse scattering (and phase-only FWI) are less sensitive to errors in the background model.

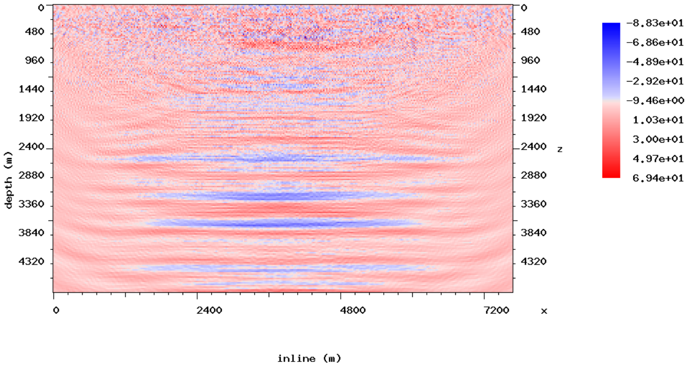


Figure 5: The cross-updated FWI method (Maharramov and Biondi, 2014c) cross-equalizes the baseline and monitor model but still fails to resolve the long-wavelength overburden changes of Figure 2. [**CR**] musa2/. xdiff

Indeed, assuming for simplicity, but without a loss of generality, a constant background s_0 and constant finite perturbation δs, the phase change for transmitted a plane wave traveling through a perturbation δs of characteristic dimension L is approximately proportional to $L\delta s/s_0$. For significant phase changes phase wraps around 2π, and this happens when the phase delay is a multiple of the incident wavelength. Fitting peaks and trough of the modeled and observed scattered wavefields (ignoring the amplitude information) then results in ambiguity of the total phase change: phase change can be resolved only within an integer multiple of incident wavelengths. This results in a well known phenomenon of cycle skipping in FWI: unless the FWI starting slowness model is known within a full wavelength of the incident wave, the model cannot be resolved from signal phase information alone.

However, for time-lapse problems phase change due to a compact velocity anomaly is only a fraction of the wavelength. Indeed, translating to the time domain, time shifts due to dilation in overburden peak at about 10 ms (Rickett et al., 2006; Maharramov and Biondi, 2015b; Maharramov et al., 2015a)—i.e., about a third of the period for a 30 Hz signal (see Figure 3). Therefore, phase changes (equivalently, time delays) of scattered wavefields for small-magnitude long-wavelength perturbations that are of interest for us can still be translated into slowness changes, albeit errors in the background will result in errors in the estimated slowness perturbation:

$$\delta s_W \; = \; \frac{s_W}{s_0}\delta s, \tag{13}$$

where s_W is the inaccurate background slowness and δs_W is the corresponding estimated slowness perturbation. Equation (13) means that even with a wrong background a qualitative perturbation magnitude information can still be extracted from the scattered wavefield. Note that location of the perturbation is determined by the illumination pattern of incident wavefields. Poor target illumination results in the ambiguity of anomaly characteristic dimension L versus the perturbation magnitude δs as the two enter in (13) in a product. For example, lack of reflectors above the velocity anomaly results in an ambiguous vertical extent of the anomaly.

To implement a practical time-lapse inversion method using phase-only FWI, we can invert two models (unperturbed baseline and perturbed monitor) simultaneously, imposing a model-difference regularization. The latter is required to create a common "background" model for both inversions making the application of (13) possible. Note that baseline and monitor inversions may still cycle-skip, but the purpose of imposing a model-difference

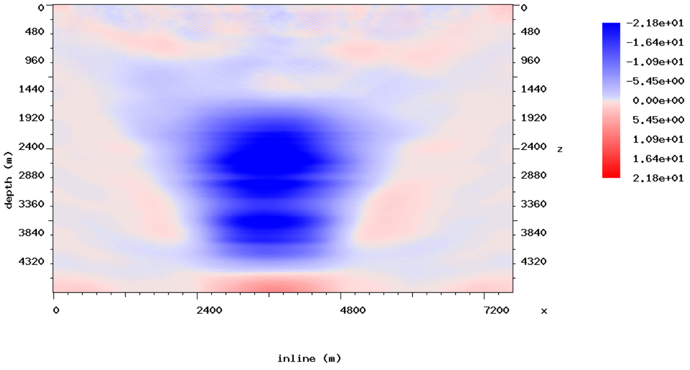


Figure 6: Simultaneous FWI with a total-variation model-difference regularization (Maharramov and Biondi, 2014e) resolves the long-wavelength overburden changes of Figure 2, but underestimates the maximum change, depending on the regularization strength. [**CR**] musa2/. tvinverted

regularization is to ensure that they are either equally accurate or equally inaccurate for wavelengths greater than the characteristic wavelength of the inverted perturbation.

For blocky, long-wavelength anomalies we impose blockiness-promoting total variation regularization (Maharramov and Biondi, 2014e), while for the recovery of short-wavelength features we use L_2 Tikhonov model-difference regularization (Maharramov and Biondi, 2013).

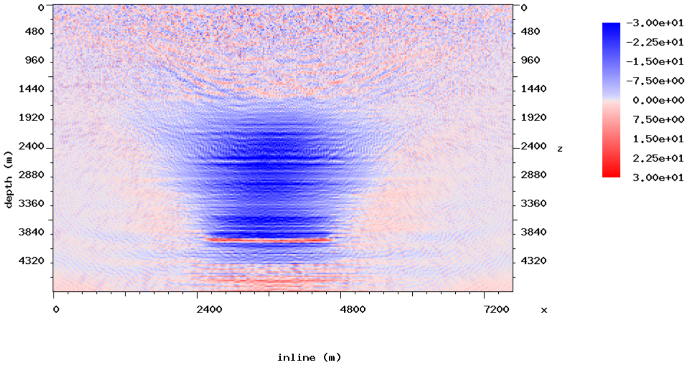


Figure 7: Simultaneous FWI using Tikhonov model-difference regularization, with the long-wavelength inversion of Figure 6 supplied as a prior. Note that such multiscale approach can now resolve the short-wavelength positive-velocity changes of Figure 2. Strong Tikhonov regularization results in underestimated velocity changes within the reservoir but correctly locates the anomalies. [**CR**] musa2/. 30multi

METHOD

Full-waveform inversion is defined as solving the following optimization problem (Tarantola, 1984; Virieux and Operto, 2009)

$$\|\mathbf{Mu} - \mathbf{d}\|_2 \ \rightarrow \ \min, \tag{14}$$

where $\mathbf{M}, \mathbf{d}$ are the measurement operator and data, $\mathbf{u}$ is the solution of a forward-modeling problem

$$\mathbf{D}(\mathbf{m})\mathbf{u} \ = \ \phi, \tag{15}$$

where $\mathbf{D}$ is the forward-modeling operator that depends on a model vector $\mathbf{m}$ as a parameter, and ϕ is a source. The minimization problem (14) is solved with respect to either both the model $\mathbf{m}$ and source ϕ or just the model. In the frequency-domain formulation of the acoustic waveform inversion, the forward-modeling equation (15) becomes

$$-\omega^2 u - v^2(x^1, \ldots, x^n)\Delta u \;=\; \phi(\omega, x^1, \ldots, x^n), \tag{16}$$

where ω is a temporal frequency, n is the problem dimension, and v is the acoustic wave propagation velocity. Values of the slowness $s = 1/v$ at all the points of the modeling domain constitute the model parameter vector $\mathbf{m}$. The direct problem (16) can be solved in the frequency domain (Virieux and Operto, 2009). The inverse problem (14) is typically solved using a multiscale approach, from low to high frequencies, supplying the output of each frequency inversion to the next step (Fichtner, 2011).

FWI applications in time-lapse problems seek to recover induced changes in the subsurface model using multiple datasets from different acquisition vintages. For two surveys sufficiently separated in time, we call such datasets (and the associated models) *baseline* and *monitor*.

Time-lapse FWI can be carried out by separately inverting the baseline and monitor models (*parallel difference*), or by inverting them sequentially with, e.g., the baseline supplied as a starting model for the monitor inversion (*sequential difference*). Another alternative is to apply the *double-difference* method, with a baseline model inversion followed by a monitor inversion that solves the following optimization problem,

$$\| \left(\mathbf{M}_m^s \mathbf{u}_m - \mathbf{M}_b^s \mathbf{u}_b\right) - \left(\mathbf{M}_m \mathbf{d}_m - \mathbf{M}_b \mathbf{d}_b\right) \|_2 \;\rightarrow\; \min, \tag{17}$$

by changing the monitor model (Watanabe et al., 2004; Denli and Huang, 2009; Zheng et al., 2011; Asnaashari et al., 2012; Raknes et al., 2013). The subscripts in equation (17) denote the baseline and monitor surveys, $\mathbf{d}$ denotes the observed data, and the $\mathbf{M}$'s are measurement operators that project the synthetic and field data onto a common grid. The superscript s indicates the measurement operators applied to the modeled data. For phase-only inversion, in all of the subsequent equations, the modeled and observed data differences should be replaced with the corresponding "phase differences"

$$\mathbf{u} \;-\; \mathbf{d} \;\Longrightarrow\; \sin\arg\mathbf{u} \;-\; \sin\arg\mathbf{d}, \tag{18}$$

where arg is the complex argument function of frequency domain wavefields. Note that unlike the traditional phase-only inversion (Fichtner, 2011), we evaluate sine of the phase to avoid phase discontinuities.

In all of these techniques, optimization is carried out with respect to one model at a time, albeit of different vintages at different stages of the inversion. In our method we invert for the baseline and monitor models *simultaneously* by solving either one of the following two optimization problems:

$$\begin{aligned}
\alpha\|\mathbf{M}_b\mathbf{u}_b - \mathbf{d}_b\|_2^2 + \beta\|\mathbf{M}_m\mathbf{u}_m - \mathbf{d}_m\|_2^2 + & \qquad (19)\\
\gamma\| \left(\mathbf{M}_m^s \mathbf{u}_m - \mathbf{M}_b^s \mathbf{u}_b\right) - \left(\mathbf{M}_m \mathbf{d}_m - \mathbf{M}_b \mathbf{d}_b\right) \|_2^2 + & \qquad (20)\\
\alpha_1\|\mathbf{W}_b\mathbf{R}_b(\mathbf{m}_b - \mathbf{m}_b^{\mathrm{PRIOR}})\|_2^2 + & \qquad (21)\\
\beta_1\|\mathbf{W}_m\mathbf{R}_m(\mathbf{m}_m - \mathbf{m}_m^{\mathrm{PRIOR}})\|_2^2 + & \qquad (22)\\
\delta\|\mathbf{W}\mathbf{R}(\mathbf{m}_m - \mathbf{m}_b - \Delta\mathbf{m}^{\mathrm{PRIOR}})\|_2^2 \;\rightarrow\; \min, & \qquad (23)
\end{aligned}$$

or

$$\alpha \|\mathbf{M}_b \mathbf{u}_b - \mathbf{d}_b\|_2^2 + \beta \|\mathbf{M}_m \mathbf{u}_m - \mathbf{d}_m\|_2^2 + \tag{24}$$

$$\gamma \| \left(\mathbf{M}_m^s \mathbf{u}_m - \mathbf{M}_b^s \mathbf{u}_b\right) - \left(\mathbf{M}_m \mathbf{d}_m - \mathbf{M}_b \mathbf{d}_b\right) \|_2^2 + \tag{25}$$

$$\alpha_1 \|\mathbf{W}_b \mathbf{R}_b (\mathbf{m}_b - \mathbf{m}_b^{\mathrm{PRIOR}})\|_1 + \tag{26}$$

$$\beta_1 \|\mathbf{W}_m \mathbf{R}_m (\mathbf{m}_m - \mathbf{m}_m^{\mathrm{PRIOR}})\|_1 + \tag{27}$$

$$\delta \|\mathbf{W}\mathbf{R}(\mathbf{m}_m - \mathbf{m}_b - \Delta\mathbf{m}^{\mathrm{PRIOR}})\|_1 \ \rightarrow \ \min, \tag{28}$$

with respect to both the baseline and monitor models $\mathbf{m}_b$ and $\mathbf{m}_m$. Problem (19-23) describes time-lapse FWI with L_2 regularization of the individual models (21,22) and model difference (23) (Maharramov and Biondi, 2014c). The second formulation (24-28) involves an L_1-regularization of the individual models and their difference (Maharramov and Biondi, 2014e; Maharramov et al., 2015b). The terms (24) correspond to separate baseline and monitor inversions, the term (25) is the optional double difference term, the terms (26) and (27) are optional separate baseline and monitor inversion regularization terms (Aster et al., 2012), and the term (28) represents regularization of the model difference. In (26)-(28), $\mathbf{R}$ and $\mathbf{W}$ denote regularization and weighting operators respectively, with the subscript denoting the survey vintage where applicable. If $\mathbf{R}$ is the *gradient magnitude* operator

$$\mathbf{R}f(x,y,z) \ = \ \sqrt{f_x^2 + f_y^2 + f_z^2}, \tag{29}$$

then (26-28) become *total-variation (TV) seminorms*. The latter case is of particular interest in this work as the minimization of the L_1 norm of gradient may promote "blockiness" of the model-difference, potentially reducing oscillatory artifacts (Rudin et al., 1992; Aster et al., 2012).

A joint inversion approach has been applied earlier to the linearized waveform inversion (Ayeni and Biondi, 2012). In Maharramov and Biondi (2013, 2014c,a), a simultaneous full-waveform inversion problem (19,23) was studied with a single model difference L_2 regularization term (23).

An implementation of the proposed simultaneous inversion algorithm requires solving a nonlinear optimization problem with twice the data and model dimensions of problems (14) and (17). The model difference regularization weights $\mathbf{W}$ and, optionally, the prior $\Delta\mathbf{m}^{\mathrm{PRIOR}}$ may be obtained from prior geomechanical information. For example, a rough estimate of production-induced velocity changes can be obtained from time shifts (Hatchell and Bourne, 2005; Barkved and Kristiansen, 2005) and used to map subsurface regions of expected production-induced perturbation, and optionally provide a difference prior. However, successfully solving the L_1-regularized problem (24-25) is less sensitive to choice of the weighting operator $\mathbf{W}$. For example, we show below that the TV-regularization using (29) with $\mathbf{W} = 1$ recovers non-oscillatory components of the model difference, while the L_2 approach would result in either smoothing or uniform reduction of the model difference.

In addition to the fully simultaneous inversion, Maharramov and Biondi (2013, 2014c) proposed and tested a *cross-updating* technique that offers a simple but remarkably effective approximation to minimizing the objective function (19),(23), while obviating the difference regularization and weighting operators $\mathbf{R}$ and $\mathbf{W}$ for problem (19,23). This technique consists of one standard run of the sequential difference algorithm, followed by a second run with the inverted monitor model supplied as the starting model for the second baseline inversion

$$\begin{aligned}\mathbf{m}_{\mathrm{INIT}} \rightarrow &\text{baseline inversion} \rightarrow \text{monitor inversion} \rightarrow \\ &\text{baseline inversion} \rightarrow \text{monitor inversion},\end{aligned} \tag{30}$$

and computing the difference of the latest inverted monitor and baseline models. Process (30) can be considered as an approximation to minimizing (19) and (23) because non-repeatable footprints of both inversions are propagated to both models, canceling out in the difference. Both the simultaneous inversion and cross-updating minimize the model difference by tackling model artifacts that are in the null space of the Fréchet derivative of the forward modeling operatorss. The joint inversion minimizes the effect of such artifacts on the model difference by either minimizing the model difference term (23) in the simultaneous inversion, or by propagating these artifacts to both models in cross-updating (30). Note that this process is not guaranteed to improve the results of the baseline and monitor model inversions but was only proposed for improving the model difference. Maharramov and Biondi (2014c,a) demonstrated a significant improvement of model difference recovery by both the L_2-regularized target-oriented simultaneous inversion and cross-updating compared to the parallel, sequential and double difference techniques. The simultaneous inversion and cross-updating yielded qualitatively similar results within the inversion target. Maharramov et al. (2015c) studied the regularized double-difference inversion (25,28).

NUMERICAL EXPERIMENTS AND DISCUSSION

Our previous work (Maharramov and Biondi, 2014e,b) has demonstrated effective recovery of blocky velocity anomalies from long-offset acquisitions in the presence of noise and repeatability issues. In this work we demonstrate the recovery of blocky anomalies in the more challenging case of phase-only inversion of narrow-offset reflection data. Conceptually our synthetic example is similar to the earlier field data case study of Maharramov and Biondi (2015b); Maharramov et al. (2015a).

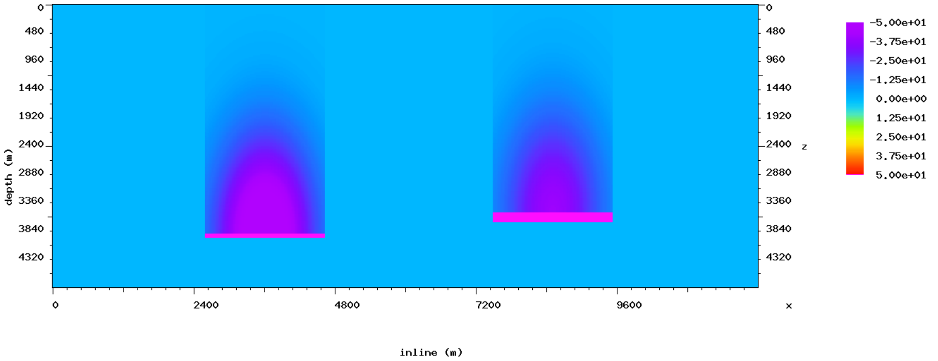


Figure 8: True model difference for demonstrating the inversion of multiple overburden anomalies. [**ER**] musa2/. twoanom

As a baseline model we use the flat reflector model of Figure 1. The target reflector (reservoir) is located at a depth of 3900 m, the monitor (perturbed) model has two velocity anomalies–a positive +300 m/s change due to compaction and fluid substitution within the reservoir, and a blocky negative velocity change in the overburden above the reservoir, peaking at −50 m/s (see Figure 2). No physical reflector movement is prescribed.

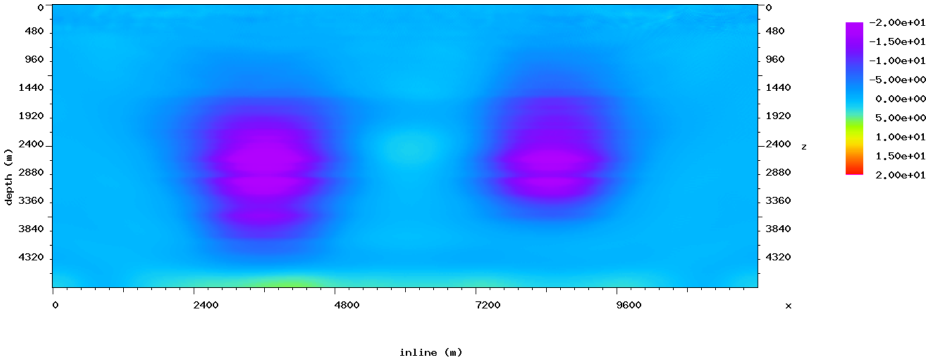


Figure 9: Inversion of the two long-wavelength overburden anomalies of Figure 8 using simultaneous time-lapse FWI with total-variation model-difference regularization. [**CR**] musa2/. twoanominv

For generating synthetic data we used a towed streamer acquisition geometry with the maximum offset of 5 km. The results of parallel difference and cross-updating are shown in Figures 4 and 5. Note that neither result succeeds in recovering the blocky anomaly. The FWI starting model used in these experiments was a smoothed true model, using a 720 m smoothing window.

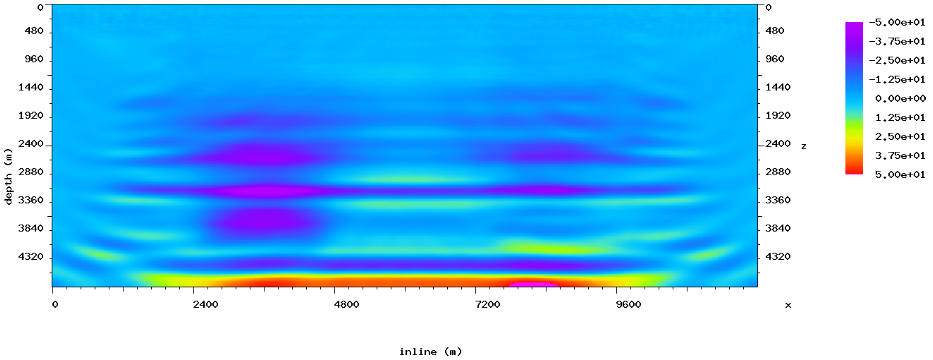


Figure 10: Inversion of the two long-wavelength overburden anomalies of Figure 8 starting from a bad initial model and using weak regularization (a small regularization parameter). FWI cycle skipped, and the baseline and monitor inversion diverged, contaminating the difference with cycle-skipping artifacts. [**CR**] musa2/. twoanombadweak

The result of simultaneous inversion with a total-variation model-difference regularization is shown in Figure 6. The result is qualitatively accurate although peak magnitudes are underestimated due to regularization. To assess the effectiveness of our inversion, in Figures 12(a) and 13(b) we show monitor images migrated using the true monitor and true baseline models, respectively. Note that the overestimated velocities in the overburden result in a downward reflector shift in Figure 13(b). However, migrating the monitor data using the sum of the baseline model and the *inverted* blocky anomaly of Figure 6 results in the image of Figure 13(a): the downward shift of reflectors in the overburden is now significantly reduced.

To recover the short-wavelength changes within the reservoir, we supplied the result of

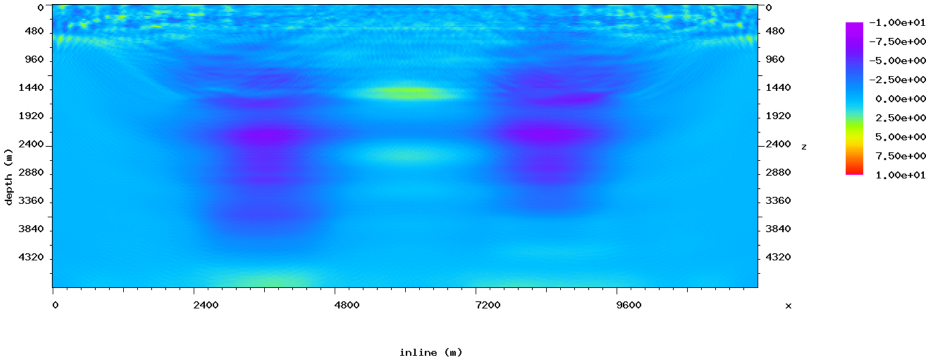


Figure 11: Inversion of the two long-wavelength overburden anomalies of Figure 8 starting from a bad initial model but using stronger regularization (a larger regularization parameter). FWI still cycle skipped, however, the strong model-difference regularization kept baseline and monitor within the characteristic wavelength of the overburden anomalies. The resulting model difference inversion is qualitatively accurate (compare with Figure 9), albeit stronger regularization has resulted in underestimated velocity magnitudes. [**CR**] musa2/. twoanombadstrong

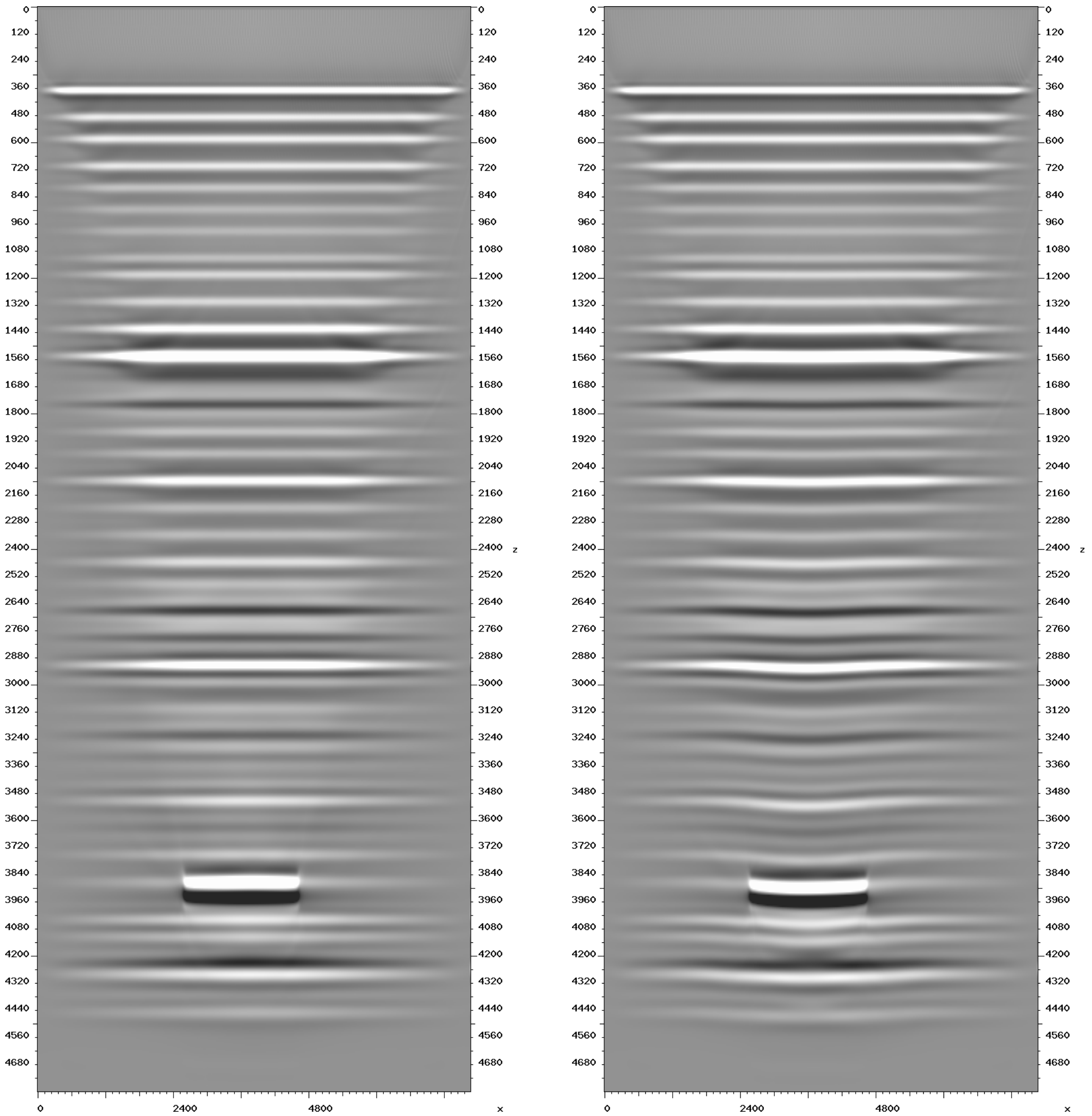
Figure 6 as a model-difference prior to inversion (19,23). Note that the resulting model features both long and short-wavelength velocity perturbations. The reservoir perturbation is underestimated due to strong regularization. Maharramov and Biondi (2015a) discuss a regularization scheme for multi-scale inversion that honors true model magnitudes.

And finally, Figures 8 and 9 demonstrate recovery of two separate overburden anomalies. In both cases FWI start from a smoothed true velocity. The result of starting FWI with a wrong velocity resulting in cycle-skipping is shown in Figure 10. We deliberately used a weak regularization parameter at model-difference regularization to demonstrate the effect of diverging baseline and monitor models on the inverted model difference. Figure 11 contains the result of using a stronger TV regularization. As described in the Theory section above, we ensure that the two models cycle-skip “in synchrony” and are still able to qualitatively recover the anomalies, although with strongly underestimated velocities—compare with equation (13).

In this work we provided a theoretical justification for the time-lapse inversion methods of Maharramov and Biondi (2013, 2014e, 2015b) and demonstrated a stable recovery of both short and long-wavelength velocity anomalies form narrow-offset reflection seismic data. We envisage wide-spread application of the simultaneous FWI with model-difference regularization and hierarchical multi-scale inversion in applications ranging from applied geophysics to electromagnetic and optical scattering.

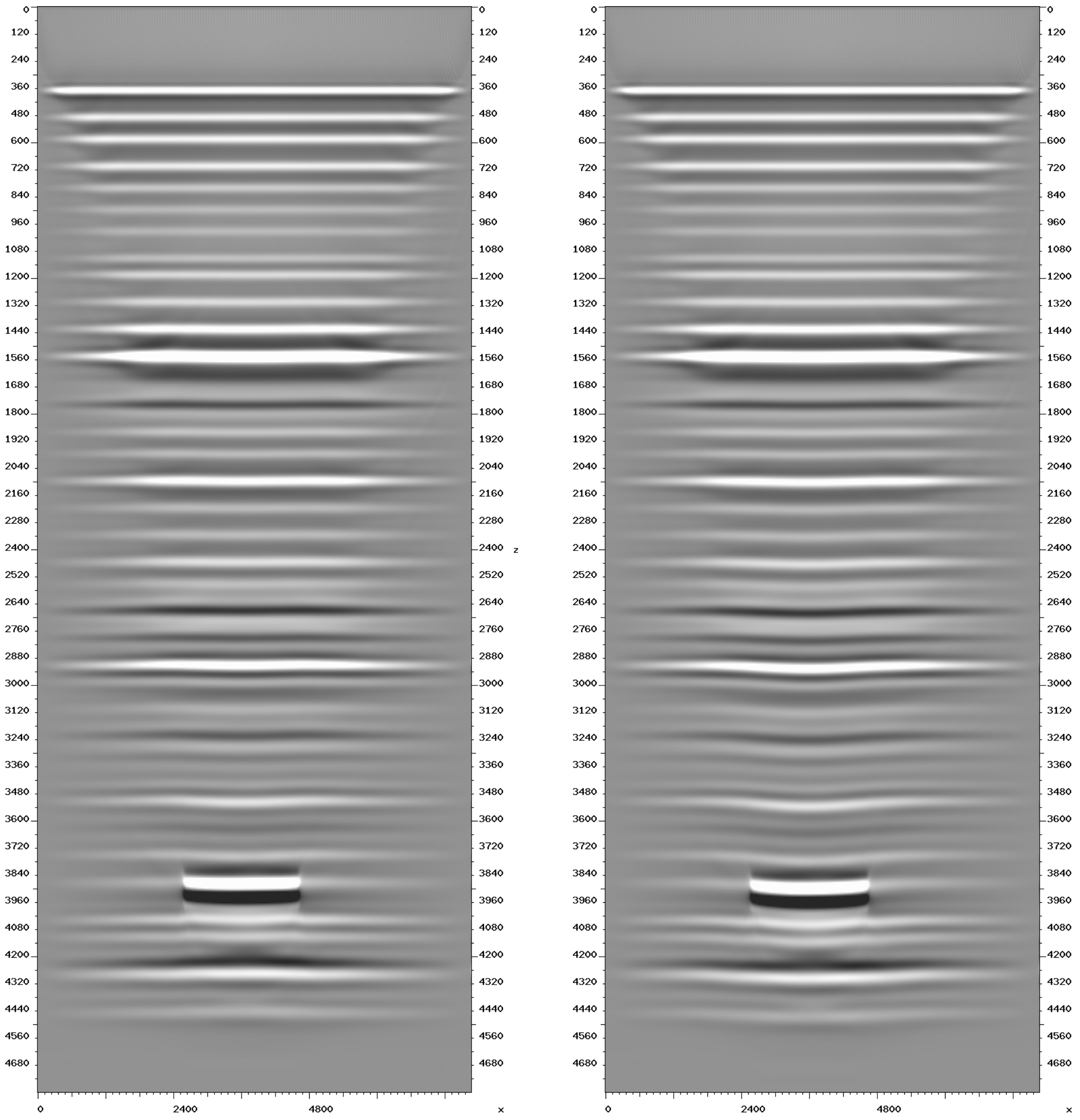
ACKNOWLEDGEMENTS

The authors would like to thank Stewart A. Levin and Mark A. Meadows for a number of useful discussions, and the affiliate members of Stanford Exploration Project for their support.



(a) (b)

Figure 12: (a) True monitor image. (b) Monitor migrated using the baseline velocity model. Note that overestimated velocity in the overburden results in a downward reflector shift in the right image. [**CR**] musa2/. montrueimg,monwithbase



(a) (b)

Figure 13: (a) Monitor image migrated using the sum of the baseline model and inverted model difference of Figure 6. (b) Monitor migrated using the baseline velocity model. Note that reflector shift in the overburden has been significantly reduced in the left image. [**CR**] musa2/. improved,monwithbase

REFERENCES

Asnaashari, A., R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux, 2012, Time-lapse imaging using regularized FWI: A robustness study: 82nd Annual International Meeting, SEG, Expanded Abstracts, doi:10.1190/segam2012-0699.1, 1–5.

Aster, R., B. Borders, and C. Thurber, 2012, Parameter estimation and inverse problems: Elsevier.

Ayeni, G. and B. Biondi, 2012, Time-lapse seismic imaging by linearized joint inversion—A Valhall Field case study: 82nd Annual International Meeting, SEG, Expanded Abstracts, doi:10.1190/segam2012-0903.1, 1–6.

Barkved, O. and T. Kristiansen, 2005, Seismic time-lapse effects and stress changes: Examples from a compacting reservoir: The Leading Edge, **24**, no. 12, 1244–1248.

Biondi, B., C. Deutsch, R. Gundes, D. Lumley, G. Mavko, T. Mukerji, J. Rickett, and M. Thiele, 1996, Reservoir monitoring: A multidisciplinary feasibility study: 66th Annual International Meeting, SEG, Expanded Abstracts, 1775–1778.

Colton, D. and R. Kress, 1998, Inverse acoustic and electromagnetic scattering theory: Springer.

Denli, H. and L. Huang, 2009, Double-difference elastic waveform tomography in the time domain: 79th Annual International Meeting, SEG, Expanded Abstracts, 2302–2306.

Duan, R. and V. Rokhlin, 2009, High-order quadratures for the solution of scattering problems in two dimensions: J. Comput. Phys., **228**, 2152–2174.

Engquist, B. and A. Majda, 1977, Absorbing boundary conditions for the numerical simulation of waves: Math. Comp., **31**, 629–651.

Etgen, J. T., 2012, 3d wave equation kirchhoff migration: SEG Technical Program Expanded Abstracts 2012, 1–5.

Fichtner, A., 2011, Full seismic modeling and inversion: Springer.

Hatchell, P. and S. Bourne, 2005, Measuring reservoir compaction using time-lapse timeshifts: 75th Annual International Meeting, SEG, Expanded Abstracts, 2500–2503.

Ishimaru, A., 1999, Wave propagation and scattering in random media: Academic Press.

Johnston, D., 2013, Practical applications of time-lapse seismic data: Society of Exploration Geophysicists.

Maharramov, M. and B. Biondi, 2013, Simultaneous time-lapse full waveform inversion: SEP Report, **150**, 63–70.

———, 2014a, Joint 4DFWI with model-difference regularization. SEG-AGU summer research workshop: SEG-AGU Summer Research Workshop. Advances in Active+Passive "Full Wavefield" Seismic Imaging: From Reservoirs to Plate Tectonics.

———, 2014b, Joint full-waveform inversion of time-lapse seismic data sets: 84th Annual Meeting, SEG, Expanded Abstracts, 954–959.

———, 2014c, Joint full-waveform inversion of time-lapse seismic data sets: SEP Report, **152**, 19–28.

———, 2014d, Multi-model full-waveform inversion: SEP Report, **155**, 187–192.

———, 2014e, Robust joint full-waveform inversion of time-lapse seismic datasets with total-variation regularization: SEP Report, **155**, 199–208.

———, 2015a, Multi-scale inversion of subsurface velocity models using cartoon-texture decomposition: SIAM Conference on Mathematical and Computational Issues in the Geosciences, MS2 Full Waveform Inversion I: Algorithms and Performance.

———, 2015b, Resolving the effects of production-induced overburden dilation using simultaneous TV-regularized time-lapse fwi: SEP Report, **158**, 1–10.

Maharramov, M., B. Biondi, and M. Meadows, 2015a, Simultaneous tv-regularized time-lapse fwi with application to field data: SEG Technical Program Expanded Abstracts 2015, 1236–1241.

Maharramov, M., B. Biondi, and S. Ronen, 2015b, Robust simultaneous time-lapse full-waveform inversion with total-variation regularization of model difference: 77th EAGE Conference and Exhibition, Extended Abstract, We P3 09.

Maharramov, M., Y. Ma, and B. Biondi, 2015c, Double-difference time-lapse FWI with a total-variation regularization: SEP Report, **158**, 263–270.

Raknes, E., W. Weibull, and B. Arntsen, 2013, Time-lapse full waveform inversion: Synthetic and real data examples: 83rd Annual International Meeting, SEG, Expanded Abstracts, 944–948.

Rickett, J., L. Duranti, T. Hudson, and N. Hodgson, 2006, Compaction and 4D time strain at the Genesis Field: 76th Annual International Meeting, SEG, Expanded Abstracts, 3215–3219.

Rickett, J., L. Duranti, T. Hudson, B. Regel, and N. Hodgson, 2007, 4D time strain and the seismic signature of geomechanical compaction at Genesis: The Leading Edge, **26**, 644–647.

Routh, P., G. Palacharla, I. Chikichev, and S. Lazaratos, 2012, Full wavefield inversion of time-lapse data for improved imaging and reservoir characterization: 82nd Annual International Meeting, SEG, Expanded Abstracts, doi:10.1190/segam2012-1043.1, 1–6.

Rudin, L. I., S. Osher, and E. Fatemi, 1992, Nonlinear total variation based noise removal algorithms: Physica D: Nonlinear Phenomena, **60**, 259–268.

Sirgue, L., O. Barkved, J. Dellinger, J. Etgen, U. Albertin, and J. Kommendal, 2010, Full waveform inversion: The next leap forward in imaging at Valhall: First Break, **28**, no. 4, 65–70.

Slaney, M., A. Kak, and L. Larsen, 1984, Limitations of imaging with first-order diffraction tomography: Microwave Theory and Techniques, IEEE Transactions on, **32**, 860–874.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, no. 8, 1259–1266.

Virieux, J. and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74, no.6**, no. 6, WCC1–WCC26.

Watanabe, T., S. Shimizu, E. Asakawa, and T. Matsuoka, 2004, Differential waveform tomography for time-lapse crosswell seismic data with application to gas hydrate production monitoring: 74th Annual International Meeting, SEG, Expanded Abstracts, 2323–2326.

Wu, R. and M. N. Toksoz, 1987, Diffraction tomography and multisource holography applied to seismic imaging: GEOPHYSICS, **52**, 11–25.

Yang, D., A. E. Malcolm, and M. C. Fehler, 2014, Time-lapse full waveform inversion and uncertainty analysis with different survey geometries: 76th EAGE Conference and Exhibition, Extended Abstract, We ELI1 10.

Ying, L., 2015, Sparsifying preconditioner for the lippmann–schwinger equation: Multiscale Modeling & Simulation, **13**, 644–660.

Zheng, Y., P. Barton, and S. Singh, 2011, Strategies for elastic full waveform inversion of timelapse ocean bottom cable (OBC) seismic data: 81st Annual International Meeting, SEG, Expanded Abstracts, 4195–4200.

Dissection of the full-waveform inversion Hessian

Biondo Biondi, Ettore Biondi, Musa Maharramov, and Yinbin Ma

ABSTRACT

We analyze the FWI Hessian and determine that is the combination of two well-understood operators in seismic imaging: 1) the Gauss-Newton Hessian and 2) a differential WEMVA operator. We illustrate this insight with several numerical examples of applying and inverting the different components of the Hessian operator to images originating from the waveform inversion of a simple synthetic dataset. We then discuss possible applications of the theory to making FWI more robust with respect to incorrect starting models, and to speeding-up the application of target-oriented fully non-linear seismic inversion.

INTRODUCTION

Improving the robustness and the convergence rate of full waveform inversion (FWI) is an important goal for any FWI algorithm. It becomes particularly important when we go beyond simple constant-density acoustic inversion and we aim at estimating several images representing different parameters, such as elastic properties, anisotropic parameters and attenuation parameters (Alves, 2015; Biondi and O'Reilly, 2015; Shen, 2015).

Newton-like methods that exploit the information contained in the Hessian matrix of the objective function (Pratt et al., 1998) are attractive and have been recently the subject of much research (Tang and Lee, 2010; Fichtner, 2011; Métivier et al., 2012; Baumstein, 2014; Deuzeman and Plessix, 2015). Pratt et al. (1998) presented an efficient frequency-domain method for evaluating the application of the Hessian to a model-perturbations vector. Epanomeritakis et al. (2008) presented an equivalent method for time-domain propagation. In this paper we start from the description of the time-domain method for applying the Hessian to a model-perturbations vector presented by Fichtner and Trampert (2011) and Fichtner (2011).

In the following section we analyze the full FWI Hessian and show that it is the composition of well-understood operators in seismic imaging; that is, the Gauss-Newton Hessian and the WEMVA operator. This new insight has the potential of leading to more robust and computationally efficient waveform-inversion algorithms. We discuss these potential applications in the discussion section of the paper, after we illustrate the properties of each component of the Hessian operator by showing its application to a simple FWI problem.

DISSECTION OF THE FULL FWI HESSIAN OPERATOR

In conventional amplitude-based full waveform inversion (FWI), we minimize the following objective function:

$$\phi(\mathbf{m}) = \frac{1}{2}\|\mathbf{d}_m(\mathbf{m}) - \mathbf{d}_o\|_2^2, \tag{1}$$

where $\mathbf{m}$ is the vector of model parameters, $\mathbf{d}_o$ is the vector of recorded data, and $\mathbf{d}_m(\mathbf{m})$ is the vector of modeled data. The modeled data are given by:

$$\mathbf{d}_m(\mathbf{m}) = \mathbf{R}\mathbf{u}(\mathbf{m}), \tag{2}$$

in which $\mathbf{R}$ is a linear operator that extracts the vector of modeled data, $\mathbf{d}_m$, from the vector of propagated wavefield, $\mathbf{u}$. In the following we assume that this wavefield is the solution of the following scalar wave equation:

$$\mathbf{L}\left(\mathbf{m}, \mathbf{u}\right) = \left[m\frac{\partial^2}{\partial t^2} - \nabla^2\right] u(\mathbf{x}, t, m) = s(\mathbf{x}, t), \tag{3}$$

where m represents the slowness squared, s is the source term, and t and $\mathbf{x}$ are the temporal and spatial coordinates, respectively. However, the insights on the full Hessian presented in this paper are valid and applicable also when the data are modeled using more complex wave equations than the one presented in equation 3.

The objective function 1 aims to minimize in a least-squares sense the data residuals, defined as:

$$\mathbf{\Delta d} = \mathbf{d}_m(\mathbf{m}) - \mathbf{d}_o. \tag{4}$$

The Hessian operator has an important role when solving the optimization problem of minimizing the objective function 1 by a Newton algorithm. If $\nabla\phi$ is the gradient vector, an improved search direction $\mathbf{\Delta m}$ can be obtained by solving the Newton system

$$\mathbf{H}\mathbf{\Delta m} = -\nabla\phi, \tag{5}$$

where $\mathbf{H}$ is the Hessian matrix. Because of the large scale of the FWI problem we seldom build the Hessian matrix explicitly, and never solve the Newton system directly. On the contrary, we invert the Hessian iteratively using an iterative method like conjugate-gradient. Furthermore, we do not iterate the inversion of $\mathbf{H}$ to full-convergence, but we stop after a small number of iterations. This approximate Newton algorithm is often referred as *truncated Newton method.*

To invert $\mathbf{H}$ iteratively we need only to be able to compute its application to an arbitrary model perturbations $\delta\mathbf{m}$. To perform this task we can precompute the Hessian matrix and perform several matrix-vector operations, or devise an algorithm that evaluates the application of $\mathbf{H}$ to a vector by performing wave propagations and correlations. The latter way is conceptually similar to the one employed to compute the FWI gradient using adjoint-state methods, and is the one that we will discuss in this section. In practice, hybrid schemes (i.e. methods that rely on precomputing some Hessian components and applying the adjoint-state methods for other components) can be actually the most computationally efficient solution.

If we start the iterative inversion of the Hessian matrix from a zero model-perturbations vector, the first search direction, $\mathbf{\Delta m}_1$, of a truncated Newton algorithm is equal to,

$$\mathbf{\Delta m}_1 = -\mathbf{H}^{'}\nabla\phi = -\mathbf{H}\nabla\phi, \tag{6}$$

where in equation 6 we use the Hermitian properties of the Hessian matrix.

In the numerical examples section we will show both the results of approximately solving the Newton system (equation 29), as well as computing the search direction $\mathbf{\Delta m}_1$ using equation 6.

The Hessian matrix is the second-order derivative of the objective function with respect to each model parameter; it can be expressed as follows:

$$\nabla_m\nabla_m\phi(\mathbf{m}) = (\nabla_u\nabla_u\phi\nabla_m\mathbf{u})\cdot\nabla_m\mathbf{u} + (\nabla_u\phi)\cdot\nabla_m\nabla_m\mathbf{u}, \tag{7}$$

where the first term on the right-hand side of the equation represents the Gauss-Newton Hessian.

As shown in Fichtner (2011), the application of the full FWI Hessian to a model perturbation, $\delta\mathbf{m}$, can be written as sum of three kernels applied to $\delta\mathbf{m}$ as:

$$\mathbf{H}(\mathbf{m}_0)\delta\mathbf{m} = [\mathbf{K}^a(\mathbf{m}_0) + \mathbf{K}^b(\mathbf{m}_0) + \mathbf{K}^c(\mathbf{m}_0)]\delta\mathbf{m}, \tag{8}$$

where $\mathbf{m}_0$ is the model around which we are linearizing the nonlinear objective function and the three kernels are:

$$\mathbf{K}^a(\mathbf{m}_0)\delta\mathbf{m} = \left[\textstyle\int_t \mathbf{u}_2^{'}\cdot\nabla_m\mathbf{L}\left(\mathbf{m}_0,\mathbf{u}_0\right)\ dt\right]\delta\mathbf{m}, \tag{9}$$

$$\mathbf{K}^b(\mathbf{m}_0)\delta\mathbf{m} = \left[\textstyle\int_t \mathbf{u}_1^{'}\cdot\nabla_m\nabla_u\mathbf{L}\left(\mathbf{m}_0,\mathbf{u}_0\right)\nabla_m\mathbf{u}\ dt\right]\delta\mathbf{m}, \tag{10}$$

$$\mathbf{K}^c(\mathbf{m}_0)\delta\mathbf{m} = \left[\textstyle\int_t \mathbf{u}_1^{'}\cdot\nabla_m\nabla_m\mathbf{L}\left(\mathbf{m}_0,\mathbf{u}_0\right)\ dt\right]\delta\mathbf{m}, \tag{11}$$

where $\int_t$ denotes time integration, $\mathbf{u}_0$ is the background wavefield that is the solution of:

$$\mathbf{L}(\mathbf{m}_0,\mathbf{u}_0) = s(\mathbf{x},t), \tag{12}$$

and $\mathbf{u}_1^{'}$ and $\mathbf{u}_2^{'}$ are the *primary adjoint* wavefield and *secondary adjoint* wavefield, respectively.

In equation 3 we parametrized the wave equation in terms of slowness squared; therefore, we can drop the third kernel of the full Hessian, $\mathbf{K}^c$, because $\nabla_m\nabla_m\mathbf{L}$ is zero. The kernel $\mathbf{K}^c$ would not be zero if more complex wave-equation were employed to model the data and the medium parameters were present non-linearly in the equation (e.g. Vp and Vs in a fully elastic wave equation). However, its evaluation would follow a flow similar the one used to evaluate $\mathbf{K}^b$.

The first step to interpret the two remaining kernels, $\mathbf{K}^a$ and $\mathbf{K}^b$, is the analysis of the two adjoint wavefields, $\mathbf{u}_1^{'}$ and $\mathbf{u}_2^{'}$. The primary adjoint wavefield, $\mathbf{u}_1^{'}$, is the solution of:

$$\nabla_u^{'}\mathbf{L}^{'}\mathbf{u}_1^{'} = -\nabla_u\phi. \tag{13}$$

Because $\mathbf{L}$ is linear with respect to the wavefield, and $\nabla_u\phi = \mathbf{\Delta d}$, equation 13 simplifies into

$$\mathbf{L}^{'}(\mathbf{m}_0,\mathbf{u}_1^{'}) = -\mathbf{\Delta d}. \tag{14}$$

The wavefield $\mathbf{u}_1^{'}$ is obtained by backpropagating the opposite of the data residual with the background model $\mathbf{m}_0$. It is the same wavefield that is computed to evaluate the gradient of objective function 1, and thus does not need to be recomputed when applying the Hessian operator.

The secondary adjoint wavefield, $\mathbf{u}_2^{'}$, is the solution of the following wave propagation problem:

$$\mathbf{L}^{'}(\mathbf{m}_0, \mathbf{u}_2^{'}) = -\left[\nabla_u \nabla_u \phi \nabla_m \mathbf{u} + \nabla_u \nabla_m \mathbf{L}^{'} \mathbf{u}_1\right]. \tag{15}$$

Since there are two source terms on the right hand side of equation 15, the secondary adjoint wavefield can be split into two components, $\mathbf{u}_{21}^{'}$ and $\mathbf{u}_{22}^{'}$, each solution of a backpropagation problem; that is:

$$\nabla_u \mathbf{L}^{'} \mathbf{u}_{21}^{'} = -\nabla_u \nabla_u \phi \nabla_m \mathbf{u}, \tag{16}$$

$$\nabla_u \mathbf{L}^{'} \mathbf{u}_{22}^{'} = -\nabla_u \nabla_m \mathbf{L}^{'} \mathbf{u}_1. \tag{17}$$

Consequently, the kernel $\mathbf{K}^a$ can be split in the sum of two kernels:

$$\mathbf{K}^{a1}(\mathbf{m}_0)\delta\mathbf{m} = \left[\int_t \mathbf{u}_{21}^{'} \cdot \nabla_m \mathbf{L}\left(\mathbf{m}_0, \mathbf{u}_0\right)\, dt\right] \delta\mathbf{m}, \tag{18}$$

$$\mathbf{K}^{a2}(\mathbf{m}_0)\delta\mathbf{m} = \left[\int_t \mathbf{u}_{22}^{'} \cdot \nabla_m \mathbf{L}\left(\mathbf{m}_0, \mathbf{u}_0\right)\, dt\right] \delta\mathbf{m}. \tag{19}$$

Next we interpret the first one, $\mathbf{K}^{a1}$. In equation 16, the derivative of the wavefield with respect of the model parameters can be written as the solution of the forward first-order Born scattering problem (Fichtner, 2011); that is:

$$\mathbf{L}(\mathbf{m}_0, \delta\mathbf{u}) = -\nabla_m \mathbf{L}(\mathbf{m}_0, \mathbf{u}_0). \tag{20}$$

Furthermore, in equation 16, $\nabla_u \nabla_u \phi$ is simply $\mathbf{R}^{'}\mathbf{R}$, which means extracting and reinjecting $\delta\mathbf{u}$ at the receiver positions.

Equation 16 can thus be rewritten as

$$\mathbf{L}^{'}(\mathbf{m}_0, \mathbf{u}_{21}^{'}) = \mathbf{R}^{'}\mathbf{R}\delta\mathbf{u}, \tag{21}$$

where $\delta\mathbf{u}$ is given by equation 20. After cross-correlation with the second time derivative of the forward propagated background wavefield ($\nabla_m \mathbf{L}(\mathbf{m}_0, \mathbf{u}_0)$), as indicated by equation 18, the application of $\mathbf{K}^{a1}$ to a model perturbation $\delta\mathbf{m}$ is equivalent to the application of the forward first-order Born modeling operator followed by the application of its adjoint. In other words, it corresponds to Born modeling followed by migration. This kernel corresponds to the Gauss-Newton component of the full Hessian. From now on, we will denote it as $\mathbf{H}_{\mathrm{GN}}$.

Next we look at the interpretation of $\mathbf{K}^{a2}$ and we analyze the source term in equation 17. This term corresponds to the first-order Born scattering of $\mathbf{u}_1^{'}$, which is the the backpropagated wavefield generated by injecting the data residuals at the receivers (equation 14). The wavefield $\mathbf{u}_{22}^{'}$ is thus obtained by: 1) computing the primary adjoint wavefield, $\mathbf{u}_1^{'}$, 2) scattering $\mathbf{u}_1^{'}$ at $\delta\mathbf{m}$, and then, 3) backpropagating the scattered wavefield backward in time. The kernel $\mathbf{K}^{a2}$ is then evaluated by cross-correlating $\mathbf{u}_{22}^{'}$ with the the second time derivative of the forward propagated background wavefield $(\nabla_m \mathbf{L}(\mathbf{m}_0, \mathbf{u}_0))$ as indicated by equation 19.

The kernel $\mathbf{K}^{a2}$ is similar to the *source* side of the adjoint of the conventional WEMVA operator, with an important difference (Sava and Vlad, 2008). The wavefield $\mathbf{u}_{22}^{'}$ is generated by injecting the data residuals, not the recorded data, as is done for the conventional WEMVA operator. However, because the operator is linear with respect to the injected wavefields, it can also be interpreted as the difference of two WEMVA operators: the first, $\mathbf{W}_s^m$, is obtained by injecting the modeled data $\mathbf{d}_m$ at the receivers, whereas the second, $\mathbf{W}_s^o$, is obtained by injecting the observed data $\mathbf{d}_o$. This second operator is exactly equivalent to the adjoint of the conventional WEMVA operator.

The next, and final, step of our analysis is to interpret the kernel $\mathbf{K}^b$, as defined by equation 10. We first notice that because of the linearity of $\mathbf{L}$ with respect to the wavefield we can write the following identity,

$$\nabla_m \nabla_u \mathbf{L}(\mathbf{m}_0, \mathbf{u}_0) = \nabla_m \mathbf{L}(\mathbf{m}_0, \mathbf{u}_0). \tag{22}$$

Then we notice that, as for equation 16, the derivative of the wavefield with respect to the model parameters is the solution of the forward first-order Born scattering problem expressed in equation 20. Therefore, the applications of kernel $\mathbf{K}^b$ to the model perturbation $\delta\mathbf{m}$ is evaluated by cross-correlating the forward scattered wavefield $\delta\mathbf{u}$ with the second time derivative of the backward propagated primary adjoint wavefield $\mathbf{u}_1^{'}$. This kernel $\mathbf{K}^b$ is similar to the *receiver* side of the adjoint of the conventional WEMVA operator. As discussed above for $\mathbf{K}^{a2}$, the operator $\mathbf{K}^b$ can also be interpreted as the difference of two WEMVA operators: the first, $\mathbf{W}_r^m$, is obtained by injecting the modeled data $\mathbf{d}_m$ at the receivers, whereas the second, $\mathbf{W}_r^o$, is obtained by injecting the observed data $\mathbf{d}_o$.

In conclusions the full Hessian operator $\mathbf{H}$ is the composition of familiar operators, as follows

$$\mathbf{H} = \mathbf{H}_{\mathrm{GN}} + (\mathbf{W}_s^m - \mathbf{W}_s^o) + (\mathbf{W}_r^m - \mathbf{W}_r^o) = \mathbf{H}_{\mathrm{GN}} + (\mathbf{W}^m - \mathbf{W}^o) = \mathbf{H}_{\mathrm{GN}} + \mathbf{H}_{\mathrm{W}}, \tag{23}$$

where $\mathbf{W}^m$ and $\mathbf{W}^o$ are the WEMVA operators obtained by injecting the modeled and observed data, respectively; $\mathbf{H}_{\mathrm{W}}$ is the difference between $\mathbf{W}^m$ and $\mathbf{W}^o$. We will refer to this difference as the *differential WEMVA* component of the full Hessian.

If the wave-equation modeling operator ($\mathbf{L}$) is sufficiently accurate to model all the important features present in the recorded data, as the inversion process converges to a model closer to the true model, the modeled data becomes closer to the observed data. In this situation, the corresponding WEMVA operators $\mathbf{W}^m$ and $\mathbf{W}^o$ become closer to each other; their difference becomes negligible and the full Hessian is equivalent to the simple Gauss-Newton Hessian. However, when the wave-equation modeling operator ($\mathbf{L}$) is too simplistic to model correctly important wave phenomena (e.g. elastic, multiples, attenuation) the modeled data does not converge to the recorded data even close to the global minimum of objective function 1, and consequently the full Hessian may never converges towards the Gauss-Newton Hessian. The implications of this lack of convergence of the full Hessian towards the Gauss-Newton Hessian for Newton methods are not obvious, and may deserve further studies.

Because the gradient is a linear function of the data injected at the surface, we can consider the gradient $\nabla\phi$ as the difference of two terms corresponding to the modeled data ($\nabla\phi^m$) and the observed data ($\nabla\phi^o$); that is,

$$\nabla\phi = (\nabla\phi^m - \nabla\phi^o)\,. \tag{24}$$

Then, following equation 6, the first search direction, $\Delta\mathbf{m}_1$, of a truncated Newton algorithm can be decomposed in 6 terms as follows:

$$\begin{aligned}
\Delta\mathbf{m}_1 \quad = \quad & -\mathbf{H}\nabla\phi = \\
- \quad & \mathbf{H}_{\mathrm{GN}}\nabla\phi^m && (25)\\
+ \quad & \mathbf{H}_{\mathrm{GN}}\nabla\phi^o && (26)\\
- \quad & \mathbf{W}^m\nabla\phi^m && (27)\\
+ \quad & \mathbf{W}^m\nabla\phi^o && (28)\\
+ \quad & \mathbf{W}^o\nabla\phi^m && (29)\\
- \quad & \mathbf{W}^o\nabla\phi^o. && (30)
\end{aligned}$$

In the numerical example section we will show examples each of these terms and discuss their roles.

Observations

Equation 23 shows the decomposition of the full FWI Hessian into two operators. We now discuss each term, and discuss its role in a truncated Newton FWI solution.

Gauss-Newton Hessian

The Gauss-Newton component of the Hessian ($\mathbf{H}_{\mathrm{GN}}$) describes the correlation between model parameters that are caused by the finite-resolution nature of the seismic imaging process (data-acquisition followed by migration). The resolution limits are determined by the finite frequency bandwidth and finite aperture of realistic seismic acquisitions, or in the case of multi-parameter inversion (anisotropic, elastic, visco-acoustic), the trade off between the parameters used to describe the unknown medium. Using an image-processing terminology, the application of $\mathbf{H}_{\mathrm{GN}}$ to a model perturbation is the convolution with non-stationary point-spread functions which takes into account the finite resolution of the seismic imaging system. Except in regions of the model that suffer from severe lack of illumination (i.e. shadow zones) the Gauss-Newton Hessian is diagonally dominant (or block diagonally dominant in case of multi-parameters inversion). However, $\mathbf{H}_{\mathrm{GN}}$ is also a singular matrix and thus its inversion must be regularized.

The application of the Gauss-Newton Hessian ($\mathbf{H}_{\mathrm{GN}}$) operator and its approximate inversion have been discussed in several SEP thesis (Valenciano, 2008; Tang, 2011; Ayeni, 2011) and in the wider literature. It is related to iterative linearized waveform inversion (Kuehl and Sacchi, 1999; Prucha et al., 2000) and least-squares migration (Jin et al., 1991; Cole and Karrenbach, 1992; Schuster, 1993; Nemeth et al., 1999).

In a single-parameter acoustic inversion the application of the approximate inverse of $\mathbf{H}_{\mathrm{GN}}$ to a migrated image improves resolution and the relative amplitudes among migrated reflectors. If appropriate regularization terms are added to the objective function in equation 1, inverting $\mathbf{H}_{\mathrm{GN}}$ can compensate for poor illumination caused by complex overburden (Clapp, 2005; Valenciano, 2008; Tang, 2011; Fletcher et al., 2012), or better equalize time-lapse images collected with different acquisition geometries (Ayeni, 2011).

When the subsurface is parametrized using more than one parameter, the inversion of $\mathbf{H}_{\mathrm{GN}}$ helps the resolution of the the trade off between model parameters and thus it can substantially improve convergence (Baumstein, 2014; Tang and Lee, 2015). For this application, $\mathbf{H}_{\mathrm{GN}}$ can be effectively approximated as a block-diagonal matrix since the modeling of the cross-talk between different parameters is its crucial contribution.

The application of $\mathbf{H}_{\mathrm{GN}}$ to a model-perturbations vector $\delta\mathbf{m}$ using the methodology introduced in this section requires two wavefield propagations: 1) forward propagation of the wavefield scattered by $\delta\mathbf{m}$ (equation 20), and 2) backward propagation of the scattered wavefield recorded and reinjected at the receivers (equation 21). Alternatively, because $\mathbf{H}_{\mathrm{GN}}$ is diagonally dominant, it can be approximated by a banded matrix and effectively precomputed and stored. The computation of the dominant elements of $\mathbf{H}_{\mathrm{GN}}$ can be approximated by phase-encoding methods (Tang, 2011; Deuzeman and Plessix, 2015) or by interpolation of the results of modeling and migrating isolated model perturbations (Fletcher et al., 2012; Tang and Lee, 2015).

Differential WEMVA component of the Hessian

The differential WEMVA component of the Hessian ($\mathbf{H}_{\mathrm{W}}$) describes the interaction between model parameters caused by multiple-scattering during wave propagation. In particular, the $\mathbf{H}_{\mathrm{W}}$ operator models the effects of second-order scattering. This second-order scattering can be both *short range* or *long range* (see Figure 4).

The long-range scattering is related to velocity errors that cause defocusing of the gradient. The application of ($\mathbf{H}_{\mathrm{W}}$) to the FWI gradient $\nabla\phi$ is equivalent to the gradient that we would obtain from a WEMVA method that aims to maximize the power of the stacked image, where the image is the FWI gradient itself, with a negative sign. The maximization of the image stack power is known to be prone to cycle-skipping problems when the background velocity is far away from the true velocity and the gradient is substantially misfocused. When the background model is far from the true one, the long-range component of the Hessian operator also cycle skips and points in the wrong direction (Figure 4(d)).

The short-range scattering of the WEMVA operator takes into account the interactions between nearby scatterers. It is related to amplitude and phase errors in the gradient caused by the first-order Born approximation. When the nearby scatterers are part of nearby reflectors, they may create internal multiples that are not modeled correctly by the first-order Born approximation. When these nearby scatterers are part of the same reflecting interface, neglecting their influence leads to errors in reflections amplitudes. For example, it is well-known that when we model reflection amplitudes from a planar interface using the first-order Born approximation we do not obtain the same amplitudes predicted by the Zoeppritz equations. Consequently, a linearized-Born inversion yields biased estimates of elastic properties at interfaces. To remove this bias, we must iterate the linearized-Born inversion in a fully non-linear algorithm.

The inclusion of the short-range scattering components of $\mathbf{H}_{\mathrm{W}}$ should improve elastic-properties estimates even without additional non linear iterations, and should speed up the convergence of a fully non-linear iterative inversion. Further analysis and tests will determine if the improved convergence justifies the additional computational cost.

SYNTHETIC EXAMPLE

To illustrate the effects of applying and inverting the Gauss-Newton and the full Hessian matrices we conducted several numerical tests based on a 2D synthetic dataset. To model the data we assumed a velocity model 3.5 km wide and 1 km deep. The background velocity was 1.5 km/s with a single reflecting layer at depth of 800 m (Figure 1). We modeled 88 regularly spaced sources and recorded the data with a fixed array of 351 receivers. The source interval was 40 m and the receiver interval was 10 m. To compute the Laplacian necessary for the wave propagation, we employed a 10th-order approximation; the minimum wavelength was 30 m and the grid spacing 5 m. As source signature we used a Ricker wavelet with central frequency of 10 Hz.

To invert the Hessian matrices we relied on the Intel MKL iterative sparse solvers (ISS) based on the reverse communication interface (RCI) (Dongarra et al., 1995). In our test we did not add a regularization term to the objective function. However, if necessary, we could regularize the inversion or apply the Leverberg-Marquardt technique to stabilize the inversion of the Hessian matrix (Pujol, 2007).

Figure 1 shows the velocity model used to model the data. Figure 2 shows one of the modeled shot gathers after muting the direct arrival. We run two iterations of FWI, with different choices of starting models and approximation of the Hessian matrix.

First-iteration results

Figure 3(a) shows the image (opposite of FWI gradient) obtained by assuming the correct background velocity, but no reflecting interface. The imaged reflector is well focused, though resolution suffers because of the effects of the finite-bandwidth wavelet. Figure 3(b) shows the image (opposite of FWI gradient) obtained by assuming a background velocity lower than the correct one by 2%. The image is not as well focused as in Figure 3(a) and the phase of the reflector is distorted. Finally, Figure 3(c) shows the image (opposite of FWI gradient) obtained by assuming a background velocity with a larger error (10%) than in the previous case; as expected, the phase of the reflector is more distorted than in the previous case.

Figure 4(a) is the search direction obtained by applying the full Hessian to the image obtained by assuming a background velocity too low by 2 % (image in Figure 3(b)). As explained in the previous section, this search direction ($\mathbf{\Delta m_1}$ in equation 6) is the first search direction of the iterative inversion of the Hessian matrix in a truncated Newton algorithm. Because we started from a constant model that did not produce any reflections, the modeled data after muting the first arrival is uniformly zero. Therefore, the application of the full Hessian to the image corresponds to only two terms out of the general 6 terms; that is, $\mathbf{H}_{\rm GN}\nabla\phi^o$ (equation 26) and $-\mathbf{W}^o\nabla\phi^o$ (equation 30). Figure 4(b) is the search direction obtained by applying the Gauss-Newton Hessian (equation 26) to the same image as the results shown in Figure 4(a). In this case, the long-wavelength component is absent.

We set the clip to be the same for displaying the two top panels in Figure 4; this clip was chosen to make visible the effects of applying to the gradient the WEMVA component of the Hessian easily visible.

Figure 4(c), shows the result of applying the WEMVA component of the Hessian operator that is, it is the difference between Figure 4(a) and Figure 4(b). The long wavelength of the model update points towards a negative slowness-squared update, and thus it points in the correct direction. In contrast, Figure 4(d) shows the result of applying the WEMVA component of the Hessian operator to the image obtained by assuming a background velocity lower by 10% than the correct one (Figure 3(c)). The long wavelength of the model update points towards a positive slowness-squared update, and thus it points in the wrong direction.

Figure 5(a) shows the steepest-descent velocity update after a simple line search. Figure 5(b) shows the truncated Newton model update after iteratively inverting the Gauss-Newton Hessian and without line search (i.e. setting the step size to be equal to one). Figure 5(c) shows the truncated Newton model update after iteratively inverting the full Hessian and without line search. For both truncated Newton results the updated model shows a sharper image of the reflector than for the simple steepest descent result.

Figure 6(a) shows a vertical section of the three models shown in Figure 5. The updated model obtained by inverting the Gauss-Newton Hessian (green line) is similar to the model obtained by inverting the full Hessian (red line). There is only a small difference in the long-wavelengths component above the reflector, as shown in the zoomed-in plot displayed in Figure 6(b). The model updated by inverting the full Hessian has a small positive increase in the background velocity to correct for the initial error in velocity.

Figure 7(a) shows the normalized norm of the residuals (in model space) during the iterative inversion of the full Hessian. We did not regularize this inversion that started to diverge after 4 iterations. Figure 7(b) shows the normalized norm of the residuals during the iterative inversion of the Gauss-Newton Hessian. In this case the inversion converges even without regularization.

Figure 8 shows the same comparison shown in Figure 6(a), but corresponding to the case when the background velocity was correct. Now the inversion of the full Hessian is not trying to update the background model and the result is similar to the inversion of only the Gauss-Newton component of the Hessian. The applications of the inverse Hessians had the expected effect to reduce the side lobes in the image, and the updated models show a nicely focused reflector.

Second-iteration results

We computed the gradient and Hessian components for a second iteration of a FWI process. The background model for the second iteration was the one obtained by using steepest descent at the first iteration; that is, the one that is shown in Figure 5(a). Figure 9 illustrates the contributions to the total search direction (opposite of the gradient) of the two components of the data residuals: the modeled and the observed data (equation 24). Figure 9(a) shows the contribution of the modeled data, whereas Figure 9(b) shows the contribution of the observed data. The latter search direction contains a long-wavelength component

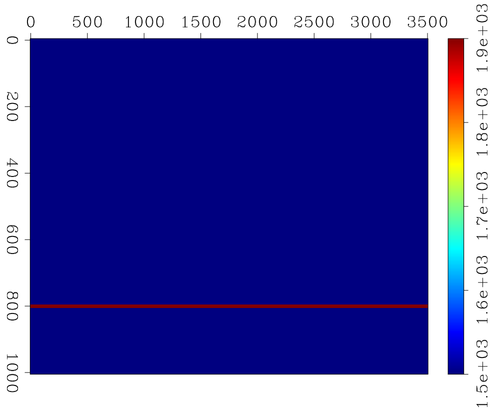


Figure 1: True velocity model with a single reflector (background 1.5 km/s perturbation .4 km/s) [**ER**] biondo1/. true-model

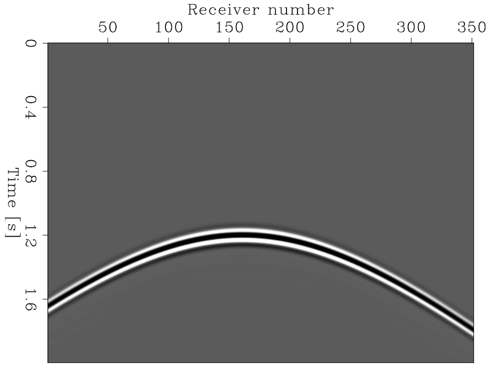


Figure 2: Common-shot gather of the observed data. Shot and receiver interval is 40 m and 10 m, respectively. [**ER**] biondo1/. true-data

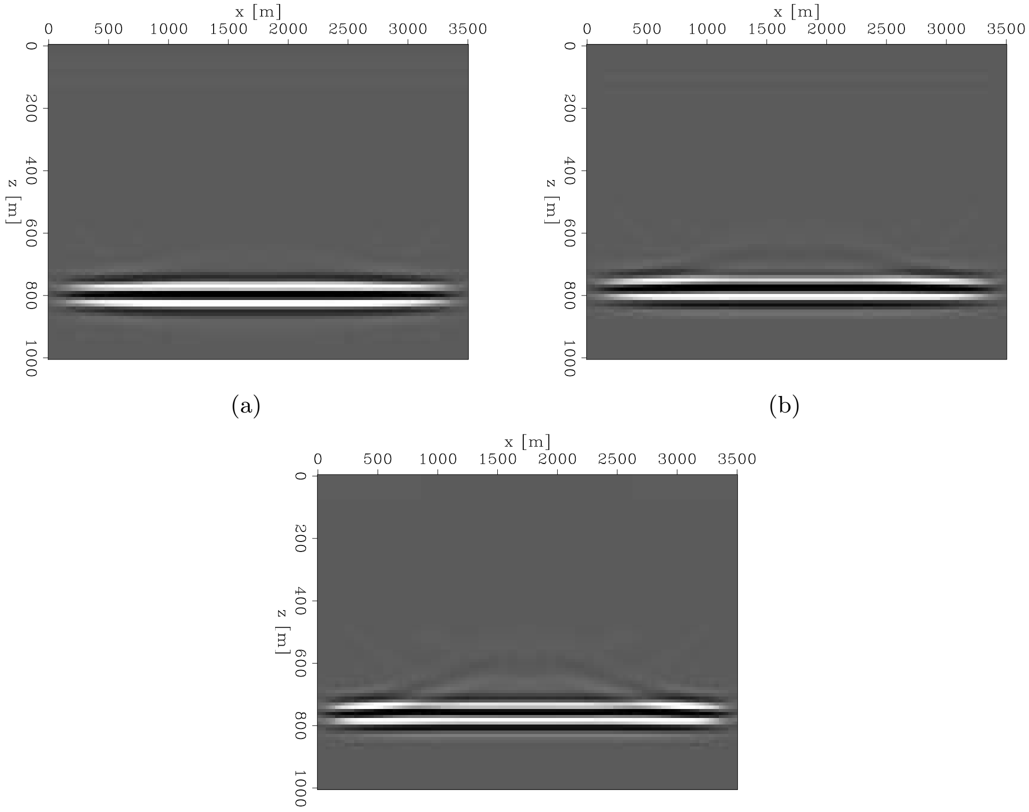
that correctly tries to increase the background velocity (decrease slowness squared). Figure 9(c) shows the sum of the two components. As for Figure 4, we set the clip to be the same for displaying all the panels in Figure 9 and this clip was chosen to make visible the long-wavelength components of the images.

Figure 10 shows the application of the Gauss-Newton component of the full Hessian to the search direction obtained by computing the total gradient; that is, the search direction shown in Figure 9(c).

Figure 11 shows the application of the four components of the differential WEMVA operator to the search direction shown Figure 9(c): Figure 11(a) shows the $-\mathbf{W}^m \nabla \phi^m$ term (equation 27), Figure 11(b) shows the $+\mathbf{W}^m \nabla \phi^o$ term (equation 28), Figure 11(c) shows the $+\mathbf{W}^o \nabla \phi^m$ term (equation 29), and Figure 11(d) shows the $-\mathbf{W}^o \nabla \phi^o$ term (equation 30). Notice that only the mixed terms (($\mathbf{W}^m \nabla \phi^o$ and $\mathbf{W}^o \nabla \phi^m$) contain significant long-wavelength components; these components have opposite signs between the two panels. Fortunately, the term that has the correct sign ($\mathbf{W}^m \nabla \phi^o$) prevails and the application of the full Hessian (Figure 12) shows a long-wavelength component pointing in the correct direction.

DISCUSSIONS

The theoretical insight presented in the first section of the paper, and illustrated in the last section, has several potential applications. In this section we speculate and elaborate some of these potential applications.

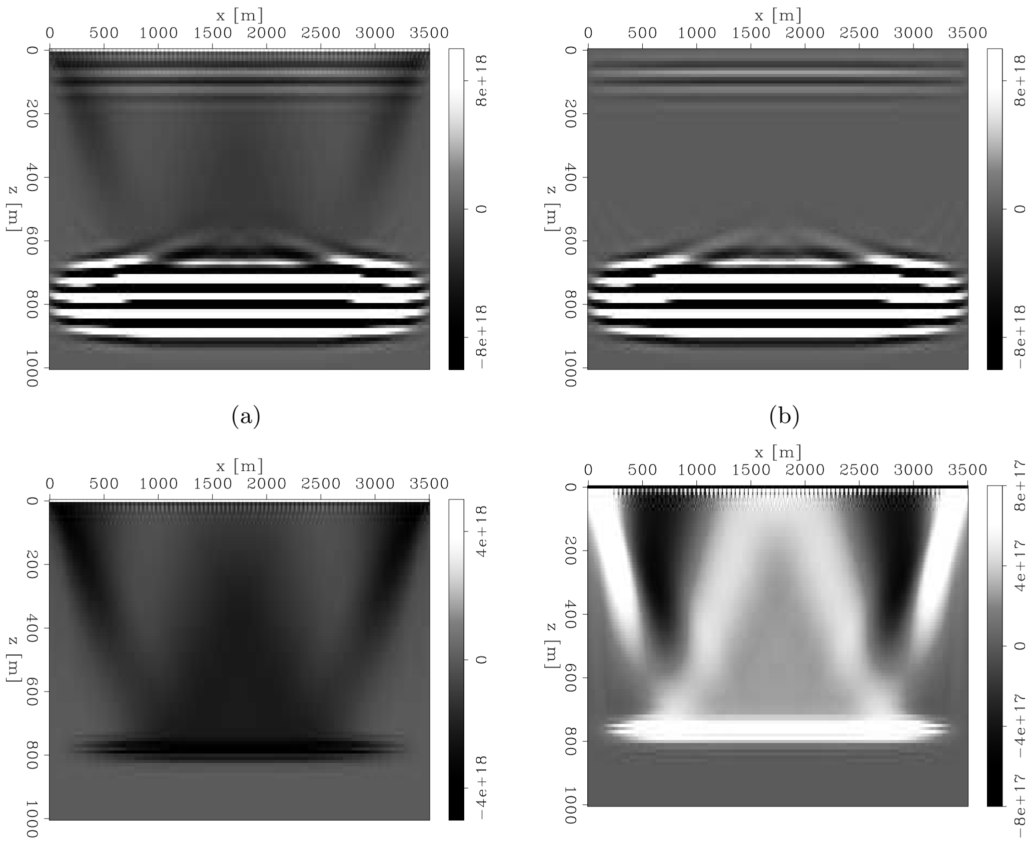


(a)

(b)

(c)

Figure 3: (a) Image computed with the correct background of 1.5 km/s. (b) Image obtained with a background with a negative velocity error of 2.0%. (c) Image obtained when the background error is -10%. [**ER**] biondo1/. true-image,image,image-10



(a)

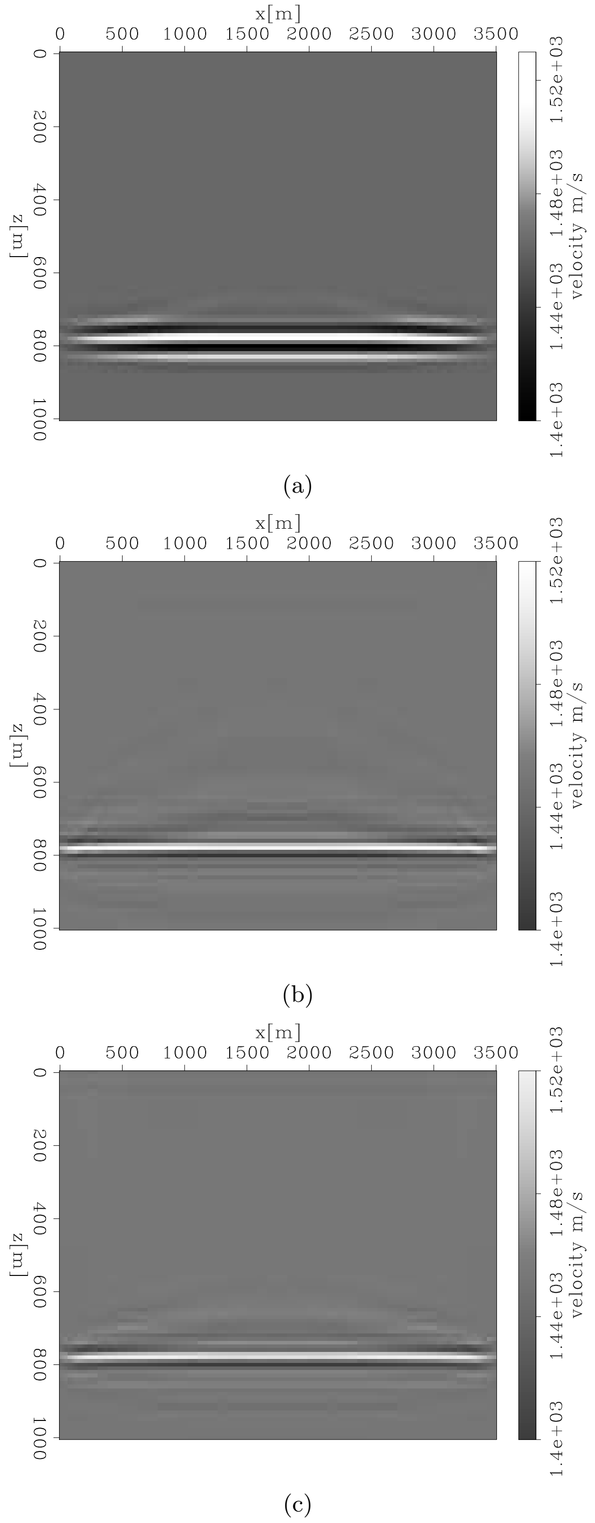
(b)

(c)

(d)

Figure 4: (a) Full and (b) Gauss-Newton Hessian applied to first search direction of Figure 3(b) (i.e., $-\mathbf{H}\nabla\phi$). (c) Difference between full and Gauss-Newton Hessians. Panel (a) and (b) have the same clipping value and they correspond to the first search direction during the Hessian inversion. (d) Hessian WEMVA component applied to the first search direction when the velocity error is -10% (Figure 3(c)). Because of the large velocity error in this case, the WEMVA search direction is pointing toward the incorrect velocity update. [**CR**]

biondo1/. Full-Newton-g0,Gauss-Newton-g0,Hessian-diff-g0,Hessian-wemva-g0-10

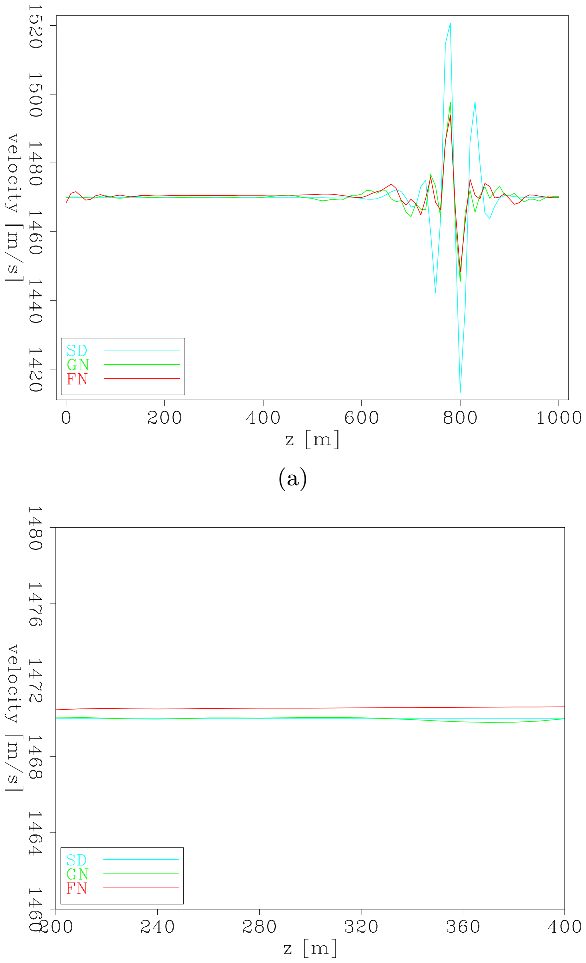


(a)

(b)

(c)

Figure 5: (a)Updated model using steepest descent with line search (parabolic fitting). (b) Updated model by inverting the Gauss-Newton Hessian without performing line search. (c) Same as panel (b) but inverting the full Hessian. [**CR**] biondo1/. steppest-model,GN-model,NW-model



(a)

(b)

Figure 6: (a) Vertical section along middle position in the model for comparing updated model using the three different schemes. (b) Close-up on the depth sections from 200 m to 400 m. Notice that the full-Newton update starts updating also the low-wavenumber component of the model above the reflector. (SD=steepest descent, GN=Gauss-Newton, FN=full-Newton) [**CR**] biondo1/. compare-models,compare-models-zoom

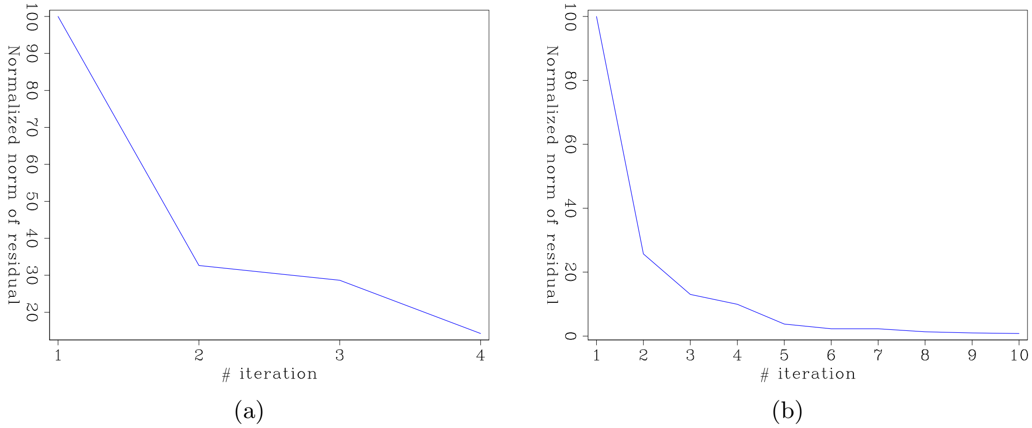


Figure 7: (a) full Hessian iterative inversion convergence, after four iterations the inversion starts diverging. (b) Gauss-Newton Hessian iterative inversion convergence. [**CR**] biondo1/. NW-convergence,GN-convergence

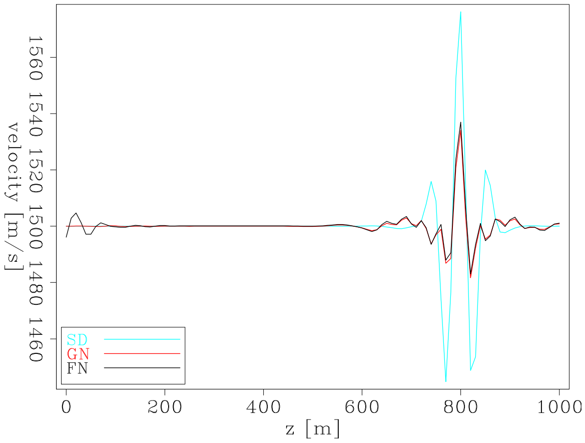
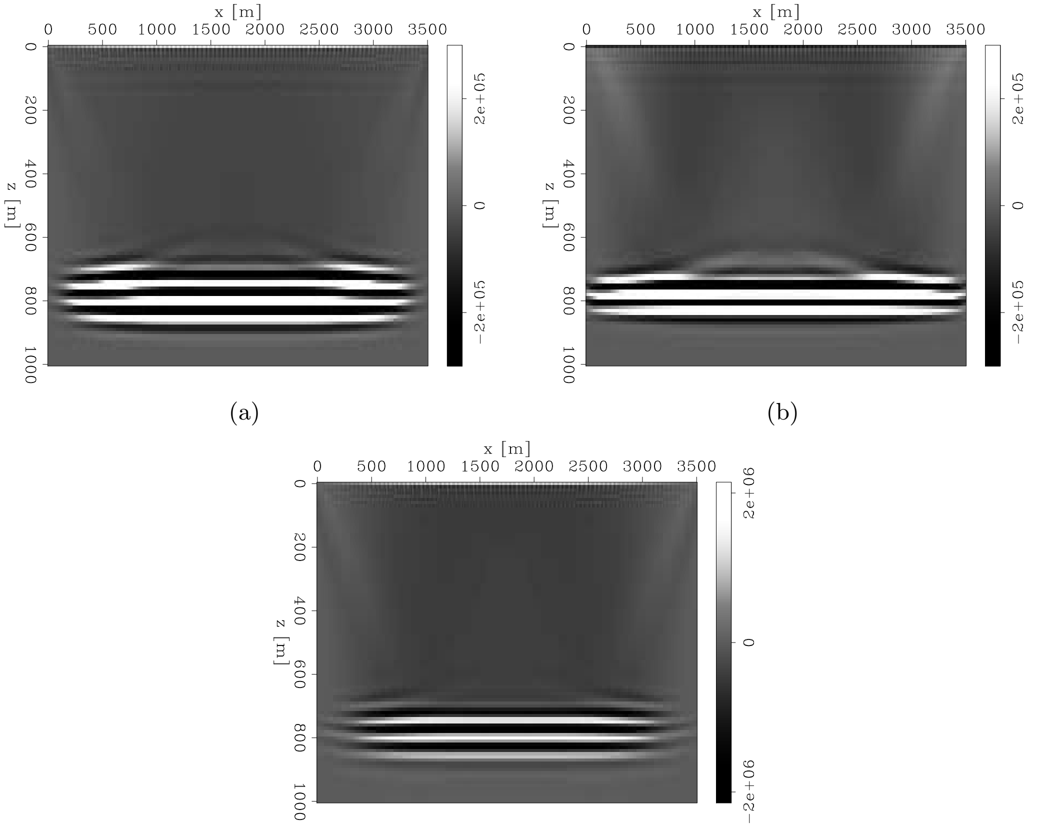


Figure 8: Vertical section along middle position in the model for comparing updated model using steepest descent, the Gauss-Newton, and full-Newton methods using the correct velocity background. As expected Hessian inversions is deconvolving the wavelet effect present in the steepest descent model. It is worth noticing that using the correct background the Gauss-Newton and full-Newton inversion schemes provide almost the same velocity updates. (SD=steepest descent, GN=Gauss-Newton, FN=full-Newton) [**CR**] biondo1/. compare-true-models



(a)

(b)

(c)

Figure 9: Search direction at the second iteration using the model of Figure 5(a). (a) Search direction given by the modeled data (i.e., $-\nabla\phi^m$). (b) Search direction given by the observed data (i.e., $\nabla\phi^o$). (c) Sum of panel(a) and panel(b) that is the total search direction (i.e., $-\nabla\phi = \nabla\phi^o - \nabla\phi^m$). The long-wavelengths of the total search direction try to decrease the slowness (and increase velocity) on top of the reflector. [**CR**]

biondo1/. second-image-dmod,second-image-dobs,second-image

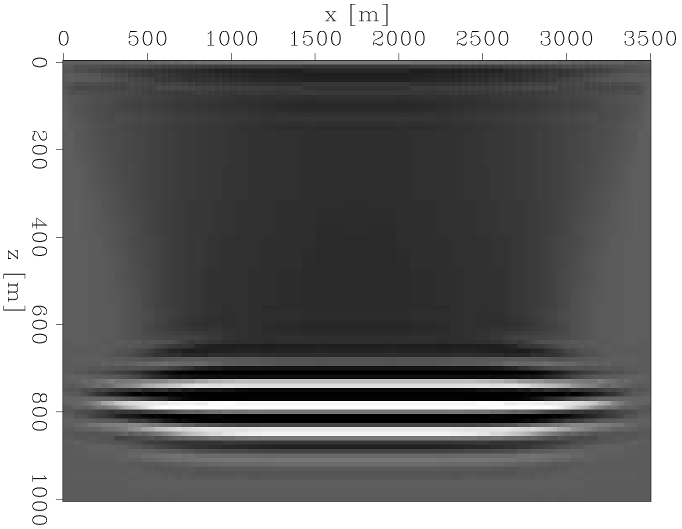
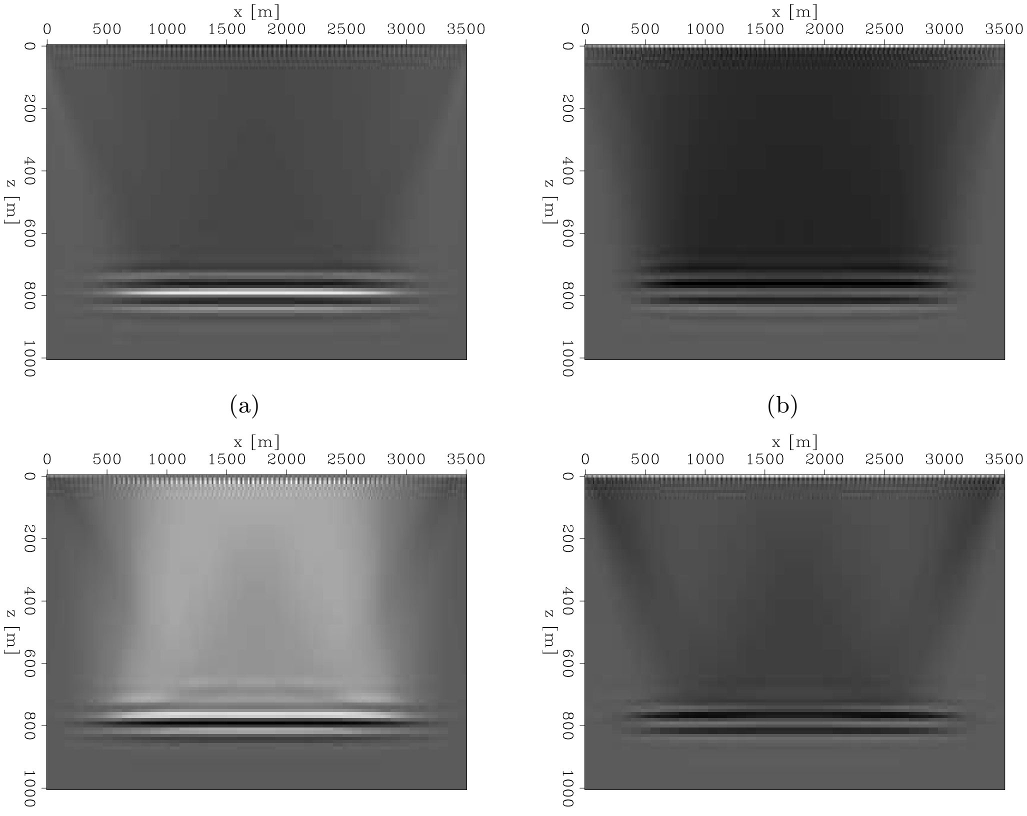


Figure 10: Gauss-Newton Hessian applied to the second search direction (Figure 9(c)). [**CR**] biondo1/. GN-second-iter



(a) (b) (c) (d)

Figure 11: Terms of the WEMVA component of the full Hessian at the second iteration: (a) $-\mathbf{W}^m \nabla \phi^m$ (equation 27). (b) $+\mathbf{W}^m \nabla \phi^o$ (equation 28). (c) $+\mathbf{W}^o \nabla \phi^m$ (equation 29). (d) $-\mathbf{W}^o \nabla \phi^o$ (equation 30). [**CR**] biondo1/. WEMVApre-mod,WEMVApre-obs,WEMVAobs-mod,WEMVAobs-obs

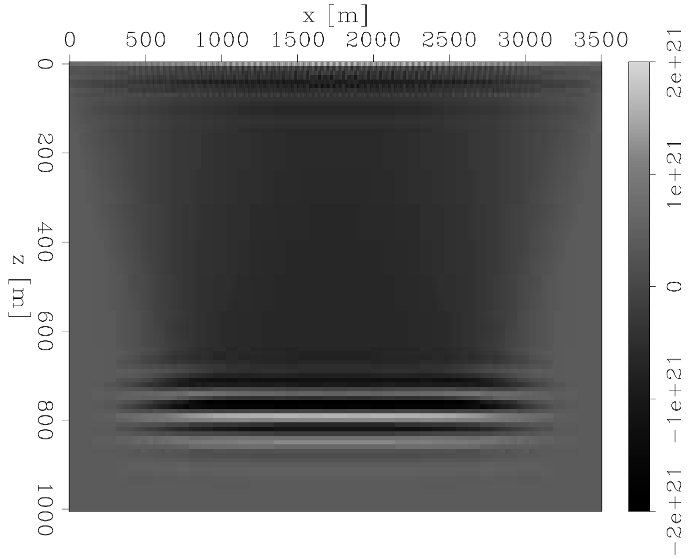


Figure 12: Full Hessian applied to the second search direction (Figure 9(c)). [**CR**] biondo1/. second-full-hessian

One application could be the improving the convergence of a truncated Newton FWI algorithm when the starting model is inaccurate. As demonstrated in one of the examples above, the WEMVA component of the full Hessian is not immune from cycle skipping when the starting model is sufficiently inaccurate. However, in contrast with the cycle-skipping occurring when the data residuals are backprojected into the gradient, it would be easier to avoid the cycle-skipping of the Hessian. A natural way of avoiding cycle skipping could be to extend the model as it is done in conventional WEMVA and TFWI. Furthermore, the detailed analysis of all the four terms in the WEMVA Hessian (equations 27–30 and Figure 11) could lead to new ways of avoiding the cycle-skipping problem.

The decomposition of the Hessian into a Gauss-Newton and a WEMVA component enables the separation of the long wavelengths of the model (velocity) from the short wavelengths (reflectivity) during the inversion. We could subtract the result of applying the inverse of the Gauss-Newton Hessian to the gradient from the application of the whole Hessian, as we did to create the images shown in Figures 4(c) and 4(d). The model updates resulting from this subtraction is dominated by the long-wavelength components of the total update. After the long-wavelengths of the model are accurately estimated, conventional FWI could be safely applied to estimate all the wavelengths in the model.

Another possible application of the insight presented in this paper is to reduce substantially the computational cost of target-oriented fully non-linear elastic FWI solution. Such a solution is attractive because it should lead to a more accurate estimation of elastic reservoir parameters and thus improve reservoir characterization.

If the background model can be assumed to be accurate, the Gauss-Newton component of the Hessian can be precomputed applying one of methods presented by Tang (2011); Deuzeman and Plessix (2015); Fletcher et al. (2012); Tang and Lee (2015), and only the near-range scattering of the differential WEMVA operator needs to be taken into account.

Substantial computational savings can be achieved because repeated applications of the near-range scattering components of the differential WEMVA operator require only local propagation of wavefields. Once the background source wavefield ($\mathbf{u_0}$ in equation 12) is propagated from the source to the target and the data residuals are backpropagated ($\mathbf{u_1^{'}}$ in equation 14) from the surface to the target the application of $\mathbf{K}^{a2}$ and $\mathbf{K}^{b}$ to a model perturbation require only the propagations and scattering of wavefields within the target volume. Therefore, after precomputing the Gauss-Newton component and the wavefields $\mathbf{u_0}$ and $\mathbf{u_1^{'}}$, the full Hessian can be iteratively inverted only by local propagation and scattering. Although in this paper we discussed only the FWI Hessian related to the acoustic wave equation, this requirement of only local computations holds also in the case of the elastic wave equation. Consequently, expensive long-range elastic propagation are needed only at each non-linear iteration and are not needed for the iterative inversion of the full Hessian. This observation could lead to the development of computationally affordable target-oriented fully-elastic inversion algorithms.

REFERENCES

Alves, G., 2015, Overview of the apache forties data set: SEP-Report, **160**, 173–176.

Ayeni, G., 2011, Time-lapse seismic imaging by linearized joint inversion: PhD thesis, Stanford University.

Baumstein, A., 2014, Extended subspace method for attenuation of crosstalk in multi-parameter full wavefield inversion: SEG Technical Program Expanded Abstracts, 213, 1121–1125.

Biondi, E. and O. O'Reilly, 2015, Two-way wave-equation operators for non-constant density acoustic isotropic media: SEP-Report, **160**, 81–100.

Clapp, M. L., 2005, Imaging under salt: Illumination compensation by regularized inversion: PhD thesis, Stanford University.

Cole, S. and M. Karrenbach, 1992, Least-squares Kirchhoff migration: SEP-Report, **75**, 101–110.

Deuzeman, A. and R.-E. Plessix, 2015, Block-diagonal approximation of the Hessian for multi-parameter FWI: 12th SIAM Conference on Mathematical and Computational Issues in Geosciences, Expanded Abstracts, MS2, Society of Industrial and Applied Mathematics.

Dongarra, J., V. Eijkhout, and A. Kalhan, 1995, Reverse communication interface for linear algebra templates for iterative methods: University of Tennessee, Knoxville, TN.

Epanomeritakis, I., V. Akçelik, O. Ghattas, and B. J., 2008, A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion: Inverse Problems, **24**, 34015–34041.

Fichtner, A., 2011, Full seismic waveform modelling and inversion: Springer Science & Business Media.

Fichtner, A. and J. Trampert, 2011, Hessian kernels of sesimic data functionals based upon adjoint techniques: Geophysical Journal international, **185**, 775–798.

Fletcher, R. P., S. Archer, D. Nichols, and W. Mao, 2012, Inversion after depth imaging: SEG Technical Program Expanded Abstracts, 546, 1–5.

Jin, S., R. Madariaga, J. Virieux, and G. Lambare, 1991, Two-dimensional asymptotic iterative elastic inversion: SEG Technical Program Expanded Abstracts 1991, 282, 1013–1016.

Kuehl, H. and M. D. Sacchi, 1999, Least-squares split-step migration using the Hartley transform: SEG Technical Program Expanded Abstracts, 396, 1548–1551.

Métivier, L., R. Brossier, J. Virieux, and S. Operto, 2012, The truncated Newton method for full waveform inversion: SEG Technical Program Expanded Abstracts, 480, 1–5.

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

Pratt, R. G., C. Shin, and G. J. Hicks, 1998, Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Prucha, M. L., R. G. Clapp, and B. Biondi, 2000, Seismic image regularization in the reflection angle domain: SEP-Report, **103**, 109–119.

Pujol, J., 2007, The solution of nonlinear inverse problems and the levenberg-marquardt method: Geophysics, **72**, no. 4, W1–W16.

Sava, P. and I. Vlad, 2008, Numeric implementation of wave-equation migration velocity analysis operators: Geophysics, **73**, no. 5, VE145–VE159.

Schuster, G. T., 1993, Least-squares cross-well migration: SEG Technical Program Expanded Abstracts, 28, 110–113.

Shen, Y., 2015, Simultaneous inversion of velocity and q using wave-equation migration analysis: SEP-Report, **160**, 71–80.

Tang, Y., 2011, Imaging and velocity analysis by target-oriented wavefield inversion: PhD thesis, Stanford University.

Tang, Y. and S. Lee, 2010, Preconditioning full waveform inversion with phase-encoded Hessian: SEG Technical Program Expanded Abstracts, 204, 1034–1038.

———, 2015, Multi-parameter full wavefield inversion using non-stationary point-spread functions: SEG Technical Program Expanded Abstracts, 215, 1111–1115.

Valenciano, A. A., 2008, Imaging by wave-equation inversion: PhD thesis, Stanford University.

SEPLib nonlinear solver library – Manual

Ali Almomin, Ettore Biondi, Yinbin Ma, Kaixi Ruan, Joseph Jennings, Robert Clapp,
Musa Maharramov and Alejandro Cabrales-Vargas

ABSTRACT

We created an SEPLib nonlinear solver library that serves as the infrastructure for many inversion implementations. The library targets gradient-based inversion schemes with a focus on ease of use, flexibility, reusability and expandability. This was achieved by taking advantage of modern language features in object-oriented programing. This manual is divided in three parts. First, we detail what the library components are and explain all the abstract parts. Second, we show how to use each component of the library. Third, we show examples of different inversion implementations.

INTRODUCTION

The purpose of this manual is to provide a reference for the SEPLib nonlinear inversion library. The design of the library is based on a divide and conquer approach where we separate the inversion into four parts:

- **Problem definition** which includes the details of how the objective function and gradient are computed
- **Stepper** which finds a proper scalar to a given search direction to reduce the objective function
- **Terminator** which determines when to stop the inversion
- **Solver** which runs the iterative inversion by using the other three parts

Our goal is to make each part independant from the other parts. This separation increase reusability of any part that doesn't need to be changed between different inversion problems.

Each part has an abstract object that define its interface, variables and procedures. By starting with an abstract base, it is possible to create as many implementations as we need as long as they adhere to the abstract object requirements. This design allows for an easy expansion of the library with more implementations in the future. We also created several concrete implementations of each abstract object using conventional inversion techniques. Moreover, we created a driver, i.e. a wrapper, that uses all the concrete implementations of the library.

The manual is divided in three sections. The first section details the library components and explains all the abstract objects for the users interested in the computer science part of the library. The second section shows how to create a concrete implementation of each abstract object and lists all implementation we created so far. This section targets users that are interested in creating their own solver and steppers or want to define their objective function with a nonconventional norm or residual. The third sections shows examples of different inversion problems targetting users that want to run the existing implementations.

ABSTRACT COMPONENTS

problem

The problem abstract objects mainly store functions pointers that calculate objective function, the residual, the gradient and the change in residual. However, the library separates different components of the inversion which can potentionally cause some recalculations to occur. Therefore, we decided to prevent such recalculation by following an encapsulation approach with this abstract object such that it stores a private local copy of the input and output variables with "get" procedures that output the needed variable. We added a simple dependency between the variable with "updated" logical flags such that changing the dependencies causes the variables to be recalculated. We decided to add output procedure that writes the local variables to disk as the inversion is running. Finally, we added counters to keep track of the number of evaluations and writes that occured so far since this is the object where the calculations occur.

The following shows the definition of the abstract object:

```
type, abstract :: prob_obj
  private
  integer, public :: nmodl = 0, ndata = 0
  integer :: fevals = 0, counter = 0
  logical :: obj_updated = .false., res_updated = .false.
  logical :: grad_updated = .false., dres_updated = .false.
  double precision :: obj = 0.
  real, dimension(:), allocatable :: modl, res, grad, dmodl, dres
  character(256), public :: rnorm = '', mmov = '', gmov = '', rmov = ''
end type
```

The variables `nmodl` and `ndata` store the size of the model and data space. The variables `fevals` and `counter` keep track of the number of evaluations and writes, the variables ending with `_updated` are the update flag for each local variable. The variable `modl` is the model, `res` is the residual which depends on `modl`, `obj` is the objective function value which depends on both `modl` and `res`, `grad` is the gradient which depends on `modl` and `res`, `dmodl` is the model update and `dres` is the change in residual which depends on `modl` and `dmodl`. The variables `rnorm`, `mmov`, `gmov` and `rmov` are the output tags for the residual norm, model movie, gradient movie and residual movie, respectively.

The procedures of this abstract object are:

```
  procedure :: objf, resf, gradf, dresf
  procedure :: alloc, set_modl, output
  procedure :: get_obj, get_res, get_grad, get_dres
  procedure :: get_fevals, get_counter, get_rnorm, get_gnorm
```

The procedures `objf`, `resf`, `gradf` and `dresf` will store the user functions that calculate each respective variable. Notice that we did use deferred procedures although we need the user to impelment these functions. The reason will be explained later. The `alloc` procedure checks that the sizes were provided then allocates all the local variables as follows:

```
integer function alloc(this)
  class(prob_obj) :: this
  if(this%nmodl < 1 .or. this%ndata < 1) then
    call erexit("problem: nmodl or ndata is not correct")
  end if
  allocate(this%modl(this%nmodl))
  allocate(this%res(this%ndata))
  allocate(this%grad(this%nmodl))
  allocate(this%dres(this%ndata))
  allocate(this%dmodl(this%nmodl))
  alloc = 0
end function
```

The `set_modl` procedure, which is intended to be used internally in this object, checks whether the provided model is different than the local model and updates the local copy if needed. Also, this procedures changes the update flags accordingly as follows:

```
integer function set_modl(this, modl)
  class(prob_obj) :: this
  real, dimension(:) :: modl
  if(any(this%modl .ne. modl)) then
    this%modl = modl
    this%obj_updated = .false.
    this%res_updated = .false.
    this%grad_updated = .false.
    this%dres_updated = .false.
  end if
  set_modl = 0
end function
```

All the `get_` procedures do three steps. First, check if the size of the input is consistent with the local sizes. Then, use the procedure `set_modl` to update the local model (if needed), and finally, calculate the requested variable if it is not updated and add the number of evaluations to `fevals`. If it is updated, the procedure will simply output the local variable. For instance, the following code shows the `set_grad` procedure:

```
integer function get_grad(this, modl, grad)
  class(prob_obj) :: this
  real, dimension(:) :: modl, grad
  integer :: stat
  if(size(modl) .ne. this%nmodl .or. size(grad) .ne. this%nmodl) then
    call erexit("problem: size of modl or res is not correct")
  end if
  stat = this%set_modl(modl)
  if(.not. this%grad_updated) then
    stat = this%get_res(this%modl, this%res)
    stat = this%gradf(this%modl, this%res, this%grad)
    this%fevals = this%fevals + 2
    this%grad_updated = .true.
  end if
  grad = this%grad
  get_grad = 0
end function
```

The `output` procedure outputs the variables if their corresponding tag is used while using `counter` to keep track of how many times it wrote to disk. When writing for the first time, it also creates the history file for the output. The code of the procedure is:

```
integer function output(this, modl)
  class(prob_obj) :: this
  real, dimension(:) :: modl
  integer :: stat
  this%counter = this%counter + 1
  if(this%counter .eq. 1) then
    if(this%mmov .ne. ’’) then
      call to_history("n1",this%nmodl,this%mmov)
      call to_history("label1","Model index",this%mmov)
      call to_history("label2","iteration",this%mmov)
    end if
  end if
  if(this%mmov .ne. ’’) then
    call to_history("n2",this%counter,this%mmov)
    call srite(this%mmov, modl, 4*size(modl))
  end if
  output = 0
end function
```

The procedure `objf`, and similarly `resf`, `gradf` and `dresf`, is impelmented as follows:

```
integer function objf(this, res, obj)
  class(prob_obj) :: this
  real, dimension(:) :: res
  double precision :: obj
  call erexit("ERROR: objf was not overridden")
```

```
    objf = 0
  end function
```

This implementation can be viewed as an optional deferred such that the user does not have to implement functions, such as `dresf`, unless needed by inversion. For the needed functions, the user should overwrite the procedure. While this approach is flexible, it has the downside that the errors will happen in run-time instead of compile-time.

stepper

The stepper abstract object has been designed in such a manner to allow for a variety of steppers to be implemented. In addition to this flexibility, this design also requires very little knowledge from the user when using any one of the steppers. In general, to describe our abstract stepper, we have decided to allow the user to clip the model by specifiying a model maximum (`mmax`) and minimum (`mmin`) in the procedure `clip_modl`. We also included a deferred `run` procedure that should do the actual execution of the stepper. The implementation of this design is shown in the following code:

```
type, abstract :: stepper_obj
    real, dimension(:), pointer :: mmin => null(), mmax => null()
  contains
    procedure(stepper_function), deferred :: run
    procedure :: clip_modl
end type
```

The abstract interface for the run routine is shown in the following code:

```
  abstract interface
    integer function stepper_function(this, modl, prob, dmodl, &
                                      alpha, success)
      import :: stepper_obj, prob_obj
      class(stepper_obj) :: this
      class(prob_obj) :: prob
      real,dimension(:) :: dmodl, modl
      double precision :: alpha
      logical :: success
    end function
  end interface
```

The abstract interface for the run routine requires the user to provide the model array `modl`, a problem object explained before `prob`, the search direction `dmodl`, an `alpha` variable to return the computed step size, and lastly a logical success variable used by the solver to check if the stepper has sucessfully computed a proper step size. If it does not find a proper step size, then when the solver finds that success is .false., it will terminate the program.

The clip_modl procedure has a simple implementation such that the bounds are optional, as shown in the following function:

```
integer function clip_modl(this, modl)
  class(stepper_obj) :: this
  real, dimension(:) :: modl
  if(associated(this%mmin)) where(modl < this%mmin) modl = this%mmin
  if(associated(this%mmax)) where(modl > this%mmax) modl = this%mmax
  clip_modl = 0
end function
```

terminator

The terminator abstract object is composed of a deferred test function and it is defined as follows:

```
type, abstract ::  terminate_obj
contains
  procedure(terminate_test), deferred :: test
end type
```

where abstract interface of this function is written as:

```
abstract interface
  logical function terminate_test(this, iter, prob)
    import :: terminate_obj
    import :: prob_obj
    class(terminate_obj) :: this
    class(prob_obj) :: prob
  end function
end interface
```

The test procedure requires the iteration number iter and a problem object prob. The problem object is required because some stopping criteria depend on the residuals or the gradient of the problem.

solver

The solver abstract object is very simple. In fact, this object is composed only by a deferred run function and it is defined as follows:

```
  type, abstract :: solver_obj
    contains
    procedure(run_solver),deferred :: run
  end type
```

where abstract interface of this function is written as:

```
 abstract interface
    integer function run_solver(this, modl, prob, stpr, term)
      import :: solver_obj, stepper_obj, terminate_obj
      import :: prob_obj
      class(solver_obj) :: this
      real, dimension(:) :: modl
      class(prob_obj) :: prob
      class(stepper_obj) :: stpr
      class(terminate_obj) :: term
    end function
  end interface
```

and requires as input the model array `modl`, a problem object explained before `prob`, a stepper object `stpr`, and a terminator object `term`. We decided to construct the abstract solver object without any initial variable to give more flexibility to a user that wants to construct its own solver. In fact, by including a variable in the abstract object, the user must declare this component in an actual solver object. It is also worth noticing that we separated the solver from the stepper. In this manner we can use different stepper algorithms in combination with an actual solver.

IMPLEMENTATIONS

problem

The problem objects contains the details of how to calculate the residuals, objective fucntion, gradient, and changes in the gradient. When creating a concrete implementation of this object, the user needs to do the following steps:

1. extend the abstract object
2. set the values of the variables `ndata` and `nmodl`
3. run the `alloc` procedure
4. overwite the procedures `objf`, `resf`, `gradf` and `dresf`

The first two steps can be done with a user-defined constructor and the last step need to be included as type-bound procedures. As long as these steps are satisfied, the user is free to add modify any number of variables or procedures for the inversion problem.

We created a concrete implmentation that minimizes a nonlinear objective function of the form:

$$J(\mathbf{m}) = \frac{1}{2}\|\mathbf{W_d}\left(\mathbf{F}(\mathbf{S}\mathbf{W_m}\mathbf{m}) - \mathbf{d}_{\text{obs}}\right)\|_2^2, \tag{1}$$

where $\mathbf{W_d}$ is a data weighting operator, $\mathbf{F}$ is a nonlinear forward operator, $\mathbf{S}$ is a preconditioning operator, $\mathbf{W_m}$ is a model weighting operator, $\mathbf{m}$ is the model and $\mathbf{d}_{\text{obs}}$ is the observed data. We made all the operators, except the foward modeling operator, optional by initializing them to an identity operator. For this problem, the object definition is:

```
type, extends(prob_obj) :: prob_nl_l2_diff
  private
  integer :: nsop
  real, dimension(:), pointer :: data
  real, dimension(:), allocatable :: temp_data, temp_modl, temp_prec
  procedure(oper), nopass, pointer :: fop, lop, wdop, wmop, sop
  procedure(update), nopass, pointer  :: uop
contains
  private
  procedure   :: objf, resf, gradf, dresf
end type
```

where `nsop` is the size of the preconditioned variable, `data` is the observed data, `temp_` are temporary variables, and the `op` pointers are procedure pointers to the operators. Notice that we added two operators: `lop` to contain the linearized operator required to calculate the gradient and `uop` to contain the operator that updates the model for the operators. We then defined an interface block to create a user-defined constructor and specify the interface of all the operators as:

```
interface prob_nl_l2_diff
  module procedure const
end interface
interface
  integer function oper(adj, add, modl, data)
    real,dimension(:) :: modl,data
    logical :: adj,add
  end function
  integer function update(modl)
    real,dimension(:) :: modl
  end function
end interface
```

where the interfaces follow the conventional SEPLib operators. The user-defined constructor satisfy the requirements we mentioned earlier as follow:

```
type(prob_nl_l2_diff) function const(fop, uop, lop, nmodl, ndata, data, &
                              wdop, wmop, sop, nsop)
```

```
    optional :: wdop, wmop, sop, nsop
    procedure(oper) :: fop, lop, wdop, wmop, sop
    procedure(update) :: uop
    integer :: nmodl, ndata, nsop, stat
    real, dimension(:), target :: data
    const%fop => fop
    const%uop => uop
    const%lop => lop
    const%nmodl = nmodl
    const%ndata = ndata
    const%data => data
    const%wdop => identity_op
    const%wmop => identity_op
    const%sop => identity_op
    const%nsop = nmodl
    if(present(wdop)) const%wdop => wdop
    if(present(wmop)) const%wmop => wmop
    if(present(sop)) const%sop => sop
    if(present(nsop)) const%nsop = nsop
    allocate(const%temp_data(const%ndata))
    allocate(const%temp_modl(const%nmodl))
    allocate(const%temp_prec(const%nsop))
    stat = const%alloc()
  end function
```

The only thing left is to calculate each variable using its function. The objective function is calculated using the residual in `objf` is:

```
  integer function objf(this, res, obj)
    class(prob_nl_l2_diff) :: this
    real, dimension(:) :: res
    double precision :: obj
    obj = 0.5*(sum(dprod(res,res)))
    objf = 0
  end function
```

The residual is calculate using the model in the `resf` function as:

```
  integer function resf(this, modl, res)
    class(prob_nl_l2_diff) :: this
    real, dimension(:) :: modl, res
    integer :: stat
    stat = this%wmop(F, F, modl, this%temp_modl)
    stat = this%sop(F, F, this%temp_modl, this%temp_prec)
    stat = this%fop(F, F, this%temp_prec, this%temp_data)
    this%temp_data = this%temp_data - this%data
    stat = this%wdop(F, F, this%temp_data, res)
    resf = 0
  end function
```

The gradient is calculated using the model and residual in the `gradf` function as:

```
  integer function gradf(this, modl, res, grad)
    class(prob_nl_l2_diff) :: this
    real, dimension(:) :: modl, res, grad
    integer :: stat
    stat = this%uop(modl)
    stat = this%wdop(T, F, this%temp_data, res)
    stat = this%lop(T, F, this%temp_prec, this%temp_data)
    stat = this%sop(T, F, this%temp_modl, this%temp_prec)
    stat = this%wmop(T, F, grad, this%temp_modl)
    gradf = 0
  end function
```

The change in residual is calculate using the model and model update in the `dresf` function as:

```
  integer function dresf(this, modl, dmodl, dres)
    class(prob_nl_l2_diff) :: this
    real, dimension(:) :: modl, dmodl, dres
    integer :: stat
    stat = this%uop(modl)
    stat = this%wmop(F, F, dmodl, this%temp_modl)
    stat = this%sop(F, F, this%temp_modl, this%temp_prec)
    stat = this%lop(F, F, this%temp_prec, this%temp_data)
    stat = this%wdop(F, F, this%temp_data, dres)
    dresf = 0
  end function
```

We also created an implementation for linear problems and another implementation that combines two problems by summing them.

stepper

The abstract stepper class has been extended to several more specific stepper classes, such as the backtracking or parabolic steppers. In general, a stepper class derived from the abstract class is done as follows:

```
  type, extends(stepper_obj) :: mystepper
  contains
    procedure :: run
  end type
```

Regardless of the stepper, it must override the run procedure that it inherits from the abstract class. This is done by first declaring the `run` function in the derived type as seen above, and later within the function definition the user writes the actual stepper algorithm. One example of this can be shown with the parabolic stepper class. In the following code listing, we have derived the `stepper_parab` type (specific stepper) from the `stepper_obj` type (abstract stepper) we omit the definition of `run` for brevity:

```
  type, extends(stepper_obj) :: stepper_parab
    integer :: c1 = 1.0, c2 = 2.0
    double precision :: ntry = 10, r = 0.25, alpha = 0.
    double precision :: alpha_min = 0.25, alpha_max = 4.0
    real, dimension(:), allocatable :: modl, res, dres
  contains
    procedure :: run
  end type
```

If the user then desires to implement their own stepper they can do so with a simple call (again using the parabolic stepper as the example):

```
type(stepper_parab) :: parab1
```

and can make changes to the members of the `parab1` object via the following calls:

```
parab1%ntry=1000
parab1%c1=-200.0
parab1%c2=200.0
```

We create one concrete line search class based on backtracking method. The method looks for a model update that satisfies Wolfe condition, as follows:

$$J(m_i + \alpha \Delta m_i) \leq J(m_i) + c_1 \alpha \mathbf{g_i}^T \nabla J(m_i) \quad (2)$$

$$|\mathbf{g_i}^T \nabla J(m_i) + \alpha \Delta m_i| \leq c_2 \mathbf{g_i}^T \nabla J(m_i), \quad (3)$$

where c_1 and c_2 need to be set. The backtracking method start from $\alpha = \alpha_{\max}$ and then $\alpha \times \rho \rightarrow \alpha$ at each step until the Wolfe condition is satisfied or we reached minmum allowed α_{min}.

Therefore we extend the line search method as:

```
  type, extends(stepper_obj) :: stepper_backtrack
    double precision :: decrease_coef = 0.01, curvatur_coef = 0.5
    double precision :: alpha = 0., alpha_min = 0.25, alpha_max = 4
    double precision :: rho = 0.5
    real, dimension(:), allocatable :: modl, res, dres, grad
  contains
    procedure :: run
  end type
```

where c_1 (**decrease_coef**), c_2 (**curvatur_coef**), ρ, α_{min} and α_{max} are specific to back tracking line search method. Those parameters has default value and can also be set by the experienced users.

We also created a few more line search implementations, such as the a sampling line search and a parabolic interpolation which is Brent's method. For Brent's method we have input a and b and try to find $a \leq \alpha b$ that minimize the objective function. The algorithm keep shrinking the interval $[a, b]$ until $J(m + a\Delta m) - J(m + b\Delta m)$ or $|a - b|$ is below a certain threshold.

terminator

The terminator object determines when to stop the inversion. This is done by the procedure `test` which returns a logical value on whether the inversion should be terminated. We created a concrete implementation that gives the user the option to stop using one of the following criteria: number of iterations, number of evaluations, number of hours passed, size of residual norm and size of gradient norm. The object definition is:

```
type, extends(terminate_obj) :: terminate_simple
  integer :: niter = 0, maxfevals = 0
  real :: maxhours = 0.
  double precision :: tolr = 1.0d-20
  double precision :: tolg = 1.0d-20
  integer(kind=8), private :: timer0 = 0, rate = 0
contains
  procedure :: test
end type
```

Because we need to keep track of time, we created a user-defined constructor that initializes the time passed as follows:

```
interface terminate_simple
  module procedure const
end interface

type(terminate_simple) function const(niter, maxfevals, maxhours, tolr, tolg)
  optional :: niter, maxfevals, maxhours, tolr, tolg
  integer :: niter
  integer :: maxfevals
  real :: maxhours
  double precision :: tolr
  double precision :: tolg
  if(present(niter)) const%niter = niter
  if(present(maxfevals)) const%maxfevals = maxfevals
  if(present(maxhours)) const%maxhours = maxhours
  if(present(tolr)) const%tolr = tolr
  if(present(tolg)) const%tolg = tolg
```

```
    call system_clock(const%timer0, const%rate)
    const%rate = const%rate * 3600.
  end function
```

The implementation of the `test` procedure checks if the user provided any stopping criteria and then test for that criteria accordingly as follows:

```
  logical function test(this, iter, prob)
    class(terminate_simple) :: this
    integer :: iter
    class(prob_obj) :: prob
    real :: hours_passed
    integer(kind=8) :: timer1
    test = .false.
    if(this%niter > 0 .and. iter > this%niter) then
      test = .true.
      write(0,*) "Terminate: maximum number of iterations reached"
    end if
    if(this%maxfevals > 0 .and. prob%get_fevals() >= this%maxfevals) then
      test = .true.
      write(0,*) "Terminate: maximum number of evaluations"
    end if
    if(this%maxhours > 0.) then
      call system_clock(timer1)
      hours_passed = 1.*(timer1-this%timer0)/this%rate
      if(hours_passed >= this%maxhours) then
        test = .true.
        write(0,*) "Terminate: maximum number hours reached", &
                   hours_passed, this%maxhours
      end if
    end if
    if(prob%get_rnorm() < this%tolr) then
      test = .true.
      write(0,*) "Terminate: minimum residual tolerance reached", &
                 real(prob%get_rnorm()), real(this%tolr)
    end if
    if(prob%get_gnorm() < this%tolg) then
      test = .true.
      write(0,*) "Terminate: minimum gradient tolerance reached", &
                 real(prob%get_gnorm()), real(this%tolg)
    end if
  end function
```

solver

When implementiong the solver object, the user needs to do the following in a loop:

1. run the problem object procedure `output`
2. calculate the search direction
3. run the stepper by the procedure `run` and test its success
4. check the terminator procedure `test` to determine when to stop the loop

If the solver doesn't require a line search, then the user need to implement the stepper procedure `clip_modl` every time the model is modified.

To show how a user can define an actual solver object, we describe the extension of the abstract solver to a non-linear conjugate-gradient solver object. First, we need to declare the extension of the abstract solver object as follows:

```
type, extends(solver_obj) :: solver_nlcg
  character(20) :: beta_type='FR'
  contains
  procedure :: run
end type
```

where the string `beta_type` defines how the step length is computed at a given iteration with the previous search direction and the current gradient. We see that in the children object `solver_nlcg` contains a `run` function as the abstract solver object. In this actual object we cannot define just a simple abstract interface. In fact, we need to implement the solver function keeping the same interface of the abstract interface (i.e., `integer function run(this, modl, prob, stpr, term)`). In this function we declare all the temporary variables that we need to run the solver. In our case this interface is written as follows:

```
class(solver_nlcg) :: this
integer :: stat
real, dimension(:) :: modl
class(prob_obj) :: prob
class(stepper_obj) :: stpr
class(terminate_obj) :: term
double precision :: beta, alpha, f
integer :: iter
logical :: success
real, dimension(:), allocatable :: grad, grad0, dmodl
```

where `alpha` is the step size, `beta` is the conjugate gradient ration between current and previous search directions, `f` is the objective function value, `grad` is the gradient, `grad0` is the previous gradient and `dmodl` is the search direction. The inversion loop is very simple and it is implemented as:

```
beta = 0.
iter = 1
do
  grad0 = grad
```

```
    stat = prob%get_grad(modl, grad)
    stat = prob%output(modl)
    if(iter > 1) beta = betaf(grad, grad0, dmodl)
    if(beta < 0.) beta = 0.
    dmodl = beta*dmodl - grad
    stat = stpr%run(modl, prob, dmodl, alpha, success)
    if(.not. success) then
      write(0,*) "Stepper couldn't find a proper step size"
      exit
    end if
    stat = prob%get_obj(modl, f)
    write(0,*) "iter", iter, " obj =", real(f), " feval =", prob%get_fevals()
    iter = iter + 1
    if(term%test(iter, prob)) exit
  end do
```

First, we start the loop by placing the previous gradient into the `grad0` variable. Then, we retrieve or compute the current gradient calling the problem object's function `get_grad` and output the previous model calling the `prob%output` function. After these steps, we compute the conjugate search direction with the `betaf` function. At the first iteration the search direction is equal to the current gradient that means stepping with the steepest descent direction. Having computed the search direction we can now call the stepper and find the step length with the `stat = stpr%run(modl, prob, dmodl, alpha, success)` line. From the result of the stepper contained in the `success` variable, we can decide whether a better model has been found or not. The last steps are to print some useful information to screen and test if the termination criteria has been achieved. The beta function is implemented separately from the solver `run` function (see the file `solver_nlcg.f90` in the `Src` directory). The reader interested in the mathematical details of non-linear conjugate gradient can find them in Dai and Yuan (1999).

We also created an implementation for the L-BFGS solver and the linear conjugate-gradient solver.

driver

We created a driver to simplify using a library, particularly for users that only want to use one of the concrete implementations in the library. This driver allows the user to either provide objects for each of the solver, stepper and terminator or just specify the number of each object. The driver has one function which has the following interface:

```
integer function nlsolver(modl, prob, nsolver, nstepper, niter, &
                          solver, stepper, term)
  optional :: nsolver, nstepper, niter, solver, stepper, term
```

The variable `modl` is the model, `prob` is the problem definition object, `nsolver` is the solver number, `nstepper` is the stepper number, `niter` is the number of iterations, `solver` is the solver object, `stepper` is the stepper object and `term` is the terminator object. If both the number and the object are provided, the driver will prioritize the object using a local pointer. If neither is provided, the driver will pick the default options which are the nonlinear conjugate-gradient solver and back-tracking stepper. The following is an example of picking the solver:

```
if(present(solver)) then
  solverl => solver
else
  nsolverl = NLCG
  if(present(nsolver)) nsolverl = nsolver
  if(nsolverl .eq. NLCG) then
    solver1 = solver_nlcg()
    solverl => solver1
  else if(nsolverl .eq. BFGS) then
    solver2 = solver_lbfgs()
    solverl => solver2
  else if(nsolverl .eq. LCG) then
    solver3 = solver_lcg()
    solverl => solver3
  end if
end if
```

After setting the solver, stepper and terminator, the driver will run the solver as:

```
stat = solverl%run(modl, prob, stepperl, terml)
```

To execute the driver, all we need is to run the following line:

```
stat = nlsolver(modl, prob1, NLCG, PARAB, niter=10)
```

EXAMPLES

Example 1

The first example we demonstrate is a linear missing data problem from chapter three of Geophysical Image Estimation by Example (GEE). The problem boils down to the following: given the filter in Figure 2 and the data in Figure 1, how can we “ensure that the restored data, after specified filtering, has minimum energy”? Now with the problem in mind, we form the following fitting goal:

$$\mathbf{0} \approx \mathbf{FJm} + \mathbf{Fm}_{\text{known}}$$

where $\mathbf{F}$ is the transient convoultion operator, $\mathbf{J}$ is a masking operator, $\mathbf{m}$ is a vector that contains all of the points shown in Figure 1 and $\mathbf{m}_{\text{known}}$ is a vector containing only the known data points. For a full derivation of this fitting goal, please refer to GEE page 72.

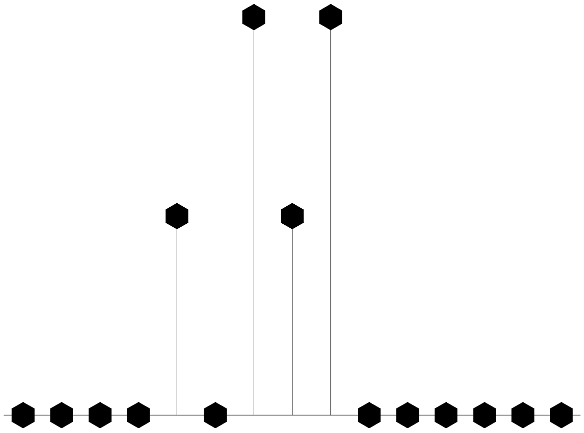


Figure 1: The data which we are attempting to restore. [**ER**] ali2/. datamiss

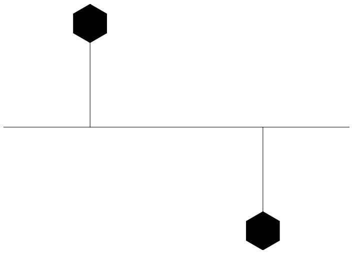


Figure 2: The filter which will be used to restore the data. [**ER**] ali2/. mlines

While this problem can be readily solved with a linear optimization scheme, for the purpose of this report, we solve this problem with a non-linear conjugate gradient solver and a parabolic fitting stepper. Also, we use the transient convolution and mask operators defined within GEELIB and supply these operators to our solver. To solve this problem with our linear solver, we will need two files. One of the files will be our main program that will do the I/O and call the solver as well as other necessary routines. The second file serve to define our problem object. In total, in order to solve this problem, the user would be required to write less than 30 lines of code. Here we go describe the code in detail:

Main program

Without the I/O and usual operator initialization, the call to perform the nonlinear optimization is shown in the following lines of code:

```
type(prob_datafill) :: prob
prob = prob_datafill(nmodl=nmodl,ndata=ndata,na=nain,data=datain)
stat = prob%alloc()
stat = nlsolver(modl, prob, NLCG, PARAB, niter=50)
```

In the first line, we create an instance of our problem object which we need to define in another module file (explained later). Then we can call our `prob` object's constructor to assign it the necessary variables (the arguments passed to the constructor). Then, with the call `prob%alloc()`, we allocate our model, residual, gradient, dmodel, and dresidual arrays that will be used in our non-linear conjugate gradient solver. With the last line, we call the non-linear solver passing it the `modl` array, the user-defined `prob` object, the type of solver, the type of stepper and finally the number of iterations. It is important to note that the only two required arguments for this `nlsolver` are the `modl` and `prob` arguments. The remainder can be left out and will be left out and will be handled by the solver. Also, while it was not done in this example, we can create stepper and solver objects that will allow us to to override default parameters set in the solvers and steppers.

In the second file, we must define the `prob_datafill` class as was shown previously in the problem section of this report. This means defining the previously mentioned `objfunc`, `resfunc`, `gradfunc`, and `dresfunc` functions, as well as a user-defined constructor that allows the user to pass additional values to the solver.

Once we have defined the main program as well as our `prob_datafill` class, we can implement our solver which prints the following at the command line:

```
Bin/Data_fill.x <  ./Dat/datamiss.H filtin=./Dat/mlines.H > fill_lines.H
 iter           1  obj =  0.6250000      feval =          11
 iter           2  obj =  0.5392157      feval =          16
 iter           3  obj =  0.4687500      feval =          21
 iter           4  obj =  0.4417293      feval =          26
 iter           5  obj =  0.4227941      feval =          31
 iter           6  obj =  0.4129246      feval =          36
 iter           7  obj =  0.4102488      feval =          41
 iter           8  obj =  0.4096886      feval =          46
 iter           9  obj =  0.4089912      feval =          51
 iter          10  obj =  0.4089340      feval =          56
 iter          11  obj =  0.4089286      feval =          64
 iter          12  obj =  0.4089285      feval =          72
 Stepper couldn't find a proper step size, will terminate solver
```

where `iter` is the iteration number, `obj` is the value of the objective function, and `feval` is the number of function evaluations.

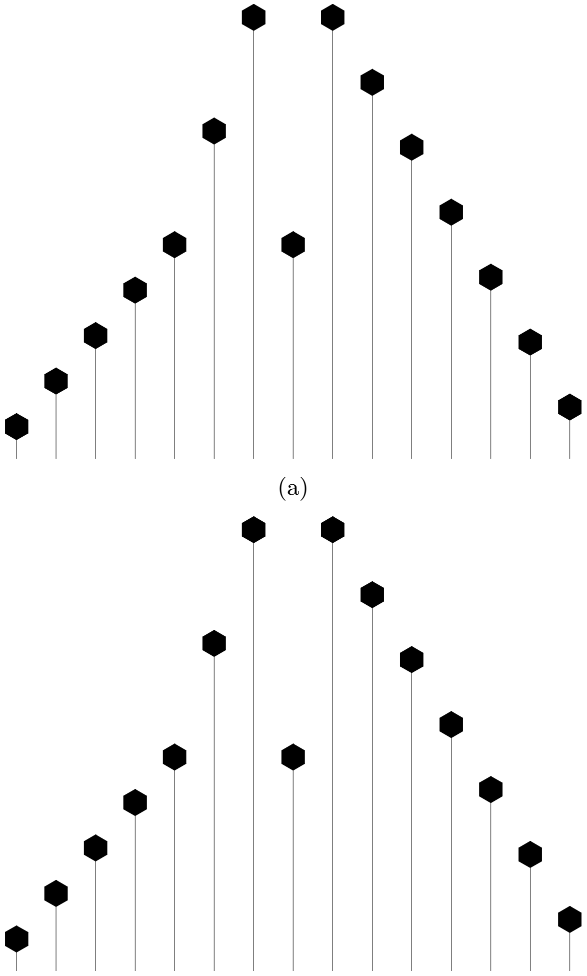
In addition to diagnostic messages shown above, it gives us the result shown in Figure 3(a). In comparing this result to Jon's original result, we can observe that they are nearly equivalent.

Example 2

Rosenbrock function

To test our nonlinear solver we implement the well-known Rosenbrock function. The multidimensional Rosenbrock function is expressed as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{N-1} \left[(1-x_i)^2 + 100(x_{i+1} - x_i^2)^2\right], \tag{4}$$



(a)

(b)

Figure 3: (a) Missing data fill with the non-linear solver and (b) missing data fill from Claerbout's GEE. [**ER**] ali2/. fill-lines,jon-fill-lines

where f is the objective function and $\mathbf{x}$ is the optimization parameter. In 2D this function reduces to:

$$f(x,y) = (1-x)^2 + 100(y-x)^2, \tag{5}$$

and is shown in Figure 4. The 2D Rosenbrock function is know to have a single global minimum at $x = 1$ and $y = 1$ inside a broad minimum valley. When we start the optimization from $x = -1$ and $y = -1$ our nonlinear conjugate gradient algorithm converges in 92 iteration to $x = 1.000112$ and $y = 1.000224$. The optimization path is displayed in Figure 4 by the cyan line and the tested points are indicated by the purple stars. As shown by this path the algorithm has found the correct minimum and stopped close it. The source code for running the example is in the `Src` directory and named `Rosenbrock_test.f90`. This program handles multidimensional Rosenbrock functions.

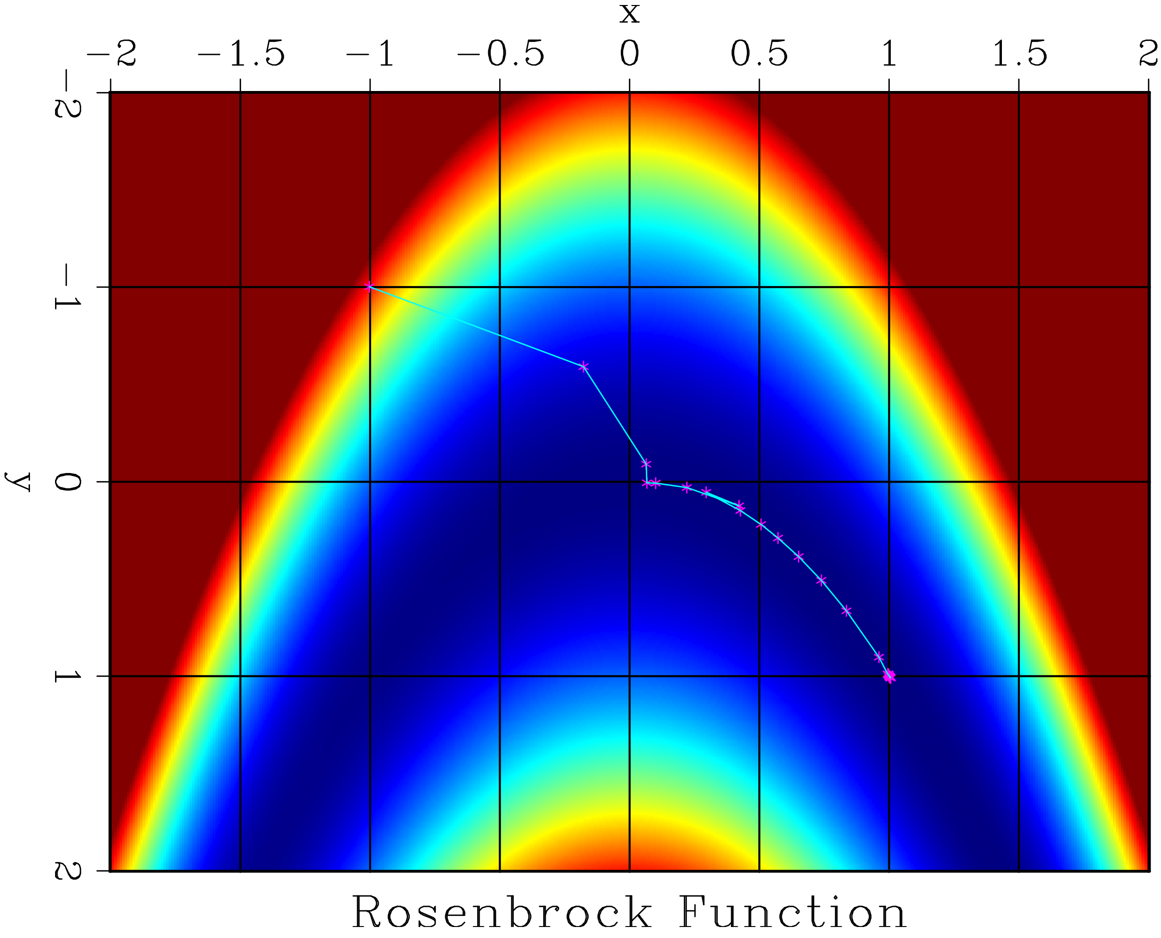


Figure 4: Bidimensional Rosenbrock function. Red and blue colors indicate high and low function values respectively. The cyan line with the purple stars is the optimization path followed when the starting solution is the point $x = -1$ and $y = -1$. [**ER**] ali2/. Rosenbrock-figure

Example 3

Deghosting

Deghosting, together with debubbling, is one of the fundamental preprocessing techniques necessary for dealing with marine datasets. If we know the filter that generates the ghost frequency notch in the data is easy to suppress its effect. One of the biggest challenge is actually to estimate the filter coefficient for the ghost. The ghost effect for a single trace and vertical propagation can be expressed in the frequency domain as:

$$d(\omega) = m(\omega)(1 - Re^{\omega\tau}), \tag{6}$$

where $d(\omega)$ represents the ghosted trace, $m(\omega)$ is unghosted data, R is the reflection coefficient at the water surface, and τ is the two-way traveltime from the source to the water surface. Knowing the R and τ in this simple 1D case is easy to eliminate the ghost effect by deconvolving $d(\omega)$ with the filter $(1 - Re^{\omega\tau})$. However, estimating these two parameters is not a simple task in the real work. In fact, the ghost, in this case the source ghost, can be hidden by the bubble, and other physical phenomena. In addition, the water surface reflection coefficient can be frequency dependent. In this simple example where are assuming that R is independent of the frequency.

In order to estimate the two physical parameters of the ghost filter we need to decide an objective function to minimize. A good target to reach is minimizing the energy of the deghosted trace, since the unghosted data should contain less energy than the ghosted one. This objective function can be written as follows:

$$E(R, \tau) = \int \frac{d^* d(\omega)}{(1 - 2cos(\omega\tau) + R^2)} d\omega, \tag{7}$$

where E is the energy of the deghosted data, and $d^* d(\omega)$ is the amplitude spectrum of the ghosted trace. Knowing the analytical expression of the objective function (equation 7) is trivial to compute the partial derivative with respect each parameter and perform a gradient-based nonlinear inversion. We test the nonlinear solver library on two synthetic cases and one real case.

The left panel of Figure 5 shows a synthetic Gaussian impulse with $15Hz$ as central frequency, we consider this wavelet our unghosted trace. The right panel of the same figure displays this wavelet after applying a ghosting filter with $R = 0.6$ and $\tau = 50ms$. Since we know analytical expression of the objective function, we can compute it for certain range of values of the two parameters. Figure 7 shows the objective function computed for different value of the reflection coefficient and ghosting lag, it exhibits a clear global minimum, particularly elongated along the R-axis. The shape of the global minimum reflects the well-known fact that the energy of deghosted data is less sensitive to the water surface reflection coefficient. This observation is also shown by the superimposed optimization path that does not change in R as we progress toward the minimum when we start with $R = 0.5$ and $\tau = 70ms$. On the left panel of Figure 7 is shown a section of the energy fixing τ to the correct value. We see that close to the true reflection coefficient the gradient of the function is close to zero. In fact, the solver does not change the R model from the starting point. On the other hand, since the gradient is higher for the τ parameter, the solver can find a solution close to the correct value. In the right panel of Figure 7 is displayed the deghosted data using inverted parameter ($R = 0.499$ and $\tau = 51.845ms$). Most of the ghost energy has been canceled; however, since the R value is not correct we are introducing undesired ringing effects.

In the second synthetic case we use a different wavelet than before (left panel on Figure 8). In this case the wavelet is composed of two impulses of the same amplitude. We apply the ghost filter used in the previous case and we obtain the trace shown in the right panel of Figure 8. In this case we have obtain a wavelet similar to a Ricker. We can again compute the objective function for different values of the filter (Figure 9). In this case we notice that we have a more complex valley. This shape is caused by the presence of the second impulse in the original wavelet that is considered as a ghost effect. In this case if we start the solver from $R = 0.4$ $\tau = 55ms$ optimizes both model parameters. For this example because the of the presence of the second impulse, the minimum is centered on higher value of R respect to the correct solution (left panel of Figure 10). Deconvolving the ghosted trace with the optimized filter parameter, we eliminate most of the ghost energy but we obtain ringing effects as before.

The last test is performed on a real trace recorded in the Seneka lake from airgun testing (Figure 11). Looking at the energy plot as function of R and τ (Figure 12), we can see multiple minima for short delay time. These minima are cause by the presence of the bubble effect in the trace. However, we can easily estimate the correct delay thanks to presence of a precursor and its ghost at approximately $60ms$ and $92ms$ respectively. In fact, if we start from a starting parameters equal to $R = 0.4$ and $tau = 33ms$ the solver find the minimum of the valley where we start the optimization. However, the presence of the bubble influences this minimum and does not allow us to find the optimal deghosting filter (see right panel of Figure 13).

ACKNOWLEDGEMENT

The authors would like to thank Dolphin Geophysical and Chelminski associates for the permission to show the lake data.

REFERENCES

Dai, Y.-H. and Y. Yuan, 1999, A nonlinear conjugate gradient method with a strong global convergence property: SIAM Journal on Optimization, **10**, 177–182.

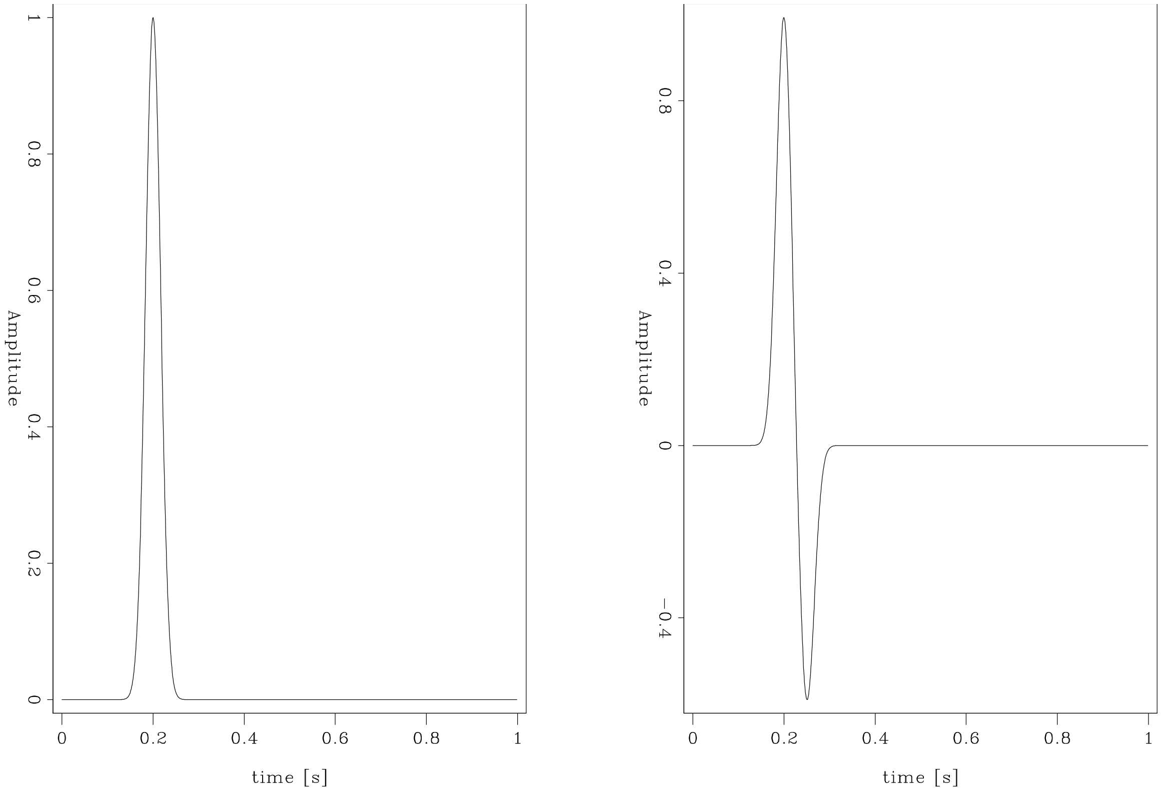


Figure 5: (left) Original unghosted trace. (right) Ghosted trace with ghosting parameters equal to $R = 0.6$ and $\tau = 50ms$. [**ER**] ali2/. problem-1

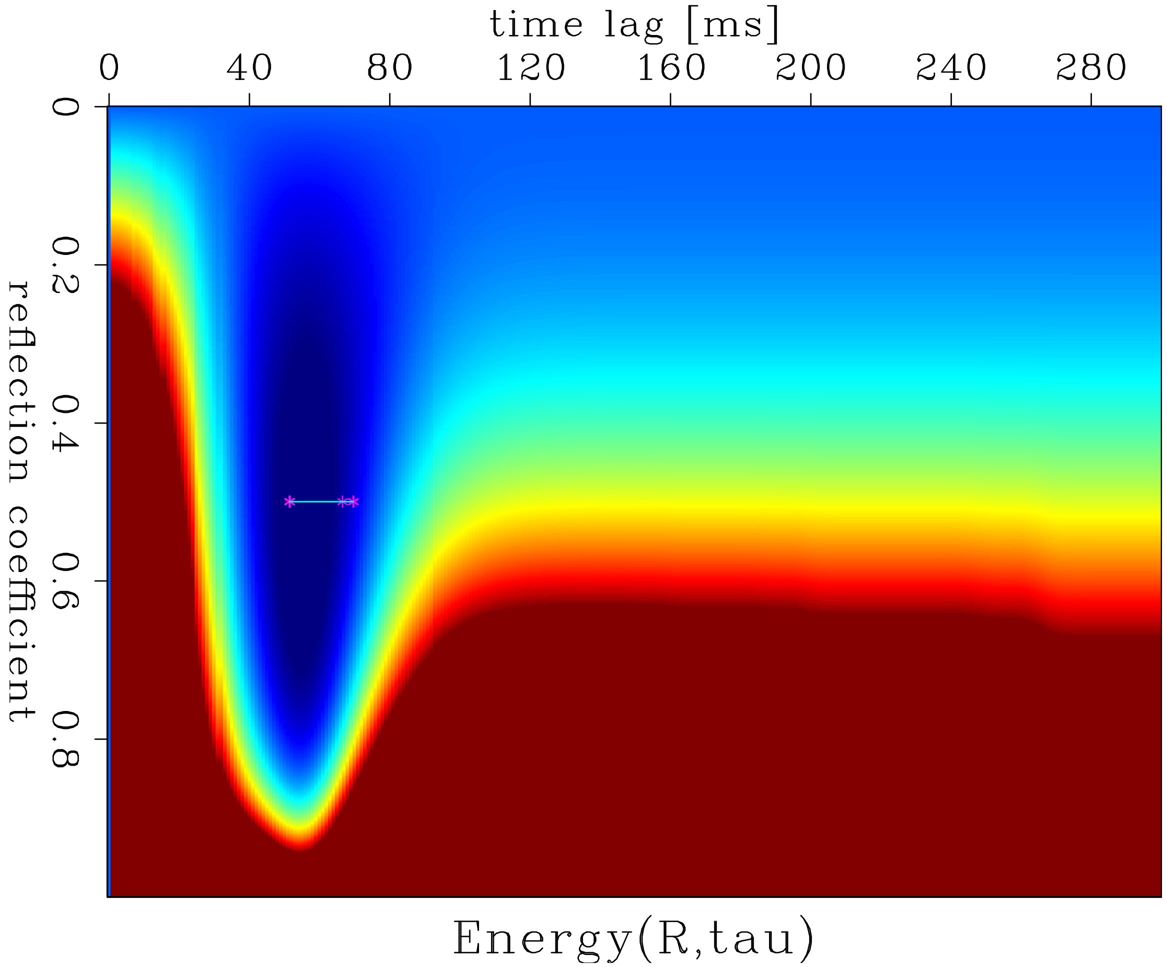


Figure 6: Energy of the ghosted trace of right panel Figure 5 as function of R and τ. Overlaid is the optimization path when the starting solution is $R = 0.5$ and $\tau = 70ms$. [**ER**] ali2/. ghost-prob-1

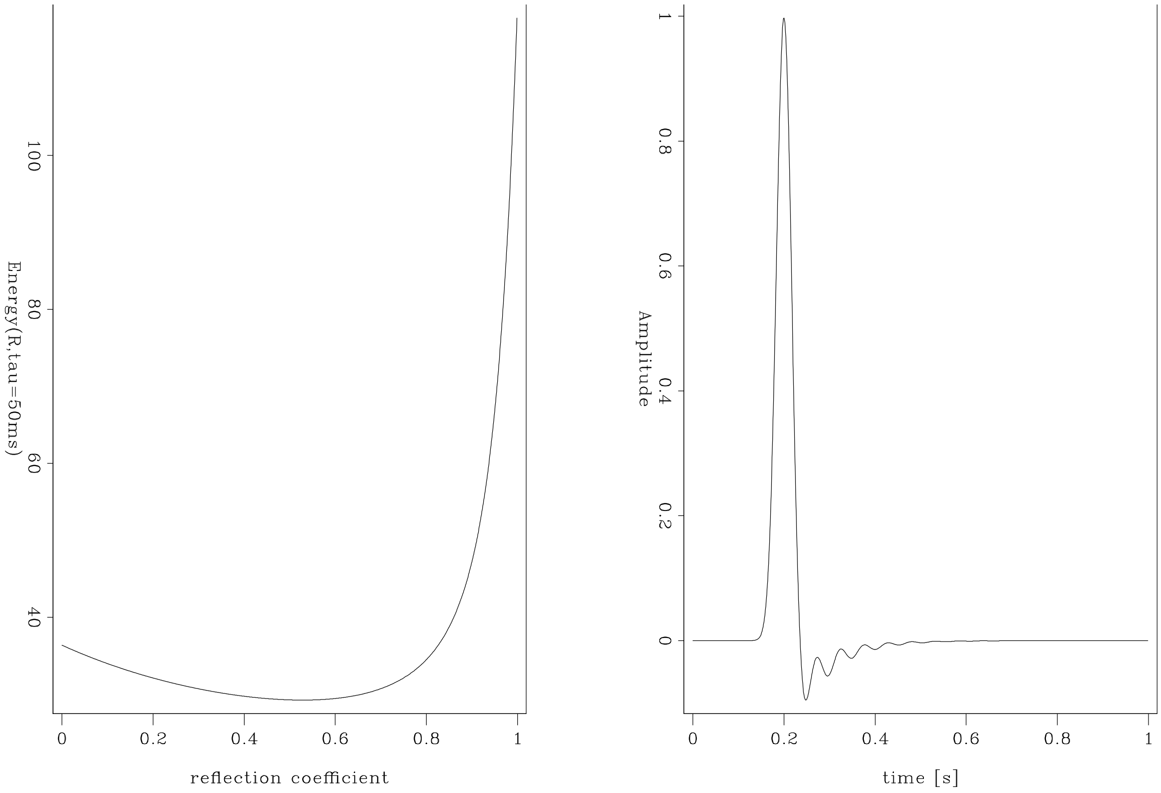


Figure 7: (left) Section of the objective function of left panel of Figure 6 for $\tau = 50ms$. (right) Deghosted trace using the optimized parameters from the solver ($R = 0.499$ and $\tau = 51.845ms$). [**ER**] ali2/. solution-1

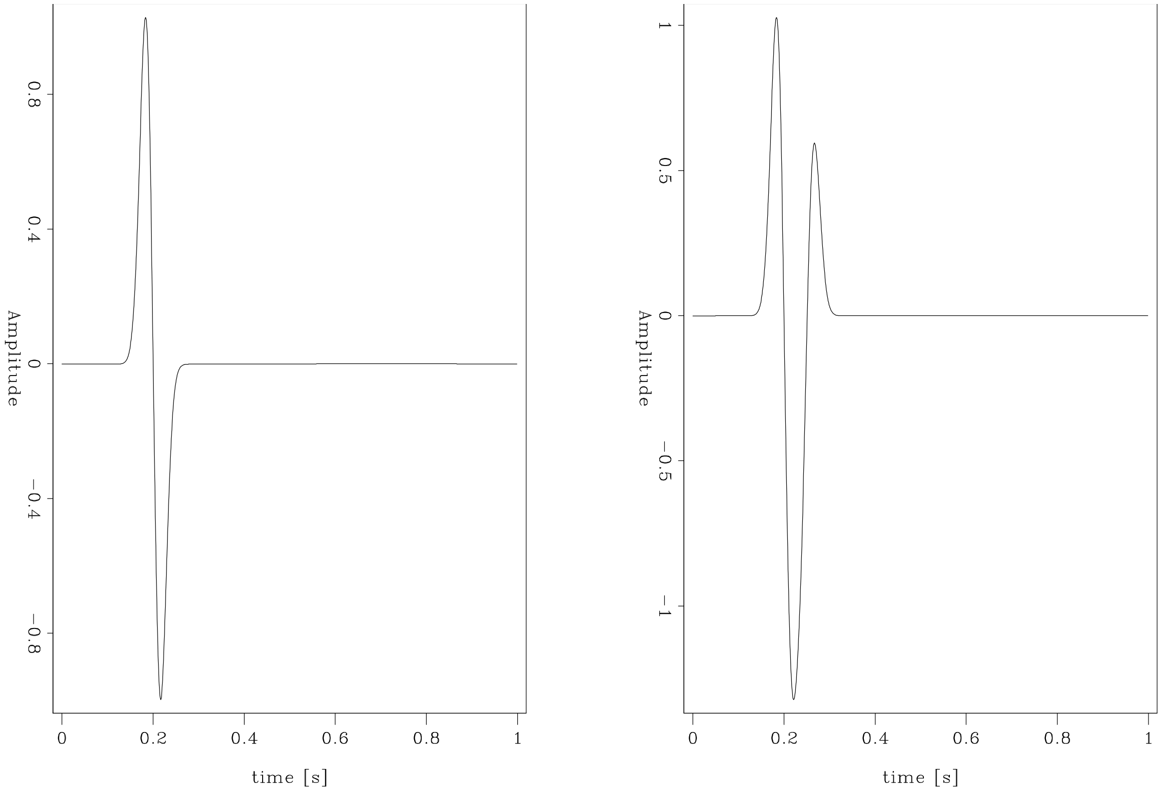


Figure 8: (left) Original unghosted trace for the second synthetic case. (right) Ghosted trace with ghosting parameters equal to $R = 0.6$ and $\tau = 50ms$. [**ER**] ali2/. problem-2

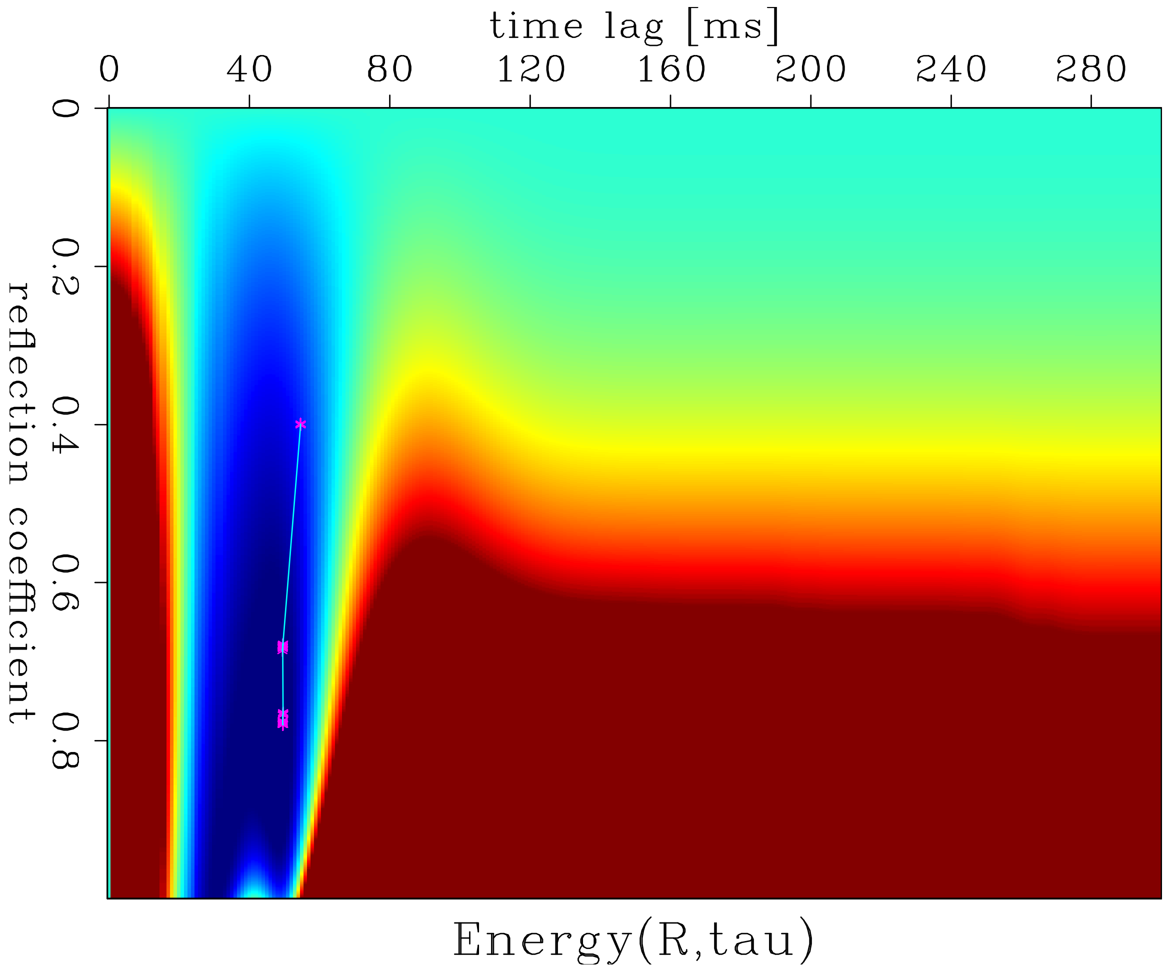


Figure 9: Same plot of Figure 6 but for the ghosted trace on the right panel of Figure 8. [**ER**] ali2/. ghost-prob-2

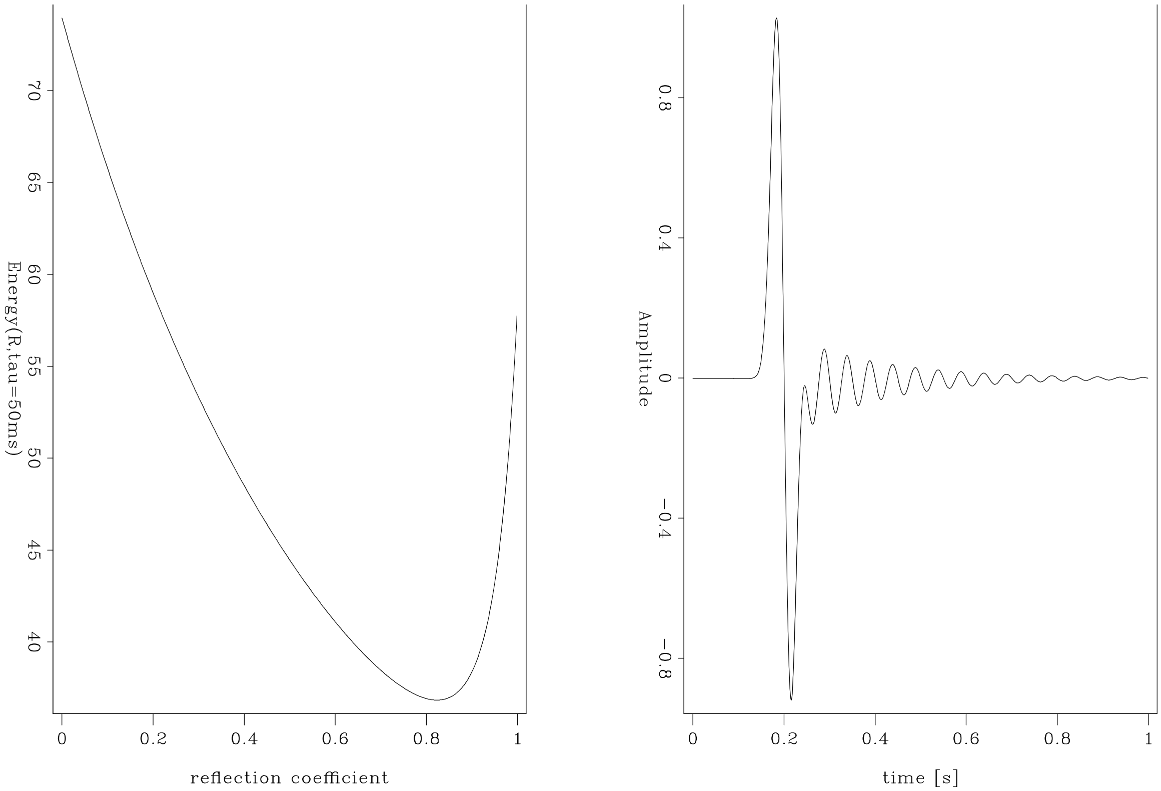


Figure 10: Section of the objective function of left panel of Figure 9 for $\tau = 50ms$. (right) Deghosted trace using the optimized parameters from the solver ($R = 0.78$ and $\tau = 49.95ms$). [**ER**] ali2/. solution-2

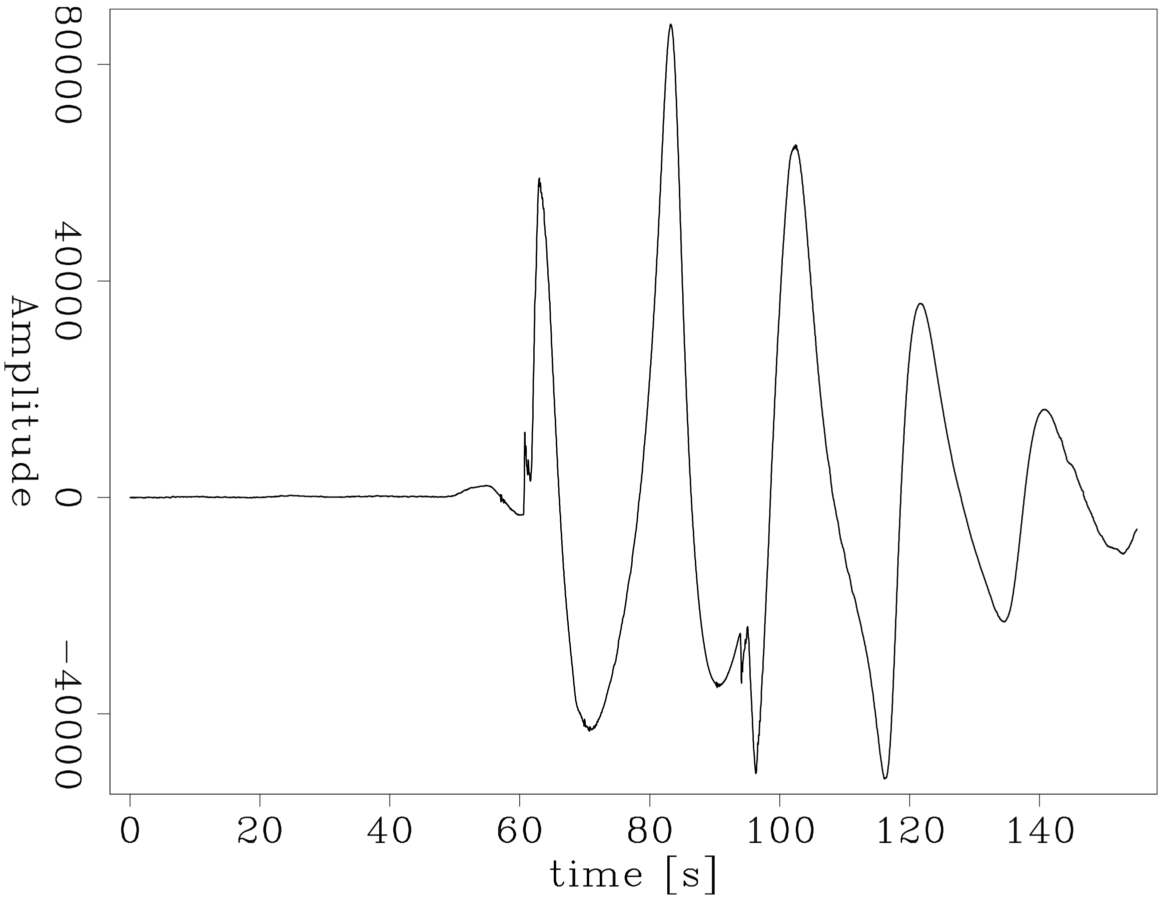


Figure 11: Real trace recorded during a airgun testing. The ghost delay is easily detectable and is approximately $33ms$. [**ER**] ali2/. wavelet-3

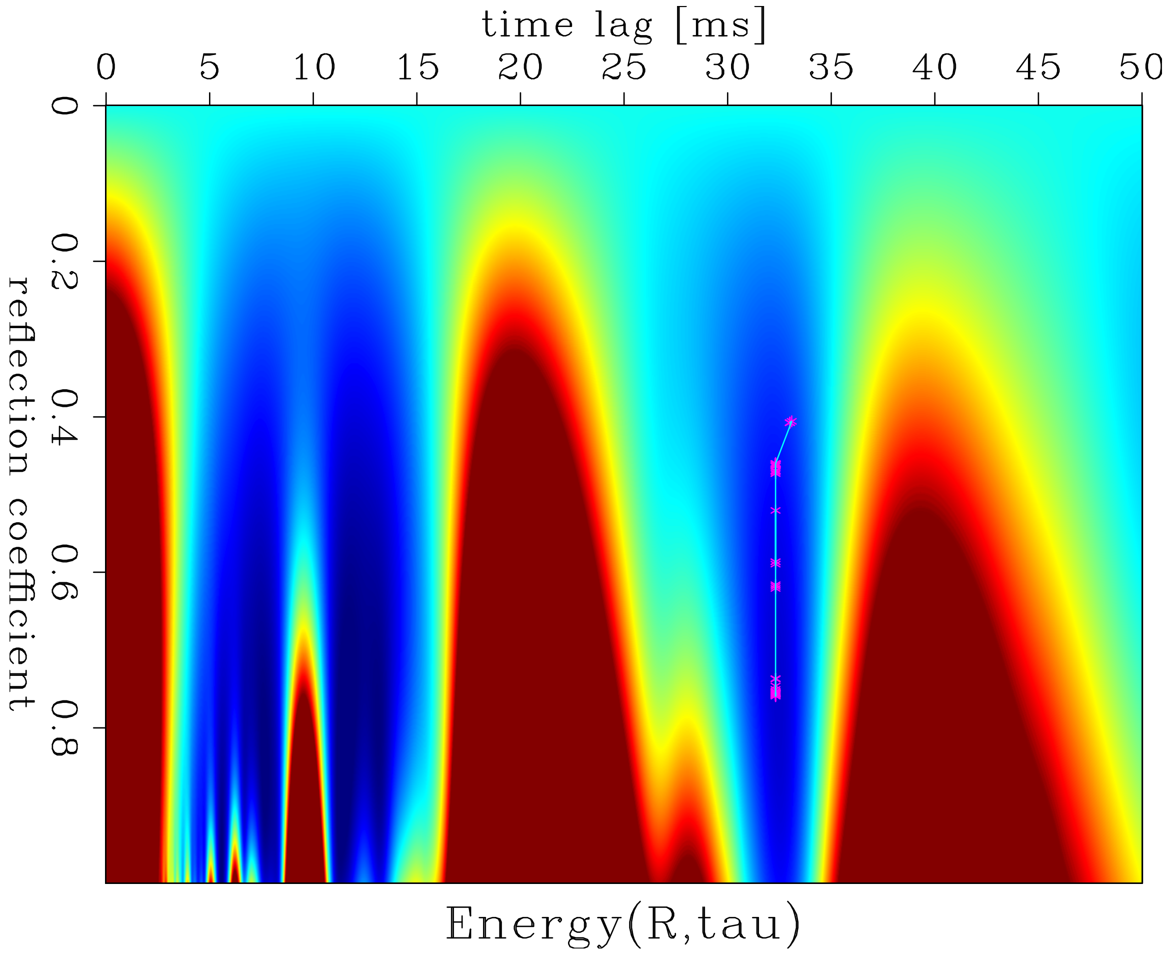


Figure 12: Energy plot and optimization path for the real trace test. [**ER**] ali2/. ghost-prob-real

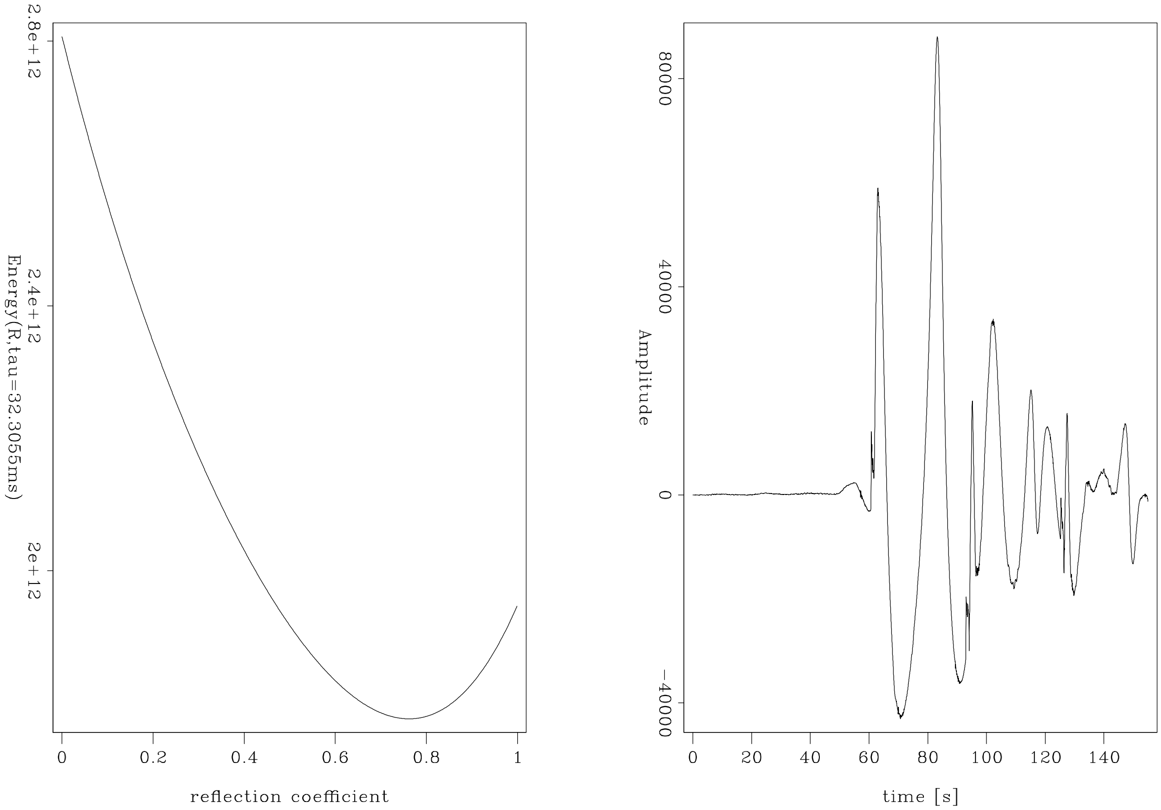


Figure 13: Section of the objective function of left panel of Figure 12 for $\tau = 32.3055ms$. (right) Deghosted trace using the optimized parameters from the solver ($R = 0.7564650$ and $\tau = 32.30554ms$). [**ER**] ali2/. solution-real

Simultaneous inversion of velocity and Q using wave-equation migration analysis

Yi Shen

ABSTRACT

I develop a method for simultaneous inversion of velocity and Q models. This method poses the simultaneous estimation problem as an optimization problem that seeks optimum velocity and Q models by minimizing user-defined image residuals. Numerical tests on a modified SEAM model with two gas clouds demonstrates the necessity of using such simultaneous inversion, when the existent velocity and Q models are inaccurate. The results show that this simultaneous inversion method is able to retrieve both velocity and Q models, as well as correct and compensate the distorted migrated image caused by inaccurate velocities and Q models.

INTRODUCTION

Because strong attenuation and low-velocity anomalies are present in gas pockets or clouds, they present notoriously challenging problems for reservoir identification and interpretation (Billette and Brandsberg-Dahl, 2005). Attenuation degrades the seismic image quality by damping the image amplitude, lowering the image resolution, distorting the phase of events, and dispersing the velocity. A wrong velocity estimation for the low-velocity anomalies also results in imaging problems, such as mis-positioning of events and discontinuity of the imaged structures. These problems impede accurate image interpretation for hydrocarbon production and well positioning. To mitigate the effects of such gas accumulations on the image and improve imaging of the subsurface, it is important to understand the properties of these gas pockets or clouds. Compressional velocity (V) and compressional quality factor (Q) play an important role in correcting and compensating for the gas-induced distortion in the image.

Shen et al. (2013, 2014) developed a method, wave-equation migration Q analysis (WEMQA), to produce a reliable Q model. This method analyzes attenuation effects from the image space, and uses wave-equation Q tomography to estimate Q models. However, this method requires highly accurate velocity models. An inaccurate velocity model used by WEMQA easily distorts the spectra of the migrated events and causes errors in spectral analysis for estimating the attenuation effects. Therefore, it is necessary to invert for both velocity and Q models if neither of these models is correct. Thus, such an inversion compensates for the errors in Q estimation caused by inaccurate velocities.

In this paper, I initially develop a method for simultaneous inversion of velocity and Q models based on the previous workflow of WEMQA (Shen et al., 2013, 2014). Then I test this method on a synthetic dataset to demonstrate its benefit and effectiveness.

THEORY

I pose the simultaneous estimation problem as an optimization problem that seeks optimum velocity and Q models by minimizing user-defined image residuals:

$$J = J_v(v,Q) + \beta J_Q(v,Q), \tag{1}$$

where β is a weighting parameter that balances two user-defined image residuals $J_v(v,Q)$ and $J_Q(v,Q)$, and can be changed through iterations. The image residuals $J_v(v,Q)$ and $J_Q(v,Q)$ are functions of the current velocity and Q models. However, $J_v(v,Q)$ emphasizes more on the kinematic changes in an image caused by an inaccurate velocity model, while $J_Q(v,Q)$ emphasizes more on the amplitude spectral change in an image caused by an inaccurate Q model.

I use the normalized differential semblance optimization (DSO) (Tang, 2011) as the criterion to mainly estimate the velocity. This objective function normalizes the square of the root-mean-squared (RMS) image amplitudes to reduce the influence of image amplitude variations caused by attenuation and uneven illumination. The normalized DSO objective function is in the subsurface-offset $\mathbf{h}$ domain:

$$J_v = \frac{1}{2}\sum_{\mathbf{x}} \frac{\sum_{\mathbf{h}} |\mathbf{h}|^2 \, |m(\mathbf{x},\mathbf{h})|^2}{\sum_{\mathbf{h}} |m(\mathbf{x},\mathbf{h})|^2}, \tag{2}$$

where $m(\mathbf{x},\mathbf{h})$ is the migrated image with the current velocity and Q models in the subsurface-offset domain. The physical interpretation of the subsurface-offset-domain DSO is that it optimizes the models by penalizing energy at non-zero subsurface offset, taking advantage of the fact that seismic events should focus at zero-subsurface offset if migrated using accurate models.

By definition for J_Q, the image residual primarily coming from attenuation is the difference between the background image computed with the current background models and the attenuation-free image. In fact, instead of computing the difference between these two images, I calculate the spectral change of the images:

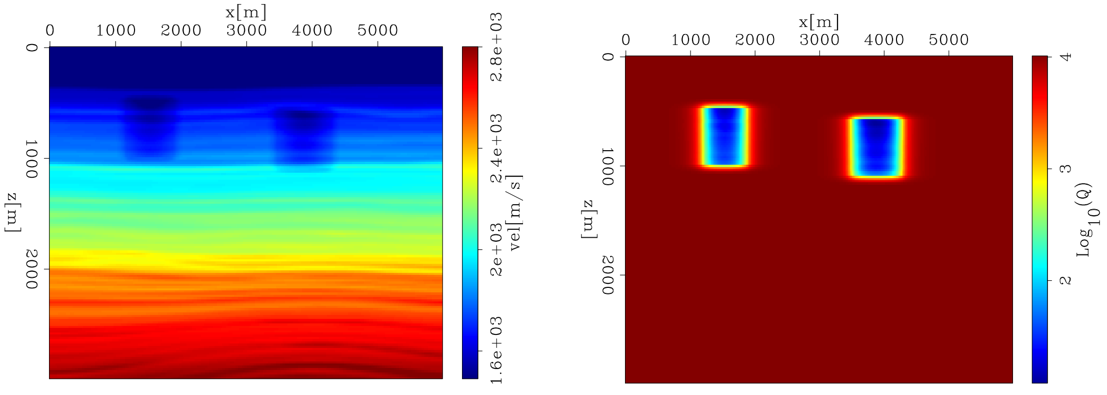
$$J_Q = \sum_{\mathbf{x}} |\rho(\mathbf{x})|^2. \tag{3}$$

The change in the spectrum can be indicated by the steepness of the slope $\rho(\mathbf{x})$ computed by the spectral ratio method (Tonn, 1991), between a number of selected, windowed events in the background image and those in the reference windows. These reference windows are carefully selected from the background image to not be contaminated by attenuation. All the windows in this method are large and have the same size; therefore, the influence of the interfering reflectivities on the spectra are statistically the same over all windows. Based on the assumption that the amplitude spectra have the same frequency content over the windows if the models used for imaging are accurate, this method minimizes the spectral differences between the selected windows and the reference windows.

These user-defined residuals are mapped to the perturbations in the current velocity and Q models by the wave-equation velocity and Q tomography operators (Tang, 2011; Shen et al., 2013, 2014). I use the mapped perturbation as gradient directions to conduct a line-search in optimization schemes, to obtain both velocity and Q models.

NUMERICAL EXAMPLE

To demonstrate this methodology, I use a portion of the SEAM synthetic velocity, adding two gas clouds with lower velocity than the surrounding sediments, as shown in Figure 1(a). The Q model (in logarithmic scale) shown in Figure 1(b) also includes these two gas clouds with high attenuation. I generate a 2D synthetic data with 56 shots with 100 m spacing, 137 receivers with 40 m spacing, and a Ricker source wavelet with 12 Hz central frequency.



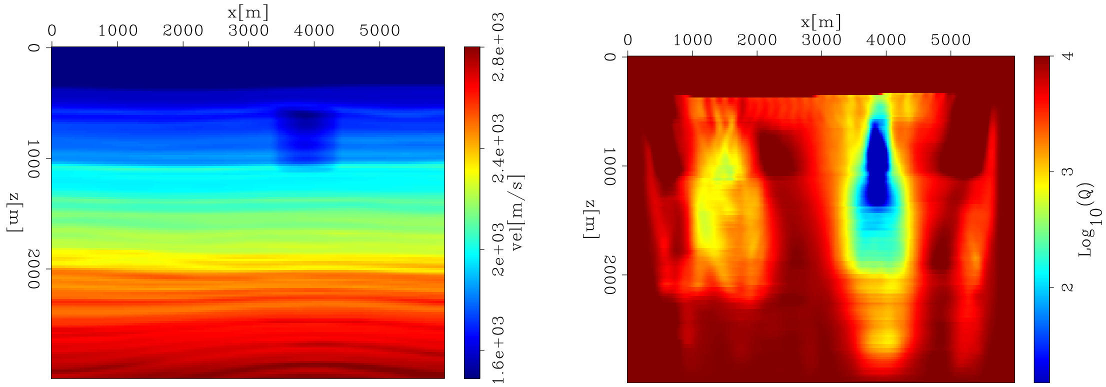
(a) (b)

Figure 1: True models: (a) A part of a modified SEAM velocity model with two gas clouds; (b) Q model (in logarithmic scale) with two gas clouds. [**CR**] yishen1/. seam2D.vel.2gas,seam2D.Q

The first test in this example is to invert for the Q model with the inaccurate velocity model shown in Figure 2(a). The inaccurate velocity in Figure 2(a) has the same background velocity as that in Figure 1(a). However, the velocity of the left gas cloud in Figure 2(a) is slightly higher than the true velocity in Figure 1(a) and is set to be the same as the surrounding sediments; while the velocity of the right gas cloud in Figure 2(a) remains unchanged from the true velocity in Figure 1(a).

The initial model for Q inversion is a model without attenuation. Figure 2(b) shows the inversion results (in logarithmic scale) using WEMQA (Shen et al., 2013, 2014). The results show that this Q inversion method with adequately accurate velocity information of the right part of the model, as shown in Figure 2(a), sufficiently recovers the location and value of the right gas cloud, as shown in Figure 2(b). However, this method with inaccurate velocity in the left part of the model, as shown in Figure 2(a), fails in retrieving the left gas cloud. The main reason for this failure is the inaccurate velocity that distorts the kinematics of the migrated structures, and subsequently degrades the accuracy of the spectra analysis for Q inversion. Therefore, simultaneous inversion of both velocity and Q models is needed to obtain a reasonable inversion results, if accurate information of these models is not available.

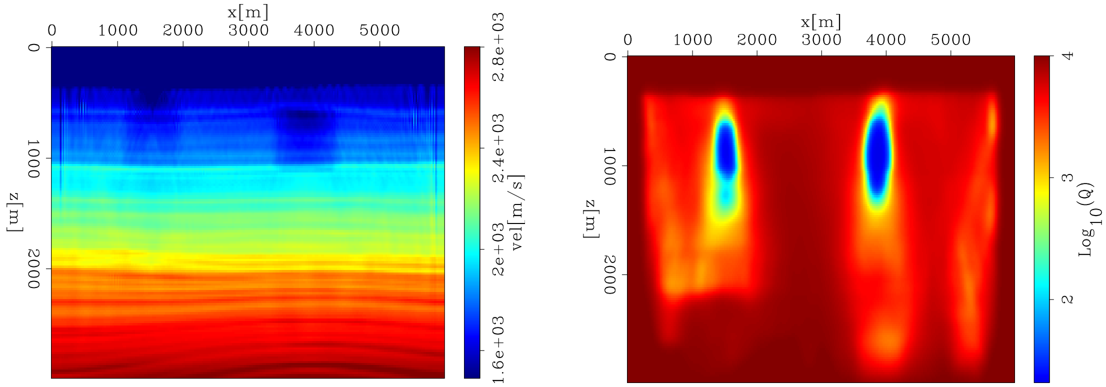
To simultaneously invert for velocity and Q models, the initial velocity model has the same background velocity and right gas velocity, as shown in Figure 1(a), but without the velocity drop in the left gas cloud. The initial Q model is a model without attenuation. Figure 3(a) is the inverted velocity model and Figure 3(b) is the inverted Q model. Simultaneous inversion successfully retrieves the locations and values of both gas clouds in



(a) (b)

Figure 2: (a) Inaccurate initial velocity model for Q inversion with only one gas cloud instead of two. Initial Q is constant. (b) Inverted Q model using inaccurate velocity model in Figure 2(a). [**CR**] yishen1/. seam2D.vel.1gas,seam2D.iQ.xvel

velocity and Q models, as shown in Figure 3.

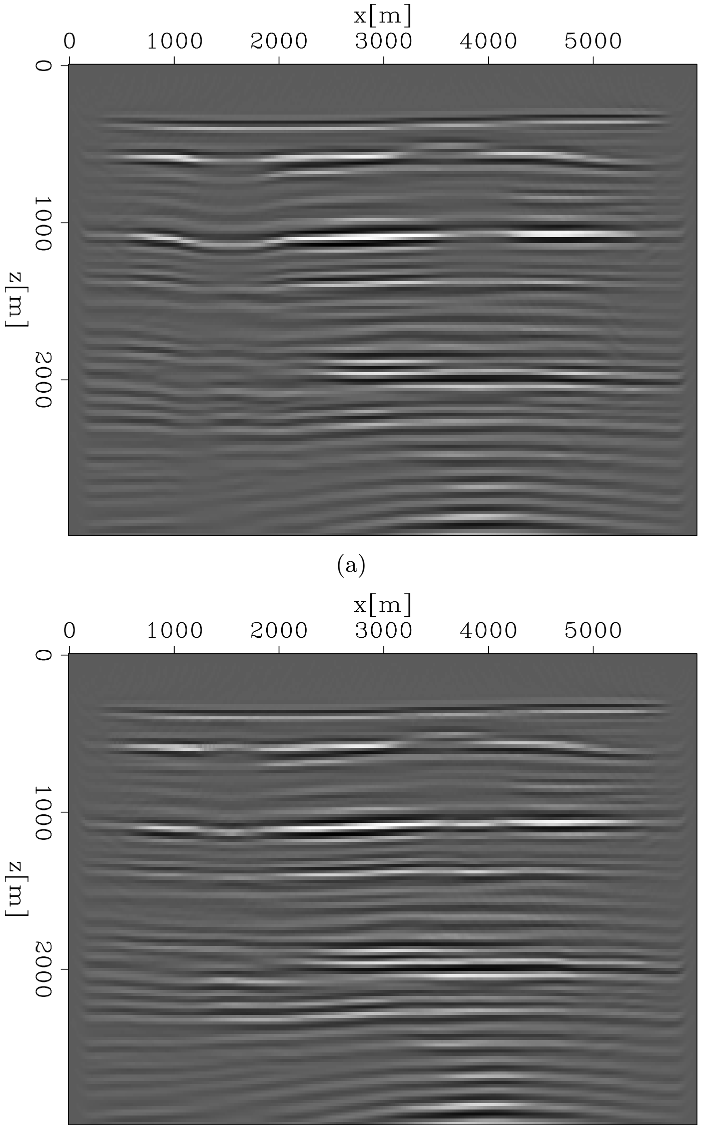


(a) (b)

Figure 3: Simultaneous inversion results: (a) The inverted velocity model. Note how the gas cloud on the left has been recovered. (b) The inverted Q model. The Q value of the left gas cloud has been recovered. [**CR**] yishen1/. seam2D.ivel,seam2D.iQ

Figure 4(a) is the migrated image using the initial velocity and Q models. The initial velocity model has a larger velocity in the left gas cloud, which causes the events below to be pushed downward and discontinuous. Attenuation caused by these two gas clouds degrades the quality of the deep imaged structures in Figure 4(a), in terms of dimming the amplitudes, making the events incoherent and stretching the wavelets. Figure 4(b) is the migrated image using the inverted models in Figure 3. Migration with the improved velocity model in Figure 3(a) moves the events below the left gas cloud upward and makes the events there more coherent. Also, compensation with the inverted Q model shown in Figure 3(b) makes the events sharper and more balanced in both phase and amplitudes, as shown in Figure 4(b).

Figure 5(a) and Figure 5(b) are the angle domain common image gathers(ADCIG)

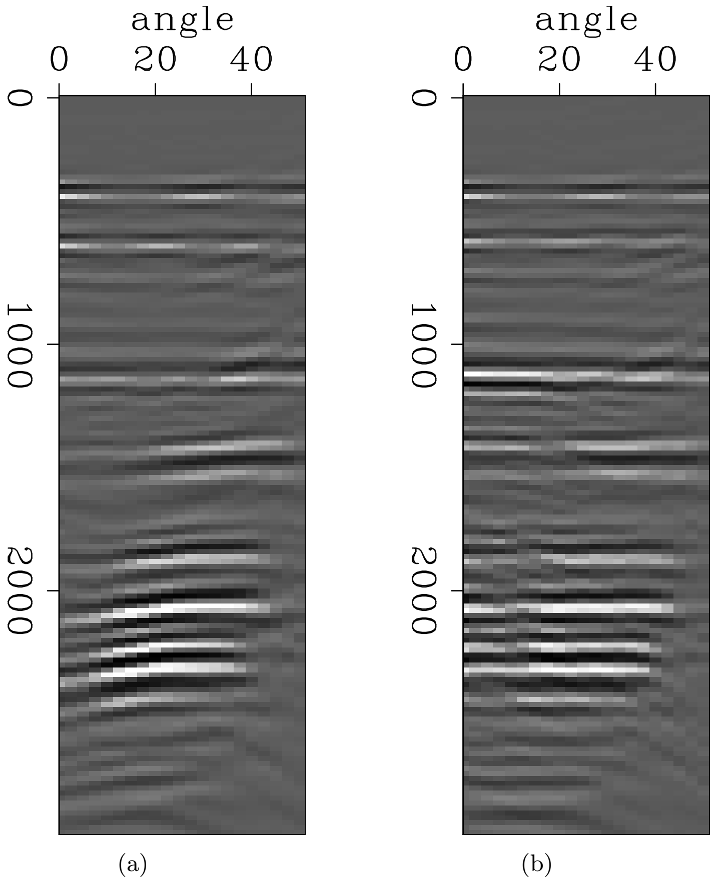


(a)

(b)

Figure 4: (a) The migrated image using the initial velocity and Q models; (b)The migrated image using the inverted models in Figure 3. The kinematics and the amplitudes under the gas cloud are corrected for by the inverted model. [**CR**] yishen1/. seam2D.bimg.bvbq,seam2D.img.iviq

extracted from the left gas cloud location (x= 1500 m) and obtained with the initial models and the inverted models in Figure 3, respectively. The inaccurate large velocity causes the events to be unflattened, as shown in Figure 5(a). The inverted velocity model shown in Figure 3(a) corrects such kinematics error caused by such wrong velocity and flattens the events in Figure 5(b). In addition, migration with the inverted Q model in Figure 3(b) compensates for the energy loss that appears especially strong at the near angle as shown in Figure 5(a), and therefore makes the amplitude of the events more balanced in Figure 5(b).



(a) (b)

Figure 5: (a)The angle domain common image gathers(ADCIG) extracted from the left gas cloud location (x= 1500 m) and obtained with the initial models. The vertical axis is depth with unit of meter. (b) The angle domain common image gather(ADCIG) extracted from the left gas cloud location (x= 1500 m) and obtained with the inverted models shown in Figure 3. The vertical axis is depth with unit of meter. The events are flattened, and the low angle amplitudes have been recovered. [**CR**] yishen1/. seam2D.bang.lgas,seam2D.iang.lgas

Figure 6(a) and Figure 6(b) are the angle domain common image gathers(ADCIG) extracted from the right gas cloud location (x= 3800 m) and obtained with the initial models and the inverted models in Figure 3, respectively. The near angles in Figure 6(a) have low amplitudes, stretched wavelets and unflattend events caused by attenuation, although the velocity used in this region is correct. Imaging with the inverted Q model in Figure 3(b) compensates the high frequency loss caused by attenuation, therefore, it recovers the amplitudes and sharpens the events at the near angles in Figure 6(b). In addition, such compensation corrects the phase distortion and velocity dispersion caused by attenuation. Subsequently, the events in Figure 6(b) become more flattened and more coherent than the events in Figure 6(a).

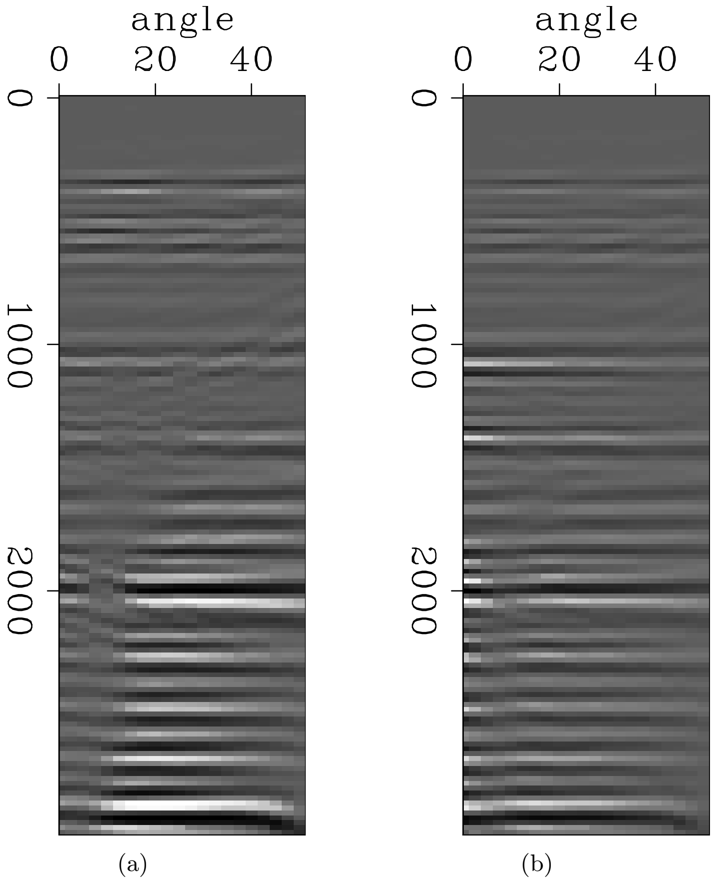


Figure 6: (a)The angle domain common image gathers(ADCIG) extracted from the right gas cloud location (x= 3800 m) and obtained with the initial models. The vertical axis is depth with unit of meter. (b) The angle domain common image gather(ADCIG) extracted from the right gas cloud location (x= 3800 m) and obtained with the inverted models in Figure 3. The vertical axis is depth with unit of meter. Imaging with the inverted Q model recovers the amplitudes and sharpens the events at the near angles. Such compensation also corrects the phase distortion and velocity dispersion caused by attenuation, which makes the events more flattened and more coherent. [**CR**] yishen1/. seam2D.bang.rgas,seam2D.iang.rgas

CONCLUSION

I developed a method for simultaneous inversion of velocity and Q models. This method poses the simultaneous estimation problem as an optimization problem that seeks optimum velocity and Q models by minimizing user-defined image residuals. The numerical tests on a modified SEAM model with two gas clouds demonstrate the benefit of using such simultaneous inversion, when the existing velocity and Q models are inaccurate. The results show that this simultaneous inversion method is able to retrieve both velocity and Q models, and to correct and compensate the distorted migrated image caused by inaccurate velocity and Q models.

ACKNOWLEDGMENTS

The author thanks Biondo Biondi and Robert Clapp, from the Stanford Exploration Project, for their advice and suggestions. Special thanks to Yunyue Li, Yang Zhang and Guillaume Barnier for their fruitful discussions.

REFERENCES

Billette, F. and S. Brandsberg-Dahl, 2005, The 2004 BP velocity benchmark: 67th Conference and Exhibition, EAGE, Extended Abstracts, B035.

Shen, Y., B. Biondi, R. Clapp, and D. Nichols, 2013, Wave-equation migration Q analysis (WEMQA): EAGE Workshop on Seismic Attenuation Extended Abstract O22.

———, 2014, Wave-equation migration Q analysis (WEMQA): SEG Technical Program Expanded Abstracts.

Tang, Y., 2011, Imaging and velocity analysis by target-oriented wavefield inversion: PhD thesis, Stanford University.

Tonn, R., 1991, The determination of seismic quality factor Q from VSP data: A comparison of different computational techniques: Geophysical Prospecting, **45**, 87–109.

Two-way wave-equation operators for non-constant density acoustic isotropic media

Ettore Biondi and Ossian O'Reilly

ABSTRACT

We derive two-way wave-equation operators in the time domain for isotropic non-constant density media with a finite-difference scheme. We present the chains of linear operators necessary for non-linear modeling, linearized modeling, and non-constant density migration. We also show that radiation patterns obtained from linearized modeling agree with theoretical results. With these correct forward-adjoint operators, multi-parameter full waveform inversion for non-constant density media can be pursued.

INTRODUCTION

Seismic data inversion is and has been one of the most challenging problems encountered by geophysicists. Since its first envision by Tarantola (1984), full waveform inversion (FWI) was derived in the isotropic non-constant density acoustic approximation. Moreover, it is known that multi-parameter FWI is now essential to correctly match real data amplitudes (Operto et al., 2013). Therefore, having reliable non-linear modeling operators for non-constant density is very important in case of acoustic approximation. In addition, obtaining accurate adjoint operators is fundamental to achieve optimal convergence rates during seismic data inversion (Ji, 2009).

We implement a chain of linear operators for two-way wave-equation non-linear and linearized modeling for non-constant density media. The derivation is carried out in the time domain with a staggered-grid finite-difference scheme. For the linearized modeling operator the radiation patterns are compared with the theoretical ones (Aki and Richards, 2002). These accurate operators are extremely important in case we perform modeling and FWI in acoustic approximation.

ACOUSTIC WAVE EQUATION AND BORN APPROXIMATION

As shown in the Appendix, the non-constant density wave equation for acoustic media can be derived assuming that shear stresses are null and can be written as:

$$\left[\frac{1}{K(\mathbf{r})}\frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho(\mathbf{r})}\nabla\right] p(\mathbf{r},t) = s(\mathbf{r},t), \tag{1}$$

where $K(\mathbf{r})$ and $\rho(\mathbf{r})$ are the medium's bulk modulus and density, respectively; $p(\mathbf{r},t)$ is the propagating pressure, $s(\mathbf{r},t)$ is the source term and $\mathbf{r}$ is the position vector defining each point in the medium. By perturbing the medium's properties and neglecting the higher-order terms in the equation we can derive the linearized wave equation, also known as the

Born approximation. This wave equation is linear with respect to property perturbations $\delta K(\mathbf{r})$ and $\delta p(\mathbf{r})$, and is defined as follows:

$$\left[\frac{1}{K_0(\mathbf{r})}\frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho_0(\mathbf{r})}\nabla\right]\delta p(\mathbf{r},t) = \frac{\delta K(\mathbf{r})}{K_0(\mathbf{r})^2}\frac{\partial^2 p_0(\mathbf{r},t)}{\partial t^2} - \nabla \cdot \frac{\delta\rho(\mathbf{r})}{\rho_0(\mathbf{r})^2}\nabla p_0(\mathbf{r},t), \tag{2}$$

where $K_0(\mathbf{r})$ and $p_0(\mathbf{r})$ are the background medium's properties, $p_0(\mathbf{r},t)$ is the pressure field propagated through the background medium, and $\delta p(\mathbf{r},t)$ is the scattered pressure field. In Equation 2 the two right-hand side terms are also commonly called secondary sources. These two terms represent the pressure scattering off perturbations in the bulk modulus and density respectively. Because our model is composed of K and ρ, we have two different imaging conditions for the two parameters, and are given by:

$$\delta\tilde{K}(\mathbf{r}) = \frac{1}{K_0(\mathbf{r})^2}\left[\frac{\partial^2 p_0(\mathbf{r})}{\partial t^2} \star \delta p'(\mathbf{r})\right](0), \tag{3}$$

$$\delta\tilde{\rho}(\mathbf{r}) = \frac{1}{\rho_0(\mathbf{r})^2}\left[\nabla p_0(\mathbf{r}) \underset{\cdot}{\star} \nabla\delta p'(\mathbf{r})\right](0), \tag{4}$$

where $[f \star g](0)$ and $[f \underset{\cdot}{\star} g](0)$ denote the zero-lag crosscorrelation and the zero-lag crosscorrelation of the scalar product of two functions respectively, and:

$$\delta p'(\mathbf{r},t) = \sum_{i=1}^{N_d} \tilde{g}(\mathbf{r},t,\mathbf{r}_i;K,\rho) * \delta d(\mathbf{r}_i,t), \tag{5}$$

in which $\delta p'(\mathbf{r},t)$ represents the back-propagated data perturbations or residual during an inversion, obtained by convolving the $\delta d(\mathbf{r}_i,t)$ with the anticausal Green's function $\tilde{g}(\mathbf{r},t,\mathbf{r}_i;K,\rho)$ for all the N_g receivers and a given source. The reader interested in the derivation of these equations can find them in the Appendix.

FORWARD AND ADJOINT PROPAGATION

The non-constant density wave equation in acoustic approximation (1) is linear with respect to the source term on the right-hand side. In fact, as shown by Almomin (2013) for the constant-density case, we can write a linear operator as:

$$\mathbf{d} = \tilde{\mathbf{F}}\mathbf{w}, \tag{6}$$

where $\mathbf{d}$ is model data, $\tilde{\mathbf{F}}$ that is the propagator which is non-linear with respect to bulk modulus and density, and $\mathbf{w}$ is the source wavelet. The propagator can be split into a chain of linear operators as:

$$\mathbf{d} = \mathbf{R}^*\mathbf{L}_g^*\mathbf{F}\mathbf{L}_s\mathbf{R}\mathbf{w}, \tag{7}$$

where $\mathbf{R}$ is a time interpolator that resamples the source wavelet from seismic sampling to finer sampling to have stable propagation; $\mathbf{L}_s$ is a space interpolation to inject the wavelet at the source position into the model, $\mathbf{F}$ is the operator that solve equation 1; $\mathbf{L}_g^*$ is again a space interpolator that extracts the pressure field at the receiver location; and $\mathbf{R}^*$ is the adjoint of the time interpolator that transforms the extracted data from fine sampling to seismic sampling.

To derive the actual form of $\mathbf{F}$, we need to discretize both time and space in equation 1. Using a simple second-order approximation for the time derivative (i.e., using a leapfrog time integrator), we can write the following recursive equation for each point in the model grid:

$$\mathbf{p}(it) = \mathbf{q}(it-1) + \mathbf{C}\mathbf{p}(it-1) - \mathbf{p}(it-2), \tag{8}$$

where $\mathbf{p}(it)$ denotes the pressure field at the time step it, $\mathbf{q}(it-1)$ is the scaled source wavelet obtained as follows:

$$\mathbf{q}(it-1) = \mathbf{S}\mathbf{w}(it-1) = \mathbf{K}\Delta t^2 \mathbf{w}(it-1), \tag{9}$$

and the operator $\mathbf{C}$ is given by:

$$\mathbf{C} = \mathbf{K}\Delta t^2 \nabla \cdot \frac{1}{\rho}\nabla, \tag{10}$$

in which $\mathbf{K}$ is a diagonal operator containing the bulk modulus of the medium; Δt is the fine propagation time sampling; and $\nabla \cdot \frac{1}{\rho}\nabla$ denotes an operator that computes the divergence of a density-scaled gradient of $\mathbf{p}(it-1)$. It is known that central different operators produce inaccurate derivative computation that introduces numerical error during the propagation (Mattsson, 2012); therefore, we implement the $\nabla \cdot \frac{1}{\rho}\nabla$ operator on a staggered grid (Figure 1). This operator in 2D can be written as follows:

$$\nabla \cdot \frac{1}{\rho}\nabla = \mathbf{D}_x^- \mathbf{B}_x \mathbf{D}_x^+ + \mathbf{D}_z^- \mathbf{B}_z \mathbf{D}_z^+, \tag{11}$$

where $\mathbf{D}^+$ and $\mathbf{D}^-$ are the forward and backward first-order derivative operators respectively, and $\mathbf{B}$ operators are diagonal matrices that contain the staggered inverse of the density values along the main diagonal. To compute these values we use a simple average on the inverse of the density along each direction.

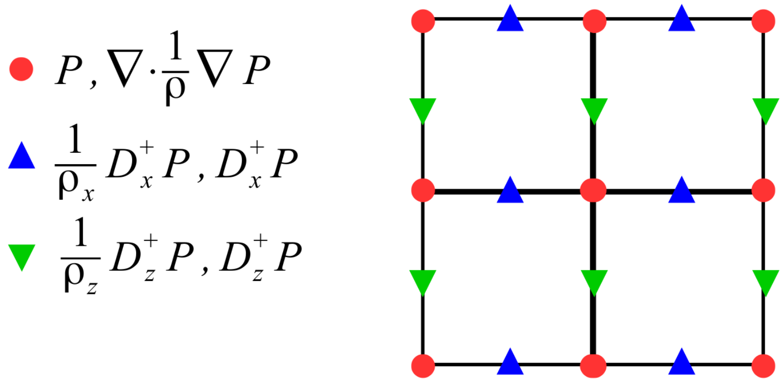

$P, \nabla\cdot\frac{1}{\rho}\nabla P$
$\frac{1}{\rho_x} D_x^+ P, D_x^+ P$
$\frac{1}{\rho_z} D_z^+ P, D_z^+ P$

Figure 1: 2D staggered-grid scheme used for computing the $\nabla \cdot \frac{1}{\rho}\nabla$. The pressure and the output of this operator lay on the regular grid (red circles). Instead, the first-order derivative in the two spatial dimension and their density-scaled versions are computed on two different staggered grids (blue and green triangles). [**NR**] ettore1/. staggered-grid-scheme

The adjoint operator of $\tilde{\mathbf{F}}$ can be easily obtained by taking the adjoint of the chain of operators of Equation 7 that is:

$$\tilde{\mathbf{w}} = \mathbf{R}^*\mathbf{L}_s^*\mathbf{S}\mathbf{P}^*\mathbf{L}_g\mathbf{R}\mathbf{d}, \tag{12}$$

where we have split the propagation operator as:

$$\mathbf{F} = \mathbf{P}\mathbf{S}, \tag{13}$$

which is a chain of a scaling operator $\mathbf{S}$ and a recursive operator $\mathbf{P}$. The adjoint of the recursive operator is given by:

$$\mathbf{q}(it) = \mathbf{p}(it+1) + \mathbf{C}^*\mathbf{q}(it+1) - \mathbf{q}(it+2), \tag{14}$$

where we are back-solving a system of equations for $\mathbf{q}$, which corresponds to back propagate the data from the receivers; and the adjoint of $\mathbf{C}$ is written as follows:

$$\mathbf{C}^* = \left(\nabla \cdot \frac{1}{\rho}\nabla\right)^* \mathbf{K}\Delta t^2. \tag{15}$$

The adjoint divergence of the scaled gradient is defined as:

$$\left(\nabla \cdot \frac{1}{\rho}\nabla\right)^* = {\mathbf{D}_x^+}^*\mathbf{B}_x{\mathbf{D}_x^-}^* + {\mathbf{D}_z^+}^*\mathbf{B}_z{\mathbf{D}_z^-}^*. \tag{16}$$

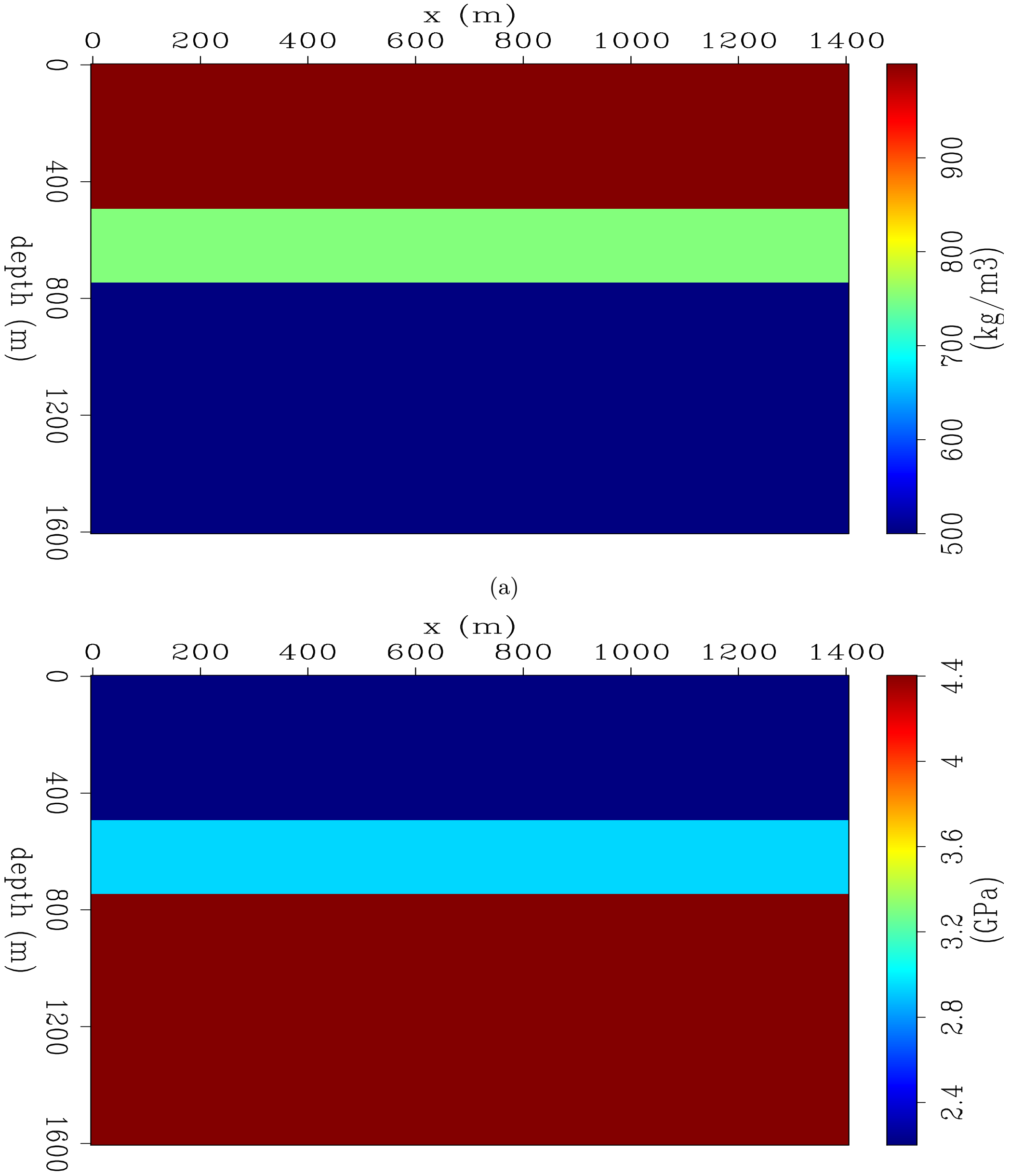
As shown in the Appendix, if we assume the pressure field is null outside the computational domain, we have:

$$\begin{aligned} {\mathbf{D}^+}^* &= -\mathbf{D}^-, \qquad (17)\\ {\mathbf{D}^-}^* &= -\mathbf{D}^+, \qquad (18) \end{aligned}$$

that cause the operator $\nabla \cdot \frac{1}{\rho}\nabla$ to be self-adjoint.

Forward and adjoint propagation examples

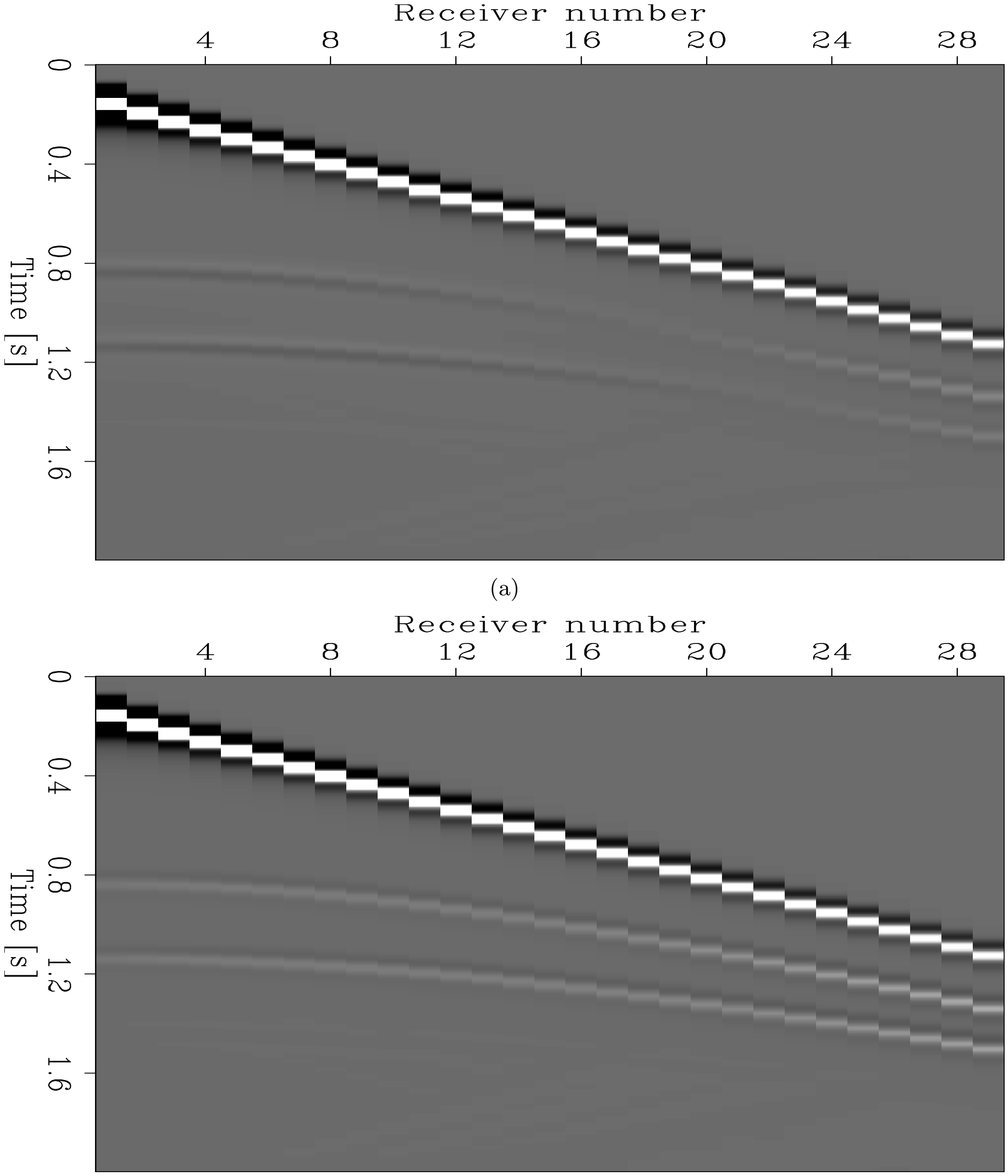
We run two numerical experiments in which we created a model with two velocity constrasts, one case these variations are caused by density changes only; whereas, in the second case we vary the bulk modulus only. Figure 2 shows the two models used for the numerical tests. In these tests we use a single source placed at the origin, and evenly spaced receivers at the surface with receiver interval of 50 m. In both examples we use a Ricker wavelet with central frequency of $10Hz$. We use 8th-order accuracy staggered-grid operators and 5 points per wavelength for the minimum one which is $30m$. Moreover, as explained in Almomin (2013), to damp spurious reflections from the model boundary, we add the absorbing boundaries described by Israeli and Orszag (1981) in the recursive equations used for propagating the pressure field. Figure 3 displays the data obtained from the two numerical simulations. In the top panel, we observe the data generated by propagating the source in the varying density model; in the bottom panel, we show the data for the varying bulk modulus. In both data, we clearly distinguish the direct arrival and two primary reflections. It is notable that for the varying density case the impedance contrasts are negative; and also that phase rotations are present as we increase receiver offset. When we vary the compressibility only,



(a)

(b)

Figure 2: Density and bulk modulus models used in two different numerical experiments having the same velocity contrasts. (a) Density model used in the first experiment where the bulk modulus is kept constant and equal to $2.2 GPa$. (b) Bulk modulus model used in the second experiment where the density is $1000 kg/m^3$ [**ER**] ettore1/. rho-1,bulk-2



(a)

(b)

Figure 3: Data resulting from the numerical tests. (a) Data from the varying density model of Figure2a. In this case the impedance contrasts are negative and we also observe phase rotations for increasing offset. (b) Data from the varying density model of Figure 2b. In this model the impedance contrasts are positive. Moreover, we do not observe phase rotations. [**ER**] ettore1/. data-1,data-2

using the same velocity contrasts of the varying density case, we obtain positive impedance contrasts and do not observe any phase rotations for increasing offset.

Since we also have at our disposal the adjoint operator of the propagator (Equation 12), we run an adjoint experiment with the varying density model and the data of Figure 3a. As expected, for a simple geometry, where a single source and multiple receivers are deployed, the output of the adjoint operator is a scaled version of the source wavelet injected for recording the data of Figure 3a (Figure 4). We also plot the forward and adjoint pressure fields for three different times in Figure 5. The forward pressure field is propagating in the positive time direction; whereas, the adjoint field is collapsing at the receiver locations as we increase the propagation time. In both cases we have reflections coming from the two contrasts. In the adjoint wavefield, we notice the propagating receivers have a strong directed arrival because this wave is in phase for all the receivers.

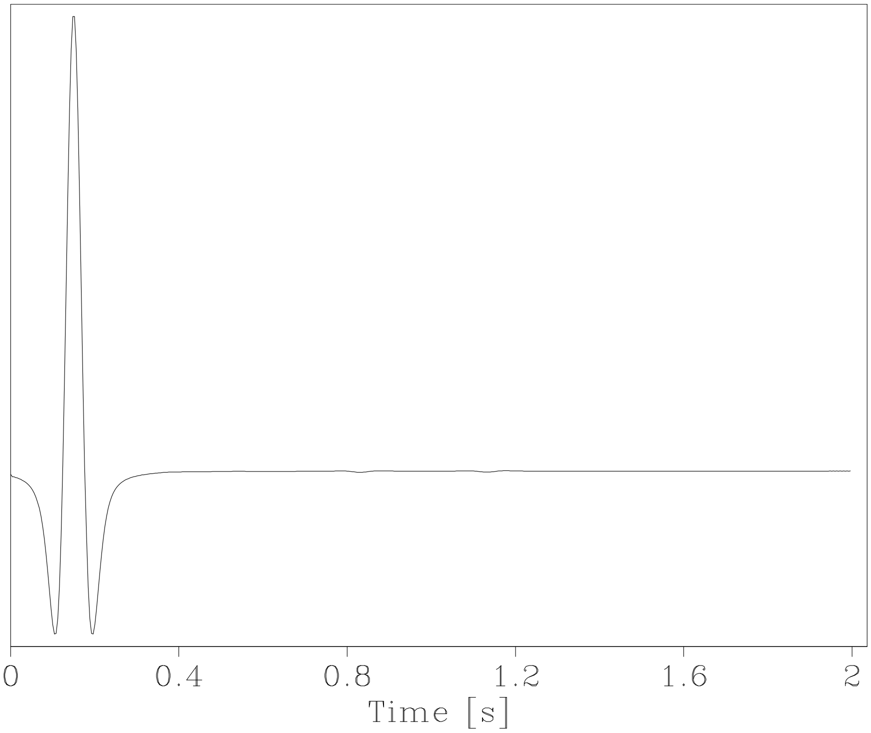


Figure 4: Adjoint source wavelet obtained using the adjoint propagator on the modeled data of Figure 3a with the varying density model of Figure 2a. [**ER**] ettore1/. back-data

BORN AND RTM OPERATORS

To write the chain of linear operators for the Born approximation and its adjoint operator (also known as RTM operator), we can use most of the operators employed in forward and adjoint propagation. However, we need to take care of the scattering condition and its adjoint shown in Equations 2, 3, and 4. The Born operator can be written as follows:

$$\Delta \mathbf{d} = \mathbf{B}\Delta \mathbf{m} = \mathbf{B} \begin{bmatrix} \Delta \mathbf{K} \\ \Delta \boldsymbol{\rho} \end{bmatrix}, \tag{19}$$

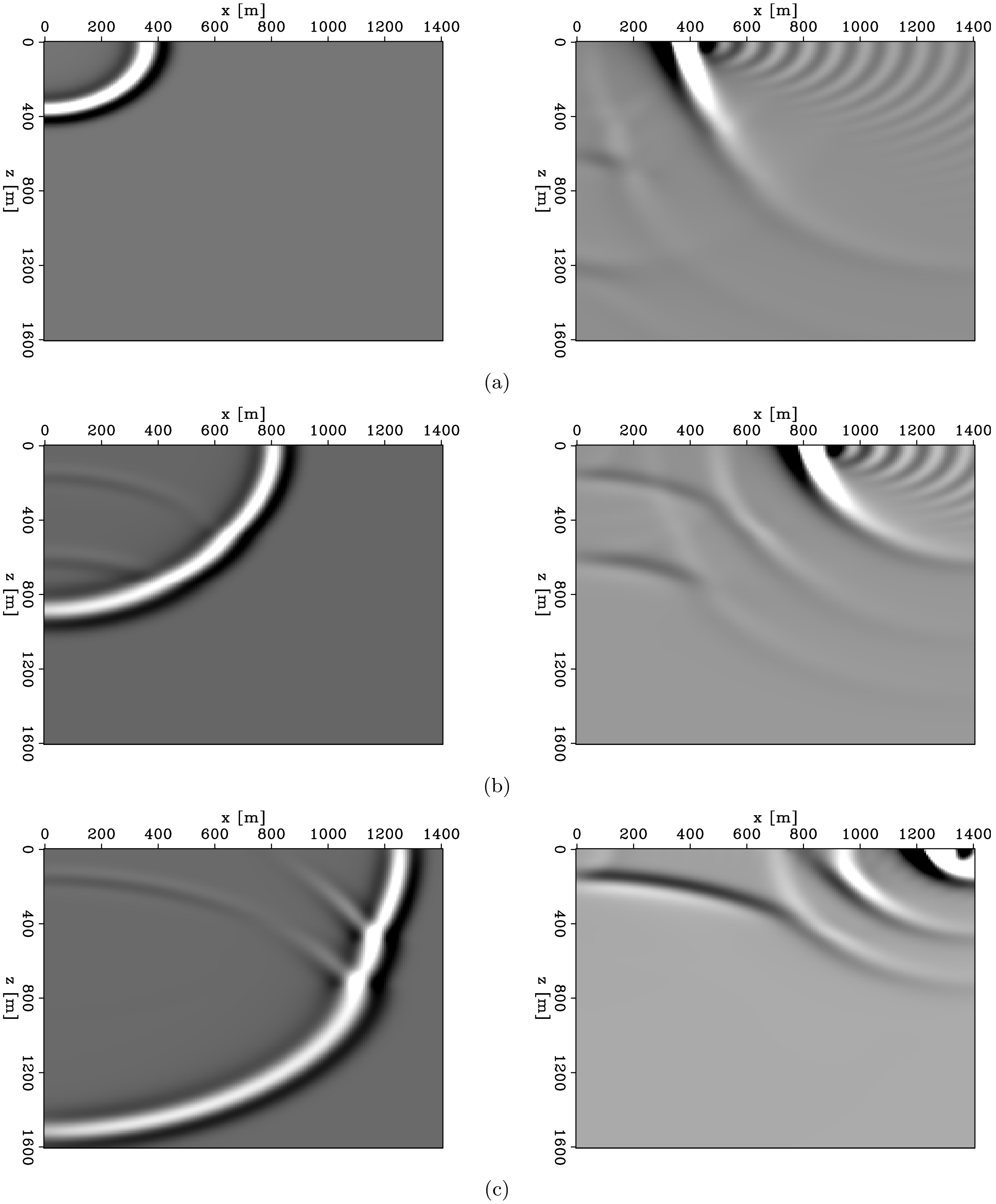


Figure 5: Forward (left panels) and adjoint (right panels) pressure fields at (a) 0.4s, (b) 0.7s, and (c) 1.0s. Since we back-solve the system of equations for the propagation, the adjoint wavefield is running backward in time. [**ER**] ettore1/. fields-1,fields-2,fields-3

where we obtain perturbed data $\Delta\mathbf{d}$ from model perturbations $\Delta\mathbf{m}$ in the background compressibility and density. In this equation $\Delta\mathbf{K}$ and $\Delta\boldsymbol{\rho}$ represent two vectors containing the bulk modulus and density perturbations for all the points in the model, respectively. The operator $\mathbf{B}$ is expanded into the following series of linear operators:

$$\Delta\mathbf{d} = \mathbf{R}^*\mathbf{L}_g^*\mathbf{F}\mathbf{R}\begin{bmatrix}\ddot{\mathbf{P}}_\mathbf{0} & \tilde{\mathbf{P}}_\mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{M}_\mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\boldsymbol{\rho}}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{K} \\ \Delta\boldsymbol{\rho}\end{bmatrix}, \tag{20}$$

where we notice part of the forward chain of operators used in Equation 7, which takes care of propagating the scattered pressure coming from the other operator on the right of it. The matrix of operators containing $\mathbf{M}_\mathbf{K}$ and $\mathbf{M}_{\boldsymbol{\rho}}$ is a scaling-spreading operator that scales the two model perturbations with the inverse of the square of the background property (i.e., $1/K_0(\mathbf{r})^2$ and $-1/\rho_0(\mathbf{r})^2$ in Equation 2) and spreads these scaled perturbations to all time steps. The two stacked operators $\ddot{\mathbf{P}}_\mathbf{0}$ and $\tilde{\mathbf{P}}_\mathbf{0}$ contain both the forward propagated pressure field in the background properties $\mathbf{K}_0$ and $\boldsymbol{\rho}_0$ obtained as follows:

$$\mathbf{p}_0 = \mathbf{R}^*\mathbf{F}\mathbf{L}_s\mathbf{R}\mathbf{w} \tag{21}$$

downsampling the propagated source wavefield by applying $\mathbf{R}^*$ to save computational time when calculating the scattered pressure. Note that after scattering, we need to resample it with $\mathbf{R}$. To maintain the same amplitudes, we need to have $\mathbf{R}^*\mathbf{R} \approx \mathbf{I}$. The scattering operator $\ddot{\mathbf{P}}_\mathbf{0}$, which accounts for perturbations in the compressibility, is obtained as:

$$diag(\ddot{\mathbf{P}}_\mathbf{0}) = \mathbf{D}_\mathbf{2}\mathbf{p}_0, \tag{22}$$

where $\mathbf{D}_\mathbf{2}$ outputs the second-order time derivative of the forward pressure field. The other scattering operator $\tilde{\mathbf{P}}_\mathbf{0}$, which accounts for density perturbations, is more complex than the $\ddot{\mathbf{P}}_\mathbf{0}$ operator, and compose a chain of linear operators written as follows:

$$\tilde{\mathbf{P}}_\mathbf{0} = \begin{bmatrix}\mathbf{D}_\mathbf{z}^- & \mathbf{D}_\mathbf{x}^-\end{bmatrix}\begin{bmatrix}diag(\mathbf{D}_\mathbf{z}^+ p_0) & \mathbf{0} \\ \mathbf{0} & diag(\mathbf{D}_\mathbf{x}^+ p_0)\end{bmatrix}\begin{bmatrix}\mathbf{G}_\mathbf{z}^+ \\ \mathbf{G}_\mathbf{x}^+\end{bmatrix}, \tag{23}$$

where the first stacked operators $\mathbf{G}$ stagger the density perturbations into the two staggered grids shown in Figure 1. Then, we multiply the staggered density perturbations with the gradient of the pressure field and afterward compute the divergence with the two $\mathbf{D}^-$ operators in the respective direction. The RTM operator is simply the adjoint of the chain of operator of Equation 20 that is:

$$\begin{bmatrix}\Delta\tilde{\mathbf{K}} \\ \Delta\tilde{\boldsymbol{\rho}}\end{bmatrix} = \begin{bmatrix}\mathbf{M}_\mathbf{K}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\boldsymbol{\rho}}^*\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{P}}_\mathbf{0} \\ \tilde{\mathbf{P}}_\mathbf{0}^*\end{bmatrix}\mathbf{R}^*\mathbf{F}^*\mathbf{L}_g\mathbf{R}\Delta\mathbf{d}, \tag{24}$$

where $\ddot{\mathbf{P}}_\mathbf{0}$ is self-adjoint, because it is a real diagonal operator, and the chain $\mathbf{R}^*\mathbf{F}^*\mathbf{L}_g\mathbf{R}$ is back-propagating data perturbations in the background model. The matrix of operators at the end of the chain is scaling the result of the output from the chain on the right of it, and taking the sum of it for each model point, which corresponds to zero-lag cross-correlation. We then clearly see that the chain $\mathbf{M}_\mathbf{K}^*\ddot{\mathbf{P}}_\mathbf{0}\mathbf{R}^*\mathbf{F}^*\mathbf{L}_g\mathbf{R}$ is computing Equation 3.

We have to verify that the chain $\mathbf{M}_{\boldsymbol{\rho}}^*\tilde{\mathbf{P}}_\mathbf{0}^*\mathbf{R}^*\mathbf{F}^*\mathbf{L}_g\mathbf{R}$ is actually equal to Equation 4. To this end we take the adjoint of equation 23 that is equal to:

$$\tilde{\mathbf{P}}_\mathbf{0}^* = \begin{bmatrix}\mathbf{G}_\mathbf{z}^{+*} & \mathbf{G}_\mathbf{x}^{+*}\end{bmatrix}\begin{bmatrix}diag(\mathbf{D}_\mathbf{z}^+ p_0)^* & \mathbf{0} \\ \mathbf{0} & diag(\mathbf{D}_\mathbf{x}^+ p_0)^*\end{bmatrix}\begin{bmatrix}\mathbf{D}_\mathbf{z}^{-*} \\ \mathbf{D}_\mathbf{x}^{-*}\end{bmatrix}, \tag{25}$$

and using the property of staggered-grid operators (17 and 18), it can be rewritten as:

$$\tilde{\mathbf{P}}_{\mathbf{0}}^{*} = -\begin{bmatrix}\mathbf{G}_{\mathbf{z}}^{+*} & \mathbf{G}_{\mathbf{x}}^{+*}\end{bmatrix} \begin{bmatrix} diag(\mathbf{D}_{\mathbf{z}}^{+} p_0)^{*} & \mathbf{0} \\ \mathbf{0} & diag(\mathbf{D}_{\mathbf{x}}^{+} p_0)^{*} \end{bmatrix} \begin{bmatrix}\mathbf{D}_{\mathbf{z}}^{+} \\ \mathbf{D}_{\mathbf{x}}^{+}\end{bmatrix}, \tag{26}$$

where we see that we are calculating the gradient of the back-propagated data perturbations; and then multiplying it with the gradient of forward pressure field with both gradients computed on the staggered-grids. We apply the adjoint of the stacked staggering operators $\mathbf{G}$ that shift the inputs from the staggered grids to the regular grid and compute the sum of the two outputs. All together we now see that the chain $\mathbf{M}_{\boldsymbol{\rho}}^{*}\tilde{\mathbf{P}}_{\mathbf{0}}^{*}\mathbf{R}^{*}\mathbf{F}^{*}\mathbf{L}_{g}\mathbf{R}$ is computing Equation 4.

Numerical tests of linearized operator and its adjoint

To test the Born operator, we run two simple experiments with same background medium's properties that we used in the previous example, where we perturb only the compressibility or the density, and we use the same acquisition geometry used in the previous propagations. We use constant background values equal to $2.2GPa$ and $1000kg/m^3$ and two single perturbations of $0.2GPa$ and $100kg/m^3$ for bulk modulus and density respectively. With these experiments we want to verify the following relations:

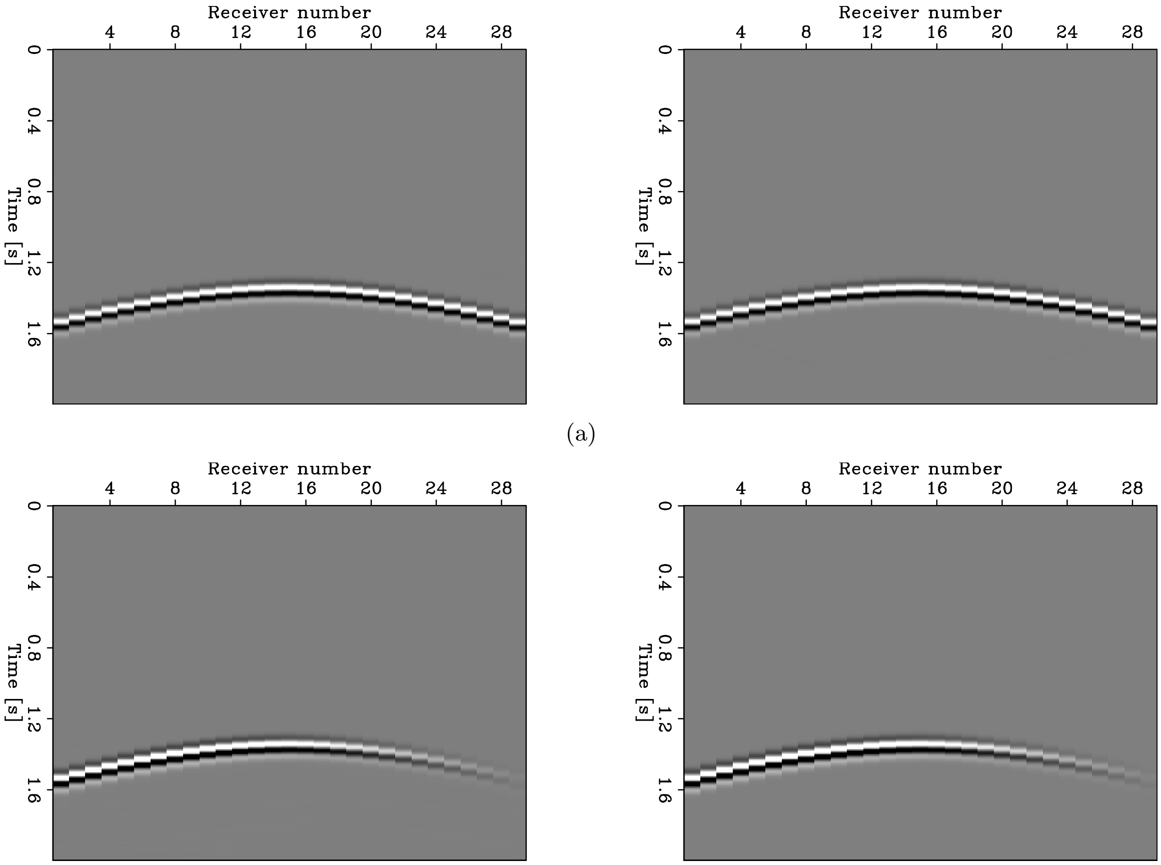
$$f(\mathbf{K}_0 + \Delta\mathbf{K}, \boldsymbol{\rho}_0) - f(\mathbf{K}_0, \boldsymbol{\rho}_0) \approx \mathbf{B}\Delta\mathbf{K}, \tag{27}$$

$$f(\mathbf{K}_0, \boldsymbol{\rho}_0 + \Delta\boldsymbol{\rho}) - f(\mathbf{K}_0, \boldsymbol{\rho}_0) \approx \mathbf{B}\Delta\boldsymbol{\rho}, \tag{28}$$

where $f(\mathbf{K}, \boldsymbol{\rho})$ represents the data generated by the propagation through the medium's properties $\mathbf{K}$ and $\boldsymbol{\rho}$. Here, we are verifying that the linearized data perturbations are actually modeled by the implemented Born operator. To verify this observation we need to have small model perturbations. Figure 6a shows the comparison of the data perturbation when we add a compressibility change in the model; while Figure 6b displays the comparison; but when we perturbe only the density model. The same clipping value is used for all the panels, and we can clearly see that for a single model perturbation the linearized scattered data matches the non-linear data perturbations.

To understand the reason why we observe different amplitude variations for increasing offsets only when perturb the density, we need to analyze the scattering pressure from the single diffraction point. Figure 7 shows the comparison of the scattered pressure and the theoretical radiation pattern (Aki and Richards, 2002) in case we perturb the bulk modulus. We observe that we have an isotropic energy scattering, which results in an isotropic back-scattered energy at the receivers (Figure 6a). Otherwise, if we look at the scattered energy from a density perturbation (Figure 8), we see an anisotropic scattered pressure. This observation is also confirmed by comparing this scattered pressure with the theoretical radiation pattern (Figure 8b). In fact, in the experiment of Figure 6b, because we are hitting the scatter not vertically, the scattered energy recorded at the receivers is affected by this anisotropic radiation pattern.

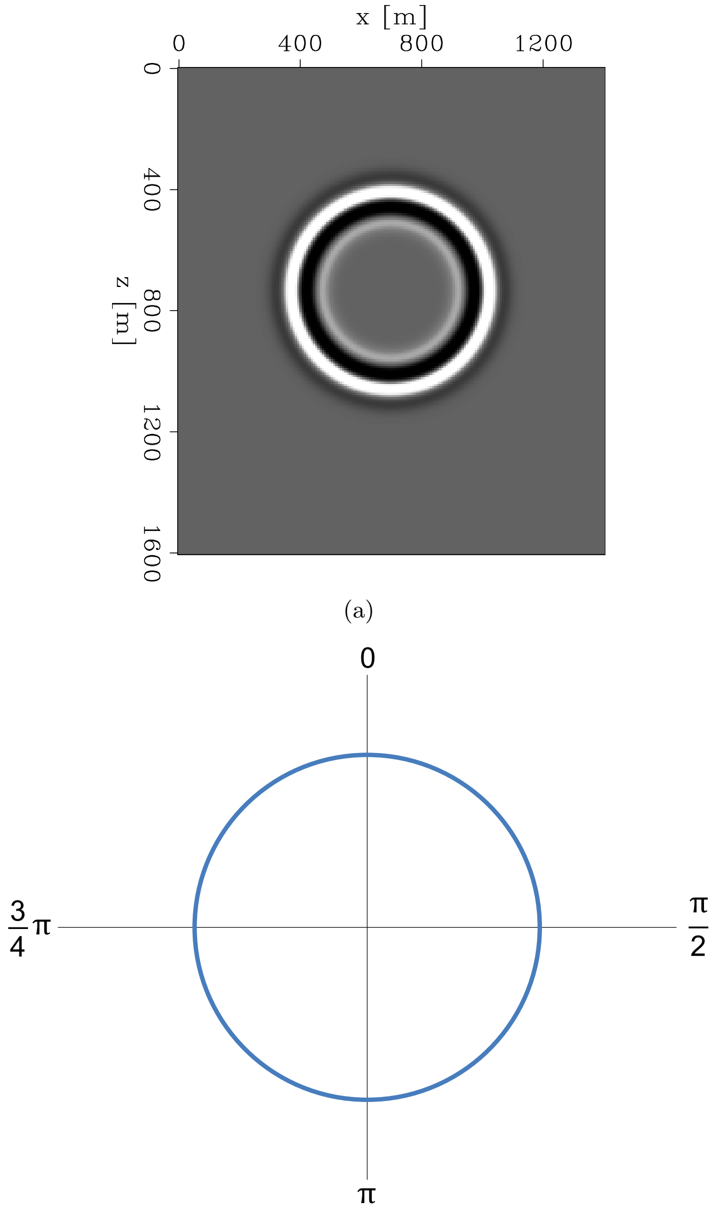
The last performed test is to apply the RTM operator to data perturbations. We apply this operator to the perturbed data obtained from a density perturbation (Figure 6b). Figure 9 shows the result of this application. We see that even if we have data generated only by a density perturbation some of the energy is leaking into the bulk modulus



(a)

(b)

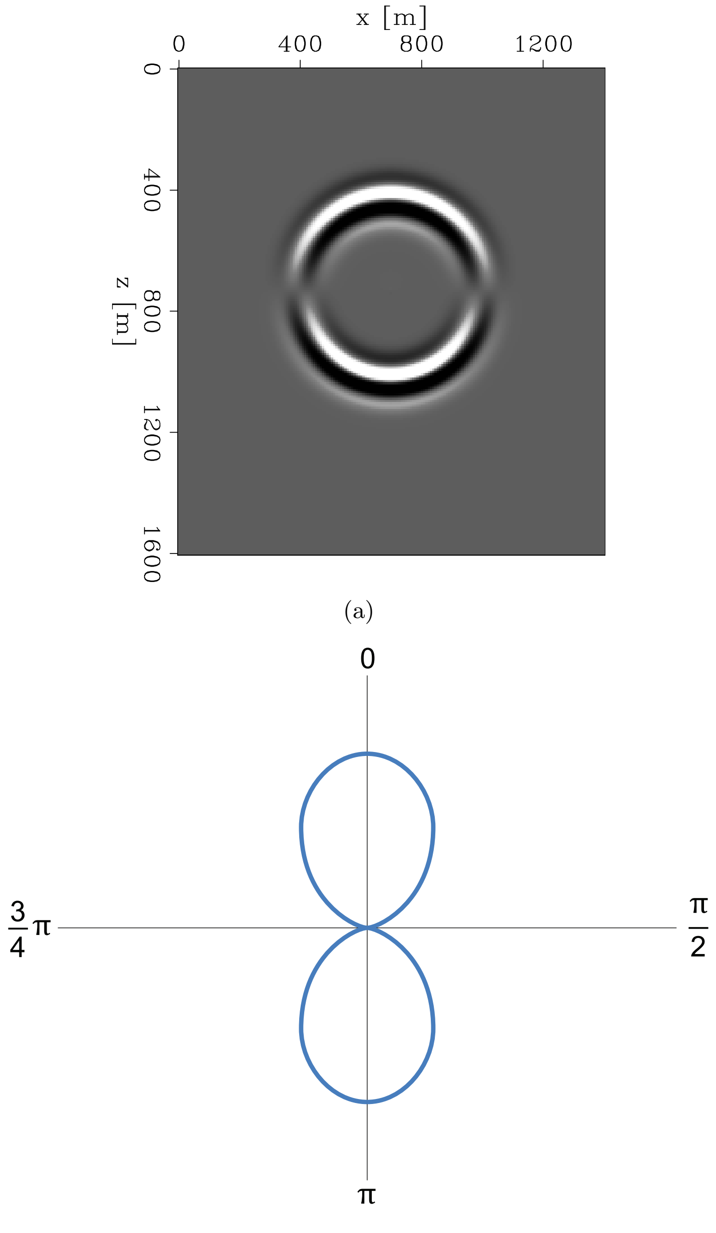
Figure 6: Comparison of the correct data perturbation given by the difference of the data obtained from a perturbed model and an unperturbed model (left panels), and the linearized data perturbation generated by the Born model operator (right panels). (a) Data comparison when we perturbe only the compressibility of the model (equation 27). (b) Same comparison but perturbing only the density model (equation 28). [**ER**] ettore1/. born-compare-2,born-compare-1



(a)

(b)

Figure 7: (a) Radiated energy from a single bulk modulus perturbation point. [**ER**](b) Theoretical radiation pattern of a vertical incident wave on a bulk modulus perturbation (Aki and Richards, 2002). [**NR**] ettore1/. scatter-bulk,radiation-bulk



(a)

(b)

Figure 8: (a) Radiated energy from a single density perturbation point. [**ER**](b) Theoretical radiation pattern of a vertical incident wave on a density perturbation (Aki and Richards, 2002). [**NR**] ettore1/. scatter-density,radiation-density

image. This effect is well known as parameter cross-talk (Operto et al., 2013). It is also important to notice that the energy of the density image is higher than in the compressibility image. One possible explanation could be because data contains both amplitude and phase information entangle together, and during FWI we are trying to simultaneously invert the two information that depend on a combination of physical parameters. In fact, it is known FWI gradient does not provide the correct amplitude of the model perturbations, first because the gradient does not have the correct units of the model perturbation and second because the gradient associated with one parameter can be affected by crosstalk from the other parameters. Hessian-matrix based inverse methods could be possible solutions to solve this problem (Innanen, 2014).

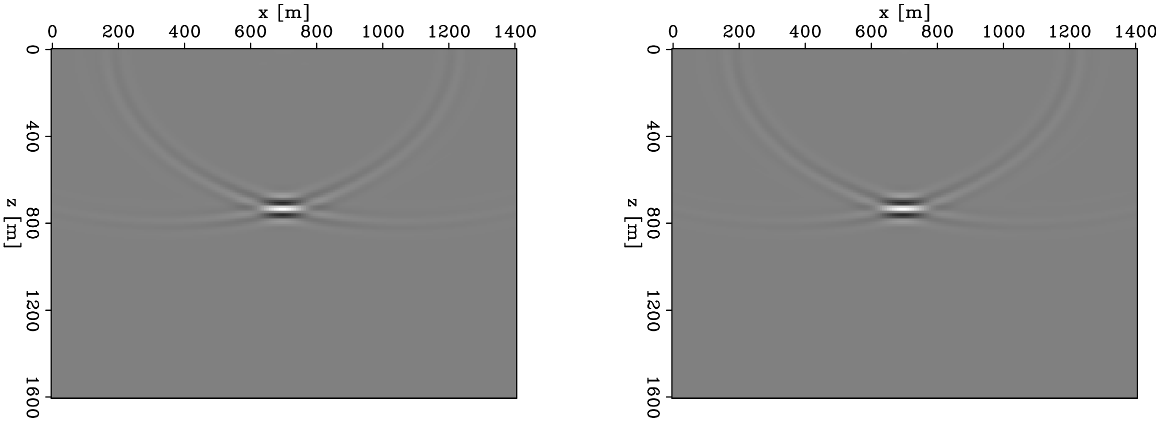


Figure 9: Images obtained applying the non-constant density RTM operator. The images look the same but the scales are different. The density image (right panel) is stronger than the bulk modulus image (left panel) by six orders of magnitude. [**ER**] ettore1/. images

CONCLUSIONS

We implemented and discussed the two-way wave-equation operators for non-constant density isotropic media. We explain that non-linear and linearized modeling can be obtained as a chain of simple linear operators. Synthetic tests illustrate how these operators work and also that scattered pressure field is consistent with the theoretical radiation patterns. These consistent linearized wave-equation operators are important when applying a gradient-based inverse scheme in the context of a multi-parameter FWI.

ACKNOWLEDGEMENT

We thank Joseph Stefani and Mark Meadows for the useful discussions and comments on the non-constant density propagation and inversion.

APPENDIX

In this Appendix we derive most of the results used in the paper.

Derive the acoustic wave equation from elastodynamic equations

First, it is important to derive the acoustic wave equation in non-constant density media such that we fully understand the approximations made when considering acoustic propagation. From Newton's second law of motion we can write:

$$\begin{aligned}
\rho(\mathbf{r})\frac{\partial^2 u_x(\mathbf{r},t)}{\partial t^2} &= \frac{\partial \tau_{xx}(\mathbf{r},t)}{\partial x} + \frac{\partial \tau_{xy}(\mathbf{r},t)}{\partial y} + \frac{\partial \tau_{xz}(\mathbf{r},t)}{\partial z} + f_x(\mathbf{r},t),\\
\rho(\mathbf{r})\frac{\partial^2 u_y(\mathbf{r},t)}{\partial t^2} &= \frac{\partial \tau_{yx}(\mathbf{r},t)}{\partial x} + \frac{\partial \tau_{yy}(\mathbf{r},t)}{\partial y} + \frac{\partial \tau_{yz}(\mathbf{r},t)}{\partial z} + f_y(\mathbf{r},t),\\
\rho(\mathbf{r})\frac{\partial^2 u_z(\mathbf{r},t)}{\partial t^2} &= \frac{\partial \tau_{zx}(\mathbf{r},t)}{\partial x} + \frac{\partial \tau_{zy}(\mathbf{r},t)}{\partial y} + \frac{\partial \tau_{zz}(\mathbf{r},t)}{\partial z} + f_z(\mathbf{r},t),
\end{aligned} \tag{29}$$

where $\rho(\mathbf{r})$ is the density of the medium, $u_i(\mathbf{r},t)$ is the particle displacement in the i direction, $\tau_{ii}(\mathbf{r},t)$ and τ_{ij} are the normal and shear stresses applied to the orthogonal plane to i direction, respectively; and $f_i(\mathbf{r},t)$ represents the external forces along the i direction. Because we are considering only acoustic isotropic media the shear stresses can be neglected, and the normal stresses are given by the following constitutive relations:

$$\begin{aligned}
\tau_{xx}(\mathbf{r},t) &= \lambda(\mathbf{r})\left(\frac{\partial u_x(\mathbf{r},t)}{\partial x} + \frac{\partial u_y(\mathbf{r},t)}{\partial y} + \frac{\partial u_z(\mathbf{r},t)}{\partial z}\right) + 2\mu(\mathbf{r})\frac{\partial u_x(\mathbf{r},t)}{\partial x},\\
\tau_{yy}(\mathbf{r},t) &= \lambda(\mathbf{r})\left(\frac{\partial u_x(\mathbf{r},t)}{\partial x} + \frac{\partial u_y(\mathbf{r},t)}{\partial y} + \frac{\partial u_z(\mathbf{r},t)}{\partial z}\right) + 2\mu(\mathbf{r})\frac{\partial u_y(\mathbf{r},t)}{\partial y},\\
\tau_{zz}(\mathbf{r},t) &= \lambda(\mathbf{r})\left(\frac{\partial u_x(\mathbf{r},t)}{\partial x} + \frac{\partial u_y(\mathbf{r},t)}{\partial y} + \frac{\partial u_z(\mathbf{r},t)}{\partial z}\right) + 2\mu(\mathbf{r})\frac{\partial u_z(\mathbf{r},t)}{\partial z},
\end{aligned} \tag{30}$$

where $\lambda(\mathbf{r})$ is the Lamé's first parameter, and $\mu(\mathbf{r})$ is the Lamé's second parameter also known as shear modulus of the medium.

If we consider isotropic stress fields the pressure field is equal to each normal stress, and:

$$p(\mathbf{r},t) = \tau_{xx}(\mathbf{r},t) = \tau_{yy}(\mathbf{r},t) = \tau_{zz}(\mathbf{r},t). \tag{31}$$

Therefore, the scalar pressure field can be expressed as:

$$p(\mathbf{r},t) = \frac{1}{3}(\tau_{xx}(\mathbf{r},t) + \tau_{yy}(\mathbf{r},t) + \tau_{zz}(\mathbf{r},t)) + p_s(\mathbf{r},t), \tag{32}$$

where $p_s(\mathbf{r},t)$ is an external pressure source; therefore, using Equation 30, we have:

$$p(\mathbf{r},t) = \left(\lambda(\mathbf{r}) + \frac{2}{3}\mu(\mathbf{r})\right)\nabla\cdot\mathbf{u}(\mathbf{r},t) + p_s(\mathbf{r},t) = K(\mathbf{r})\nabla\cdot\mathbf{u}(\mathbf{r},t) + p_s(\mathbf{r},t), \tag{33}$$

where we have introduced a new elastic property of the medium $K(\mathbf{r})$ called bulk modulus or compressibility, which measures the medium's resistance to uniform compression. Because $K(\mathbf{r})$ is time independent, taking the second-order time derivative of equation 33, and substituting Newton's second law (i.e., Equation 29) we can write:

$$\frac{\partial^2 p(\mathbf{r},t)}{\partial t^2} = K(\mathbf{r})\nabla\cdot\frac{1}{\rho(\mathbf{r})}\nabla p(\mathbf{r},t) + \frac{\partial^2 p_s(\mathbf{r},t)}{\partial t^2} + \nabla\cdot\frac{\mathbf{f}(\mathbf{r},t)}{\rho(\mathbf{r})}. \tag{34}$$

Reordering Equation 34 we can express it as the common wave equation:

$$\left[\frac{1}{K(\mathbf{r})}\frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho(\mathbf{r})}\nabla\right] p(\mathbf{r},t) = s(\mathbf{r},t), \tag{35}$$

where $s(\mathbf{r},t)$ represents both the volume injection and external force per unit volume (i.e., last two right-side terms in Equation 34, respectively). From this last relation, we can easily derive the well-known acoustic isotropic constant-density wave equation.

Born approximation and adjoint conditions for non-constant density media

Before deriving the Born approximation and the adjoint conditions for non-constant density, we first introduce the Green's function of Equation 35:

$$\left[\frac{1}{K(\mathbf{r})}\frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho(\mathbf{r})}\nabla\right] g(\mathbf{r},t,\mathbf{r}',t';K,\rho) = \delta(\mathbf{r}-\mathbf{r}')\delta(t-t'). \tag{36}$$

Considering time invariant medium's properties the solution of the acoustic non-constant density wave equation can be expressed as:

$$p(\mathbf{r},t) = \int g(\mathbf{r},t,\mathbf{r}';K,\rho) * s(\mathbf{r}',t)d\mathbf{r}', \tag{37}$$

where $*$ denotes time convolution. During active seismic experiments sources can be considered as points in space with a given source signature $w(t)$ (i.e., $s(\mathbf{r},t) = w(t)\delta(\mathbf{r}-\mathbf{r}_s)$), therefore, the previous equation becomes:

$$p(\mathbf{r},t) = g(\mathbf{r},t,\mathbf{r}_s;K,\rho) * w(t). \tag{38}$$

Finally, because seismic receivers can be considered as recording points, the data are given by:

$$d(\mathbf{r}_g,t) = p(\mathbf{r},t)\delta(\mathbf{r}-\mathbf{r}_g). \tag{39}$$

To derive the Born approximation for non-constant density isotropic acoustic media we perturbe the background medium's properties $K_0(\mathbf{r})$ and $\rho_0(\mathbf{r})$:

$$K(\mathbf{r}) = K_0(\mathbf{r}) + \delta K(\mathbf{r}), \tag{40}$$

$$\rho(\mathbf{r}) = \rho_0(\mathbf{r}) + \delta\rho(\mathbf{r}), \tag{41}$$

where $\delta K(\mathbf{r})$ and $\delta\rho(\mathbf{r})$ are the bulk modulus and density perturbations respectively. Considering small perturbations respect to the background properties, we can assume the pressure field propagating in the perturbed medium is given by:

$$\left[\frac{1}{K_0(\mathbf{r})+\delta K(\mathbf{r})}\frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho_0(\mathbf{r})+\delta\rho(\mathbf{r})}\nabla\right] (p_0(\mathbf{r},t) + \delta p(\mathbf{r},t)) = s(\mathbf{r},t). \tag{42}$$

After expanding all terms in the left-side of the previous equation and dropping high-order terms, the perturbed pressure field $\delta p(\mathbf{r},t)$ is given by:

$$\left[\frac{1}{K_0(\mathbf{r})}\frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho_0(\mathbf{r})}\nabla\right] \delta p(\mathbf{r},t) = \frac{\delta K(\mathbf{r})}{K_0(\mathbf{r})^2}\frac{\partial^2 p_0(\mathbf{r},t)}{\partial t^2} - \nabla \cdot \frac{\delta\rho(\mathbf{r})}{\rho_0(\mathbf{r})^2}\nabla p_0(\mathbf{r},t), \tag{43}$$

where we have used:

$$\frac{1}{m_0(\mathbf{r})+\delta m(\mathbf{r})}=\frac{1}{m_0(\mathbf{r})}-\frac{\delta m(\mathbf{r})}{m_0(\mathbf{r})^2}, \tag{44}$$

and $p_0(\mathbf{r},t)$ corresponds to the pressure field propagated in the background medium:

$$\left[\frac{1}{K_0(\mathbf{r})}\frac{\partial^2}{\partial t^2}-\nabla\cdot\frac{1}{\rho_0(\mathbf{r})}\nabla\right]p_0(\mathbf{r},t)=s(\mathbf{r},t). \tag{45}$$

In Equation 43 the right-side two terms describe the scattered pressure caused by a compressibility and density perturbations, respectively; and they can be considered as secondary sources generated by the energy coming from the primary source $s(\mathbf{r},t)$. The Born modeled data are then obtained as:

$$\delta d(\mathbf{r}_g,t)=\delta p(\mathbf{r},t)\delta(\mathbf{r}-\mathbf{r}_g). \tag{46}$$

To simplify the discussion we continue it in the frequency domain and discretize both in time and space all the variables so far introduced. Given these two observations we can write Equation 38 as:

$$P(\mathbf{x}_i,\omega_k)=G(\mathbf{x}_i,\omega_k,\mathbf{x}_s;K,\rho)W(\omega_k)\quad i=1,\dots,M\quad k=1,\dots,N_t, \tag{47}$$

which provides the pressure field for all M points in space and N_t frequencies. Assuming Equation 39, seismic data are found as:

$$D(\mathbf{x}_g,\omega_k)=P(\mathbf{x}_i=\mathbf{x}_g,\omega_k)\quad g=1,\dots,N_g, \tag{48}$$

where N_g denotes the number of active receivers for a given source.

The solution of Equation 43 can be found by multiplying the unperturbed Green's function with the two secondary source terms for all the points in space:

$$\begin{gathered}\Delta P(\mathbf{x}_i,\omega_k)=\textstyle\sum_{j=1}^{M}G(\mathbf{x}_i,\omega_k,\mathbf{x}_j;K_0,\rho_0)\\ \left(\tfrac{\Delta K(\mathbf{x}_j)}{-K_0(\mathbf{x}_j)^2}\omega^2-\nabla\cdot\tfrac{\Delta\rho(\mathbf{x}_j)}{\rho_0(\mathbf{x}_j)^2}\nabla\right)G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)W(\omega_k)\end{gathered} \tag{49}$$

Because we want to find two separate linear kernels for compressibility and density perturbations we write the total perturbed pressure field as:

$$\Delta P(\mathbf{x}_i,\omega_k)=\Delta P_K(\mathbf{x}_i,\omega_k)+\Delta P_\rho(\mathbf{x}_i,\omega_k), \tag{50}$$

where

$$\Delta P_K(\mathbf{x}_i,\omega_k)=-\sum_{j=1}^{M}G(\mathbf{x}_i,\omega_k,\mathbf{x}_j;K_0,\rho_0)\omega^2G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)W(\omega_k)\frac{\Delta K(\mathbf{x}_j)}{K_0(\mathbf{x}_j)^2}, \tag{51}$$

$$\Delta P_\rho(\mathbf{x}_i,\omega_k)=-\sum_{j=1}^{M}G(\mathbf{x}_i,\omega_k,\mathbf{x}_j;K_0,\rho_0)\nabla\cdot\frac{\Delta\rho(\mathbf{x}_j)}{\rho_0(\mathbf{x}_j)^2}\nabla G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)W(\omega_k). \tag{52}$$

In Equation 52, the perturbed pressure field depends on the divergence of the product of the density perturbation and gradient of the background pressure field. This operation is linear;

however, it contains multiple terms that can be eliminated under certain assumptions. We can rewrite this equation using the following relation:

$$\psi\nabla\cdot(\mathbf{v}) = -(\nabla\psi)\cdot\mathbf{v} + \nabla\cdot(\psi\mathbf{v}), \tag{53}$$

as:

$$\begin{gathered}\Delta P_\rho(\mathbf{x}_i,\omega_k,\mathbf{x}_s;K_0,\rho_0) = \\ \sum_{j=1}^{M}\frac{\Delta\rho(\mathbf{x}_j)}{\rho_0(\mathbf{x}_j)^2}\nabla G(\mathbf{x}_i,\omega_k,\mathbf{x}_j;K_0,\rho_0)\cdot\nabla G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)W(\omega_k) \\ -\sum_{j=1}^{M}\nabla\cdot\frac{\Delta\rho(\mathbf{x}_j)}{\rho_0(\mathbf{x}_j)^2}G(\mathbf{x}_i,\omega_k,\mathbf{x}_j;K_0,\rho_0)\nabla G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)W(\omega_k).\end{gathered} \tag{54}$$

Using the divergence theorem, the second right-side term of Equation 54 can be written as a surface integral along the medium's boundaries; assuming homogeneous conditions or only internal density perturbations this term vanishes. Therefore, knowing that we are recording data only at the surface (Equation 46), we can write two linear kernels respect to the medium perturbations as:

$$\Delta D_K(\mathbf{x}_g,\omega_k) = \sum_{j=1}^{M}\left[\frac{\partial\Delta P_K(\mathbf{x}_g,\omega_k)}{\partial\Delta K(\mathbf{x}_j)}\right]\Delta K(\mathbf{x}_j), \tag{55}$$

$$\Delta D_\rho(\mathbf{x}_g,\omega_k) = \sum_{j=1}^{M}\left[\frac{\partial\Delta P_\rho(\mathbf{x}_g,\omega_k)}{\partial\Delta\rho(\mathbf{x}_j)}\right]\Delta\rho(\mathbf{x}_j), \tag{56}$$

where the total data perturbation of Equation 46 is given by:

$$\Delta D(\mathbf{x}_g,\omega_k) = \Delta D_K(\mathbf{x}_g,\omega_k) + \Delta D_\rho(\mathbf{x}_g,\omega_k). \tag{57}$$

The Born operator is applied to model perturbations and is returning data perturbation $\Delta D(\mathbf{x}_g,\omega_k)$ for all frequencies N_t and receivers N_g. Therefore, the adjoint of it is applied to data perturbation and provides back model perturbations, that mathematically corresponds to:

$$\Delta\tilde{K}(\mathbf{x}_j) = \sum_{k=1}^{N_t}\sum_{i=1}^{N_d}\left[\frac{\partial\Delta P_K(\mathbf{x}_g,\omega_k)}{\partial\Delta K(\mathbf{x}_j)}\right]^*\Delta D(\mathbf{x}_i,\omega_k), \tag{58}$$

$$\Delta\tilde{\rho}(\mathbf{x}_j) = \sum_{k=1}^{N_t}\sum_{i=1}^{N_d}\left[\frac{\partial\Delta P_\rho(\mathbf{x}_g,\omega_k)}{\partial\Delta\rho(\mathbf{x}_j)}\right]^*\Delta D(\mathbf{x}_i,\omega_k). \tag{59}$$

Substituting Equations 51 and 54 into the two previous relations we have:

$$\begin{gathered}\Delta\tilde{K}(\mathbf{x}_j) = -\frac{1}{K_0(\mathbf{x}_j)^2}\sum_{k=1}^{N_t}\sum_{i=1}^{N_d}\omega^2 W(\omega_k)^* G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)^* \\ G(\mathbf{x}_j,\omega_k,\mathbf{x}_i;K_0,\rho_0)^*\Delta D(\mathbf{x}_i,\omega_k)\end{gathered} \tag{60}$$

$$\begin{gathered}\Delta\tilde{\rho}(\mathbf{x}_j) = \frac{1}{\rho_0(\mathbf{x}_j)^2}\sum_{k=1}^{N_t}\sum_{i=1}^{N_d}\left[\nabla G(\mathbf{x}_j,\omega_k,\mathbf{x}_s;K_0,\rho_0)W(\omega_k)\right]^*\cdot \\ \nabla G(\mathbf{x}_j,\omega_k,\mathbf{x}_i;K_0,\rho_0)^*\Delta D(\mathbf{x}_i,\omega_k),\end{gathered} \tag{61}$$

where we have used the reciprocity property of the Green's function:

$$G(\mathbf{x}_i,\omega_k,\mathbf{x}_j;K_0,\rho_0) = G(\mathbf{x}_j,\omega_k,\mathbf{x}_i;K_0,\rho_0). \tag{62}$$

Equations 60 and 61 in the time and continuous domain are:

$$\delta\tilde{K}(\mathbf{r}) = \frac{1}{K_0(\mathbf{r})^2}\sum_{i=1}^{N_d}\left[\frac{\partial^2 p_0(\mathbf{r})}{\partial t^2} \star \delta p_i'(\mathbf{r})\right](0), \tag{63}$$

$$\delta\tilde{\rho}(\mathbf{r}) = \frac{1}{\rho_0(\mathbf{r})^2}\sum_{i=1}^{N_d}\left[\nabla p_0(\mathbf{r}) \underset{\cdot}{\star} \nabla \delta p_i'(\mathbf{r})\right](0), \tag{64}$$

where $[f \star g](0)$ and $[f \underset{\cdot}{\star} g](0)$ denote the zero-lag crosscorrelation and the zero-lag crosscorrelation of the scalar product of two functions, respectively; $p_0(\mathbf{r},t)$ is given by Equation 38, and $\delta p_i'(\mathbf{r},t)$ is obtained as:

$$\delta p_i'(\mathbf{r},t) = \tilde{g}(\mathbf{r},t,\mathbf{r}_i;K,\rho) * \delta d(\mathbf{r}_i,t), \tag{65}$$

in which we are convolving the anticasual Green's function with the data perturbation. Because we are considering a single source, we can rewrite Equations 63 and 64 as:

$$\delta\tilde{K}(\mathbf{r}) = \frac{1}{K_0(\mathbf{r})^2}\left[\frac{\partial^2 p_0(\mathbf{r})}{\partial t^2} \star \delta p'(\mathbf{r})\right](0), \tag{66}$$

$$\delta\tilde{\rho}(\mathbf{r}) = \frac{1}{\rho_0(\mathbf{r})^2}\left[\nabla p_0(\mathbf{r}) \underset{\cdot}{\star} \nabla \delta p'(\mathbf{r})\right](0), \tag{67}$$

where:

$$\delta p'(\mathbf{r},t) = \sum_{i=1}^{N_d}\tilde{g}(\mathbf{r},t,\mathbf{r}_i;K,\rho) * \delta d(\mathbf{r}_i,t). \tag{68}$$

Property of staggered-grid operators

Staggered-grid derivative operators have a useful property that can be used when applying their adjoints. We define the forward and backward first-order derivative with second-order accuracy as:

$$\left.\frac{df(x)}{dx}\right|_{\mathbf{x}_{i+\frac{1}{2}}} = \frac{f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i)}{\Delta x}, \tag{69}$$

$$\left.\frac{df(x)}{dx}\right|_{\mathbf{x}_{i-\frac{1}{2}}} = \frac{f(\mathbf{x}_{i}) - f(\mathbf{x}_{i-1})}{\Delta x}, \tag{70}$$

respectively. If we assume the function $f(x)$ vanishes outside the computational domain, we can write the operator matrix $\mathbf{D}^+$ of Equation 69 using transient convolution (Claerbout, 2014) as:

$$\mathbf{D}^+ = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 \end{bmatrix}. \tag{71}$$

Taking the adjoint of this operator we have:

$$\mathbf{D}^{+*} = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix} =$$

$$- \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix} = -\mathbf{D}^{-}, \tag{72}$$

where $\mathbf{D}^{-}$ denotes the operator matrix of Equation 70. From relation 72 follows that:

$$\mathbf{D}^{-*} = -\mathbf{D}^{+}. \tag{73}$$

These equalities hold also in case higher-accuracy finite-difference operators are used.

REFERENCES

Aki, K. and P. G. Richards, 2002, Quantitative seismology, volume **1**.

Almomin, A., 2013, Accurate implementation of two-way wave-equation operators: SEP-Report, **149**, 281–288.

Claerbout, J. F., 2014, Geophysical image estimation by examples.

Innanen, K. A., 2014, Reconciling seismic avo and precritical reflection fwi–issues in multi-parameter gradient-based updating: Presented at the 76th EAGE Conference and Exhibition, European Association of Geoscientists and Engineers.

Israeli, M. and S. A. Orszag, 1981, Approximation of radiation boundary conditions: Journal of Computational Physics, **41**, 115–135.

Ji, J., 2009, An exact adjoint operation pair in time extrapolation and its application in least-squares reverse-time migration: Geophysics, **74**, H27–H33.

Mattsson, K., 2012, Summation by parts operators for finite difference approximations of second-derivatives with variable coefficients: Journal of Scientific Computing, **51**, 650–682.

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

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

Reverse-time migration: Comparing three numerical solvers

Alejandro Cabrales-Vargas

ABSTRACT

I compare the results of three numerical schemes for seismic modeling and reverse-time migration: the rapid expansion method, the Lax-Wendroff method, and the pseudo-analytical method. The rapid expansion method uses coarse time steps without developing instabilities, but it can present frequency dispersion when using coarse grids. Moreover, its implementation is difficult. The Lax-Wendroff method can avoid such limitations at the expense of time step refinement, thereby more Laplacian computations. Both methods allow the representation of the space derivatives either in the spatial domain (finite differences method) or in the Fourier domain (pseudospectral method). The pseudo-analytical method offers an accurate solution and easy implementation, but it is restricted to the Fourier domain.

INTRODUCTION

During the last decade, reverse-time migration (RTM) has been adopted as the ultimate solution for imaging very complex geology. Although geoscientists have known about RTM potential for 30 years (Baysal et al., 1983; Kosloff and Baysal, 1983; McMechan, 1983; Gazdag and Carrizo, 1986), it only recently became affordable. However, it is still computationally intensive, so the search of cheaper and more accurate implementations of the two-way wave equation continues. This fact is particularly crucial when thinking of forward and reverse-time propagation as the main engines behind state-of-the-art iterative solutions, such as least-squares migration and full-waveform inversion.

Traditionally, second-order time derivatives in the wave equation have been approximated using second-order finite differences (Baysal et al., 1983; Kosloff and Baysal, 1983; Gazdag and Carrizo, 1986; Dablain, 1986). This scheme is easy to implement and conceptually simple. However, for the sake of stability the Courant-Friedrichs-Lewy (CFL) criterion usually requires time step refinements (Dablain, 1986). Computations increase in proportion to the interpolation factor. Making the grid coarser solves the problem, but reducing the maximum frequency that can be handled without dispersion. The balance between stability and numerical dispersion for the second-order approximation is stringent because it originates from a Taylor series expansion with only two terms, thus assuming small time steps in relation to the grid size.

One obvious solution to relax the stability threshold is to implement better approximations that tolerate less refinements of time steps. In the limit, we may want an approximation that does not require refinement altogether. Incorporating more terms to the approximation in principle achieves this goal, but high-order times derivatives appear. Fortunately, we can compute such time derivatives by means of second-order spatial derivatives according to the wave equation. The Lax-Wendroff method (LWM) (Dablain, 1986; Chen, 2007; Zhang et al., 2007; Pestana and Stoffa, 2010) makes use of this technique approximating the fourth-order time derivative by cascading the Laplacian operator (multiplied by the medium velocity) twice. On the contrary, the rapid expansion method (REM) (Tal-Ezer et al., 1987; Kosloff et al., 1989; Jastram and Behle, 1991; Tessmer, 2011; Stoffa and Pestana, 2009; Pestana and Stoffa, 2010; Araujo et al., 2014) takes a further step, using an expansion in Chebyshev polynomials and cascading as many terms as required to achieve convergence. The number of terms is tied to the time step, so that any time step can be used when enough terms are incorporated. The spatial derivatives in LWM and REM can be computed using finite differences (Jastram and Behle, 1991) or the pseudospectral method (Tal-Ezer et al., 1987; Kosloff et al., 1989; Jastram and Behle, 1991; Tessmer, 2011; Stoffa and Pestana, 2009; Pestana and Stoffa, 2010; Araujo et al., 2014)

A relatively new solution is the pseudo-analytical method (PAM), originally proposed by Etgen and Brandsberg-Dahl (2009), and later modified by dos Santos and Pestana (2012). The original propose uses so-called *pseudo-Laplacians* to propagate constant-velocity wavefields, which are subsequently interpolated to incorporate velocity variations. In their modified version, dos Santos and Pestana (2012) compute a *pseudo-Laplacian* that ”pivots” around a reference constant velocity (usually the minimum or the maximum in the model), and introduces velocity variations by means of perturbations of such reference velocity. The advantage of this method is that it is almost as simple as the second-order time marching scheme, but keeping stability at coarser time steps, and almost free of frequency dispersion. It is considerably easier to implement than the REM. One drawback is that it only can be implemented in the Fourier domain.

In the following section I briefly present the theory of the three methods. Next, I compare their performance in modeling and migration applied to a synthetic, normal fault model. Finally, I present the conclusions of this report.

LAX-WENDROFF METHOD

Given an exploding source, $\mathbf{f}(\mathbf{x},t)$, ignited in an acoustic, constant density medium, the propagation wavefield satisfies the 2D acoustic wave equation:

$$\frac{\partial^2 \mathbf{u}}{\partial t^2} = -\mathbf{L}^2\mathbf{u} + \mathbf{f}, \tag{1}$$

where $\mathbf{u}(\mathbf{x},t)$ is the displacement field, and $-\mathbf{L}^2 = v(x,z)^2\nabla^2$ constitutes the Laplacian scaled by the squared of the interval velocity. The corresponding expressions in the physical domain and in the Fourier domain are, respectively,

$$-\mathbf{L}^2 = v(x,z)^2\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right), \tag{2}$$

and

$$-\mathbf{L}^2 = -v(x,z)^2(\mathbf{k_x}^2 + \mathbf{k_z}^2), \tag{3}$$

where $\mathbf{k_x}$ and $\mathbf{k_z}$ are the spatial wavenumbers. Note that in the last equation we mixed physical domain variables with Fourier domain variables. While formally violating the assumption of constant velocity, in practice this artifice eases maneuvering the equations, and the conclusions extracted are fundamentally correct.

The solution the homogeneous version of Equation 1 (source term equals to zero) subject to non-zero initial conditions is

$$\frac{\partial^2 \mathbf{u}}{\partial t^2} = -\mathbf{L}^2\mathbf{u}, \tag{4}$$

$$\mathbf{u}(t=0) = \mathbf{u}(0), \frac{\partial \mathbf{u}(t=0)}{\partial t} = \dot{\mathbf{u}}(0), \tag{5}$$

where $\mathbf{u}(0)$ and $\dot{\mathbf{u}}(0)$ constitute the initial displacement field and its first time derivative, respectively. Thus, the formal solution of Equation 4 for forward propagation in time is

$$\mathbf{u}(t) = \mathbf{u}(0)\cos(\mathbf{L}t) + \dot{\mathbf{u}}(0)\frac{\sin(\mathbf{L}t)}{\mathbf{L}}. \tag{6}$$

Likewise, the solution for backward propagation in time is

$$\mathbf{u}(-t) = \mathbf{u}(0)\cos(\mathbf{L}t) - \dot{\mathbf{u}}(0)\frac{\sin(\mathbf{L}t)}{\mathbf{L}}. \tag{7}$$

Adding Equation 6 to Equation 7 eliminates the derivative of the initial displacement:

$$\mathbf{u}(t) + \mathbf{u}(-t) = 2\cos(\mathbf{L}t)\mathbf{u}(0). \tag{8}$$

Using Taylor series expansion to approximate the cosine operator:

$$\mathbf{u}(t) + \mathbf{u}(-t) = \left[2 + t^2\mathbf{L}^2 + \frac{t^4}{12}\mathbf{L}^4 + ...\right]\mathbf{u}(0). \tag{9}$$

Finally, replacing the reference time, $t = 0$, for any time, t, and propagating the solution in time steps, Δt, we obtain:

$$\mathbf{u}(t+\Delta t) + \mathbf{u}(t-\Delta t) = \left[2 + \Delta t^2\mathbf{L}^2 + \frac{\Delta t^4}{12}\mathbf{L}^4\right]\mathbf{u}(t). \tag{10}$$

In Equation 10 I have truncated the series beyond the fourth-order term, therefore it constitutes a fourth-order approximation of the second derivative in time. Equation 10 is the classical solution of the acoustic wave equation using the Law-Wendroff method (Dablain, 1986; Chen, 2007). Operator $\mathbf{L}^2$ can be implemented in the Fourier domain (pseudospectral) or with finite differences (Equations 2 and 3, respectively).

RAPID EXPANSION METHOD: ONE-STEP APPROACH

The formal solution of the acoustic wave equation (Equation 1) subject to zero initial conditions is (Tal-Ezer et al., 1987; Kosloff et al., 1989)

$$\mathbf{u}(\mathbf{x}, t) = \int_0^t \frac{\sin(\mathbf{L}(t-\tau))}{\mathbf{L}}\mathbf{f}(\mathbf{x}, \tau)d\tau. \tag{11}$$

The numerical approximation to Equation 11 when the source is separable, $\mathbf{f}(\mathbf{x},t) = \mathbf{g}(\mathbf{x})s(t)$, is

$$\mathbf{u}(\mathbf{x},t) = 2\sum_{k=0}^{M} b_{2k+1}(t)\frac{R}{i\mathbf{L}}Q_{2k+1}\left(\frac{i\mathbf{L}}{R}\right)\mathbf{g}(\mathbf{x}), \tag{12}$$

where Q_k are modified Chebyshev polynomials of first kind and order k (the coefficients are positive, whereas in original polynomials they have alternating signs), and R is a scalar larger than the largest eigenvalue in $\mathbf{L}$, therefore keeping the norm of the polynomials argument less than 1. Function $\mathbf{g}(\mathbf{x})$ is an impulse spike in space that indicates the position of the source, and

$$b_k(t) = \frac{1}{R}\int_0^t J_k(\tau R)s(t-\tau)d\tau, \tag{13}$$

where J_k are Bessel functions of first kind and order k, and $s(t)$ is the source signature. Note that only odd k values are required in the series. The value of R can be calculated as

$$R = \frac{1.1\pi V_{max}\sqrt{2}}{min(\Delta x, \Delta z)}. \tag{14}$$

Summation 12 converges for $M > Rt_{max}$ expansion terms. Given recording times of typical seismic surveys, subsurface velocities, and grid sizes, the value of M would be the order of thousands. This fact discourages the explicit computation of the Chebyshev polynomials coefficients. Instead, the polynomials are recursively calculated for the source position function $\mathbf{g}(\mathbf{x})$, initializing with

$$\frac{R}{i\mathbf{L}}Q_1\left(\frac{i\mathbf{L}}{R}\right)\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), \tag{15}$$

$$\frac{R}{i\mathbf{L}}Q_3\left(\frac{i\mathbf{L}}{R}\right)\mathbf{g}(\mathbf{x}) = 3\mathbf{g}(\mathbf{x}) - 4\left(\frac{\mathbf{L}}{R}\right)^2\mathbf{g}(\mathbf{x}), \tag{16}$$

and calculating successive terms according to

$$\frac{R}{i\mathbf{L}}Q_{2k+1}\left(\frac{i\mathbf{L}}{R}\right)\mathbf{g}(\mathbf{x}) = 2\left[1 - 2\left(\frac{\mathbf{L}}{R}\right)^2\right]\frac{R}{i\mathbf{L}}Q_{2k-1}\left(\frac{i\mathbf{L}}{R}\right)\mathbf{g}(\mathbf{x}) - \frac{R}{i\mathbf{L}}Q_{2k-3}\left(\frac{i\mathbf{L}}{R}\right)\mathbf{g}(\mathbf{x}). \tag{17}$$

The factor $\frac{R}{i\mathbf{L}}$ makes the odd terms of the series contain only even powers of $\frac{i\mathbf{L}}{R}$, yielding flipping signs instead of imaginary numbers.

The computation of the $b_k(t)$ coefficients is time consuming. Fortunately, they are independent of the source position, which means they are computated once for all the shots in a survey, or everytime the source signature significantly varies with position.

As mentioned before, one-step REM approach assumes separability of the source in two independent time and space functions. Such assumption stands for forward propagation of the source wavefield, but breaks down for backward propagation of the receiver wavefield. We can use the recursive implementation of the REM for this case.

RAPID EXPANSION METHOD: RECURSIVE APPROACH

Equation 8 can be approximated by expanding the cosine operator with Chebyshev polynomials:

$$\mathbf{u}(t) + \mathbf{u}(-t) = \sum_{k=0}^{M/2} C_{2k} J_{2k}(Rt) Q_{2k}\left(\frac{i\mathbf{L}}{R}\right)\mathbf{u}(0), \tag{18}$$

where coefficients $C_0 = 1$, and $C_k = 2$ for $k > 0$. The main differences between Equations 12 and 18 are the use of the Bessel functions instead of the b_k coefficients, and the even terms in the summation. The computation can be performed in a single step as shown before. Alternatively (Pestana and Stoffa, 2010; Tessmer, 2011), substituting smaller time steps, Δt, for the total time, t, and an arbitrary time, t, for the reference time, t =0, yields

$$\mathbf{u}(t+\Delta t) + \mathbf{u}(t-\Delta t) = \sum_{k=0}^{M} C_{2k} J_{2k}(R\Delta t) Q_{2k}\left(\frac{i\mathbf{L}}{R}\right)\mathbf{u}(t), \tag{19}$$

which can be solved recursively in time. The initial conditions are injected at each time step. The number of terms required to ensure convergence is $M > R\Delta t$, which is usually less than 20. Therefore, the explicit computation of the Chebyshev polynomials coefficients becomes affordable in recursive REM.

Now we re-order terms of Equation 19 in factors of $\left(\frac{\mathbf{L}}{R}\right)^2$:

$$\mathbf{u}(t+\Delta t) + \mathbf{u}(t-\Delta t) = \sum_{k=0}^{M} A_{2k}\left[\frac{i\mathbf{L}}{R}\right]^{2k}\mathbf{u}(t), \tag{20}$$

where

$$A_k = \sum_{l=k}^{M} C_l J_l(R\Delta t) q_{k,l} \tag{21}$$

and $q_{k,l}$ represents the l coefficient of the Chebyshev polynomial of order k.

Notice that Equation 20 represents a polynomial expansion in even powers of the Laplacian operator. Moreover, if we set $R = 1$, and instead of using Equation 21 we define coefficients A_k as

$$A_0 = 2;\ A_2 = 1;\ A_4 = \frac{1}{12};\ A_k = 0\ \forall\ k > 4, \tag{22}$$

we obtain the representation of the LWM (Equation 10).

PSEUDO-ANALYTICAL METHOD

Following the variant of the PAM proposed by dos Santos and Pestana (2012), we can use the time step marching version of Equation 8,

$$\mathbf{u}(t+\Delta t) + \mathbf{u}(t-\Delta t) = 2\cos(\mathbf{L}_0\Delta t)\mathbf{u}(t). \tag{23}$$

Here I substitute the operator $\mathbf{L}_0$ for $\mathbf{L}$ to indicate the use of a constant reference velocity, usually defined as the minimum or maximum velocity in the model (dos Santos and Pestana, 2012). The cosine term is recast using the well known trigonometric identity, $\cos(2\theta) = 1 - 2\sin^2(\theta)$, so we obtain

$$\mathbf{u}(t+\Delta t) - 2\mathbf{u}(t) + \mathbf{u}(t-\Delta t) = -2\sin^2\left(\frac{\mathbf{L}_0\Delta t}{2}\right)\mathbf{u}(t). \tag{24}$$

Multiplying and dividing by the appropriate terms, the sine function becomes a sinc function [$\text{sinc}(\theta) = \frac{\sin(\theta)}{\theta}$]:

$$\mathbf{u}(t+\Delta t) - 2\mathbf{u}(t) + \mathbf{u}(t-\Delta t) = -\Delta t^2\mathbf{L}_0^2\text{sinc}^2\left(\frac{\mathbf{L}_0\Delta t}{2}\right)\mathbf{u}(t). \tag{25}$$

In the limit as $\Delta t \to 0$, Equation 25 becomes the constant-velocity, second-order approximation, of the wave equation. The advantage of this scheme is the compensation for frequency-dispersion-related numerical errors (Etgen and Brandsberg-Dahl, 2009), otherwise present when using conventional second-order approximation.

The extension to variable velocity is performed by defining an auxiliary wavefield,

$$\mathbf{p}(\mathbf{x},t) = \alpha(\mathbf{x})\mathbf{u}(\mathbf{x},t), \tag{26}$$

where $\alpha(\mathbf{x}) = 1 - \frac{v_0^2}{v(\mathbf{x})^2}$ incorporates the velocity variation as a perturbation of the reference velocity, v_0. This auxiliary wavefield becomes a source term in the acoustic wave equation:

$$\frac{\partial^2\mathbf{u}}{\partial t^2} = -\mathbf{L}_0^2\mathbf{u} + \frac{\partial^2\mathbf{p}}{\partial t^2}. \tag{27}$$

Equation 27 becomes homogeneous by introducing another auxiliary field,

$$\mathbf{u}(\mathbf{x},t) = \mathbf{w}(\mathbf{x},t) + \mathbf{p}(\mathbf{x},t), \tag{28}$$

thus obtaining

$$\frac{\partial^2\mathbf{w}}{\partial t^2} = -\mathbf{L}_0^2(\mathbf{w}+\mathbf{p}). \tag{29}$$

Finally, we apply the expression for the pseudo-analytical method (Equation 25), to obtain

$$\mathbf{w}(t+\Delta t) - 2\mathbf{w}(t) + \mathbf{w}(t-\Delta t) = -\Delta t^2\mathbf{L}_0^2\text{sinc}^2\left(\frac{\mathbf{L}_0\Delta t}{2}\right)(\mathbf{w}(t)+\mathbf{p}(t)). \tag{30}$$

Equation 30 can be solved for wavefield $\mathbf{w}(\mathbf{x},t)$ like in the constant-velocity case. Function $\alpha(\mathbf{x},t)$ is precomputed, then $\mathbf{p}(\mathbf{x},t)$ is calculated to initialize the recursion in time. The original wavefield is recovered as $\mathbf{u}(\mathbf{x},t) = \mathbf{w}(\mathbf{x},t)/(1-\alpha(\mathbf{x},t))$.

METHODOLOGY AND IMPLEMENTATION

I test the methods discussed in this report modeling and migrating a simple two-bed, normal fault model of the subsurface. The shallow bed has a velocity gradient ranging from 2000 to 2500m/s approximately, and the deep bed has a constant velocity of 5000m/s (Figure 1). Despite its conspicuous structural simplicity, the velocity contrast is strong and the fault dip is about 60°, therefore making the analysis simple in rather challenging conditions. The gridsize is $\Delta x = 12.5$m and $\Delta z = 10$m, covering an extension of 4500m (The model begins at −500m). Therefore, I should be able to model frequencies up to 80Hz (based on the pseudospectral method). The dataset consists of 36 shots equally spaced every 100m. Each shot has 361 receivers spaced every 12.5m, which means that each one covers the model extension.

In order to make fair comparisons with the PAM, I prepared the LWM and the REM tests using the pseudospectral method. All the methods employ tapering boundary conditions (Cerjan et al., 1985). In the migrations tests I use wavefield separation in the imaging condition according to Liu et al. (2011).

In the modeling and RTM tests I employed two different Ricker wavelets as input source: a) dominant frequency of 15Hz and maximum frequency of 50Hz; b) dominant frequency of 50Hz, maximum frequency slightly above 80Hz (the maximum frequency theoretically handled).

For the implementation of the REM, I employed the one-step solution to compute the source wavefield (Equation 12), and the recursive solution for the receiver wavefield (Equation 19).

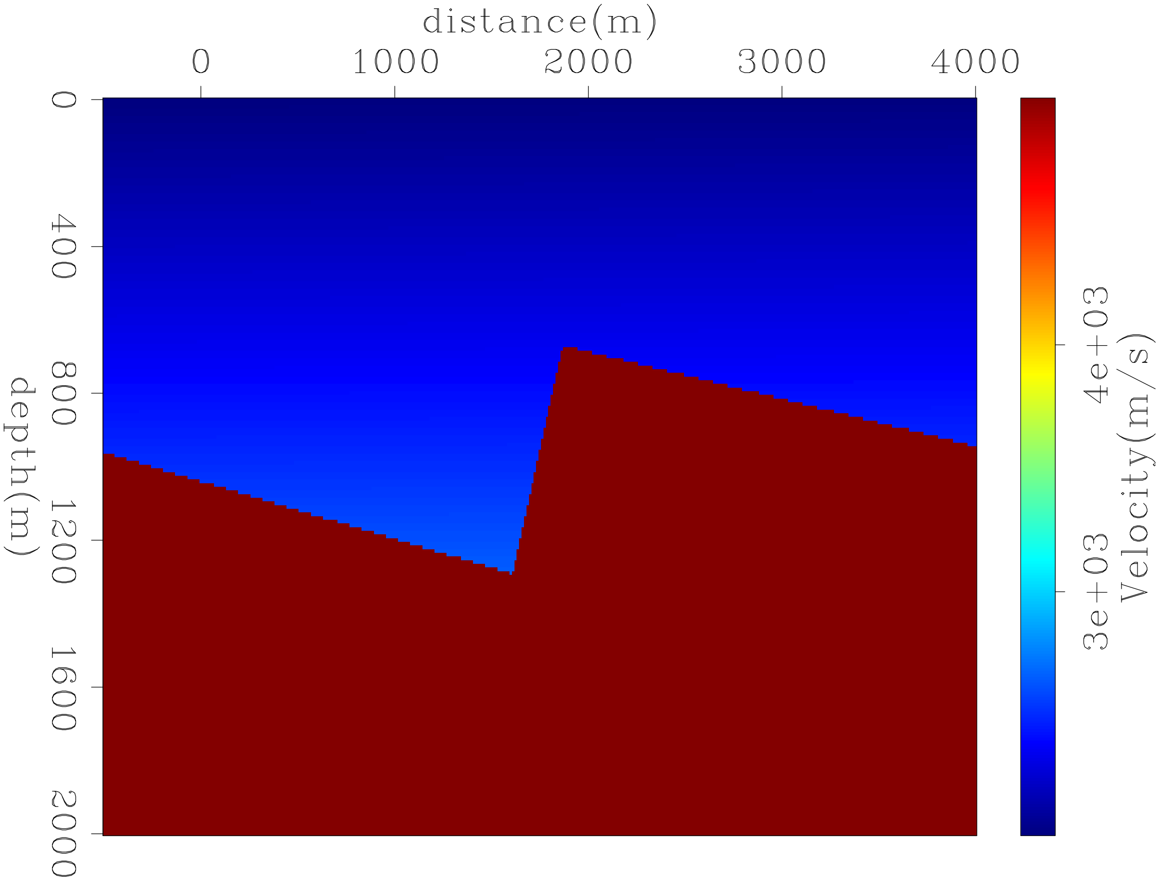


Figure 1: Normal-fault velocity model [**NR**] alejandro1/. Dip-Bed-vmodel

NUMERICAL RESULTS

Figure 2 shows the results of the modeling tests with the source wavelet a) in one time frame. We observe the reflection and transmision waves corresponding to the dipping velocity contact between positions 500m and 1800m, and between $1 - 1.6$ km. The diffraction originates at the cusp of the footwall block, partially reflecting back to the 2000m/s layer, whereas the other part transmits through the 5000m/s layer. The rest of the events correspond to attenuated waves that wrapped around the taper boundaries. Such events are mostly present in the Lax-Wendroff and the pseudo-analytical modeling tests, which are virtually identical. In the case of the rapid expansion method such artifacts were highly attenuated. I applied the same boundary conditions in the three methods, but while in Lax-Wendroof and pseudo-analytical methods the tapering is applied to the wavefields every time step, in the rapid expansion method such taper is applied to every Chebyshev polynomial evaluation in Equation 12, before multiplying by the $b_k(t)$ coefficients.

Figures 3 show the results of the tests using the source wavelet b), where the transmitted and reflected events contain higher frequency as expected. In this case we observe that the pseudo-analytical method begins to develop frequency dispersion. Notwithstanding that the stability condition is enforced, small frequency components beyond the theoretically maximum frequency are enough to cause such dispersion. Note the absence of this phenomenon in the Lax-Wendroff and the rapid expansion frames.

I finally present migration tests in Figures 4 and 5 using the source a) and b), respectively, to compute the corresponding source wavefields. It is interesting to notice the presence of low-wavenumber migration artifacts in the case of the Lax-Wendroff method and the rapid expansion method. They seem to be mostly related to the presence of the shap lateral velocity contrast introduced by the normal fault, and can constitute remnants of the RTM typical low-wavenumber numerical noise. Such noise obscures part of the fault plane. There are also high-wavenumber artifacts, very likely related to source undersampling. Both types of numerical artifacts are almost absent in the pseudo-analytical test, so the fault plane is nicely imaged. Moreover, the frequency dispersion observed when going to higher source frequencies is not harming the migrated image. The high reduction of high-wavenumber undersampling artifacts can be explained by the use of the sinc function of the pseudo-Laplacian operator, which is better behaved in the vicinity of the Nyquist wavenumbers. In the case of the reduction of low-wavenumber artifacts, one possible explanation is the computation of the Laplacian entirely in the reference velocity framework. As previously seen, only after such stage the velocity variation is introduced by means of the perturbation α.

CONCLUSIONS

In this report I compare three methods to perform modeling and reverse-time migration: the rapid expansion method, the pseudo-analytical method, and the Lax-Wendroff method. I test their performance on a normal fault model of the subsurface, with high velocity contrast.

The rapid expansion method offers stable solutions at arbitrary time steps, but it is cumbersome to code, and requires time functions dependent of Bessel functions that have to be precomputed, more than once in case of spatially-variable source signatures, therefore

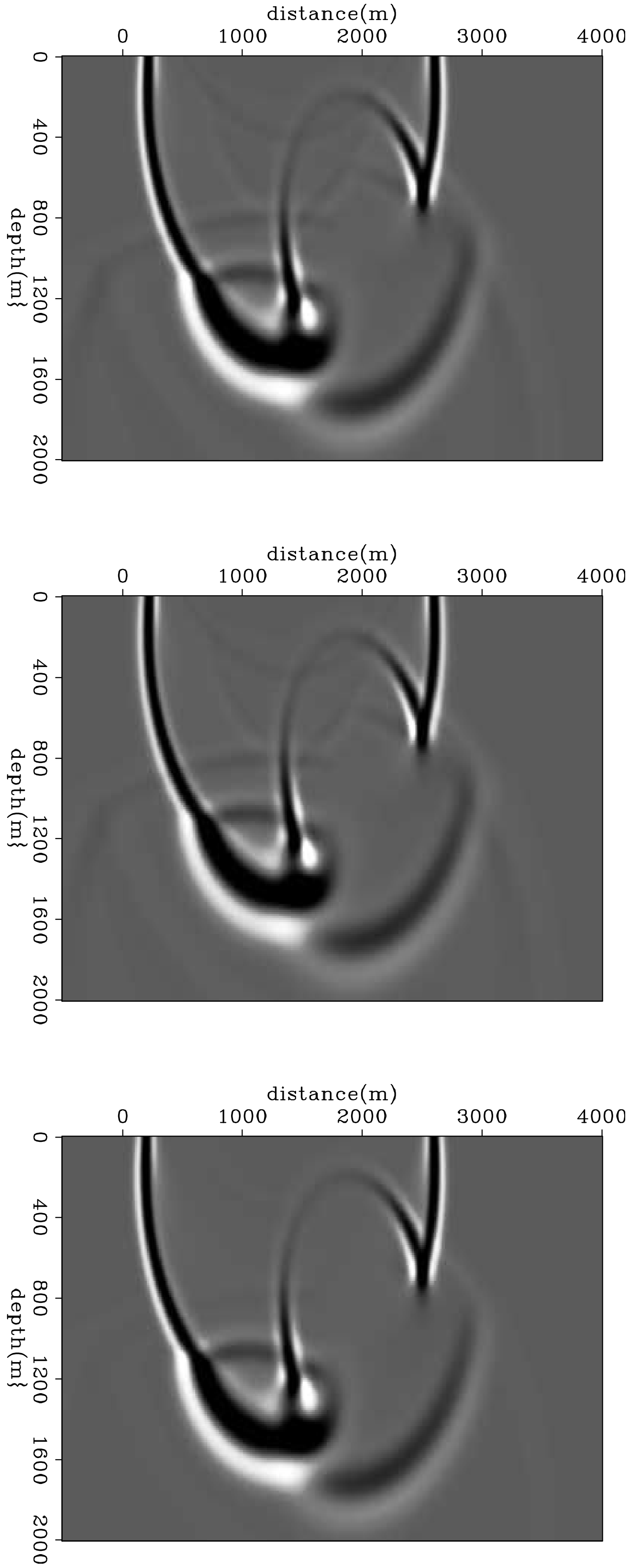


Figure 2: Single-source wavefront propagation using 15Hz Ricker wavelet. *Top*: Lax-Wendroff method; *Middle*: Pseudo-analytical method; *Bottom*: Rapid Expansion method [**CR**] alejandro1/. test1/Dip-Bed-fwdprops

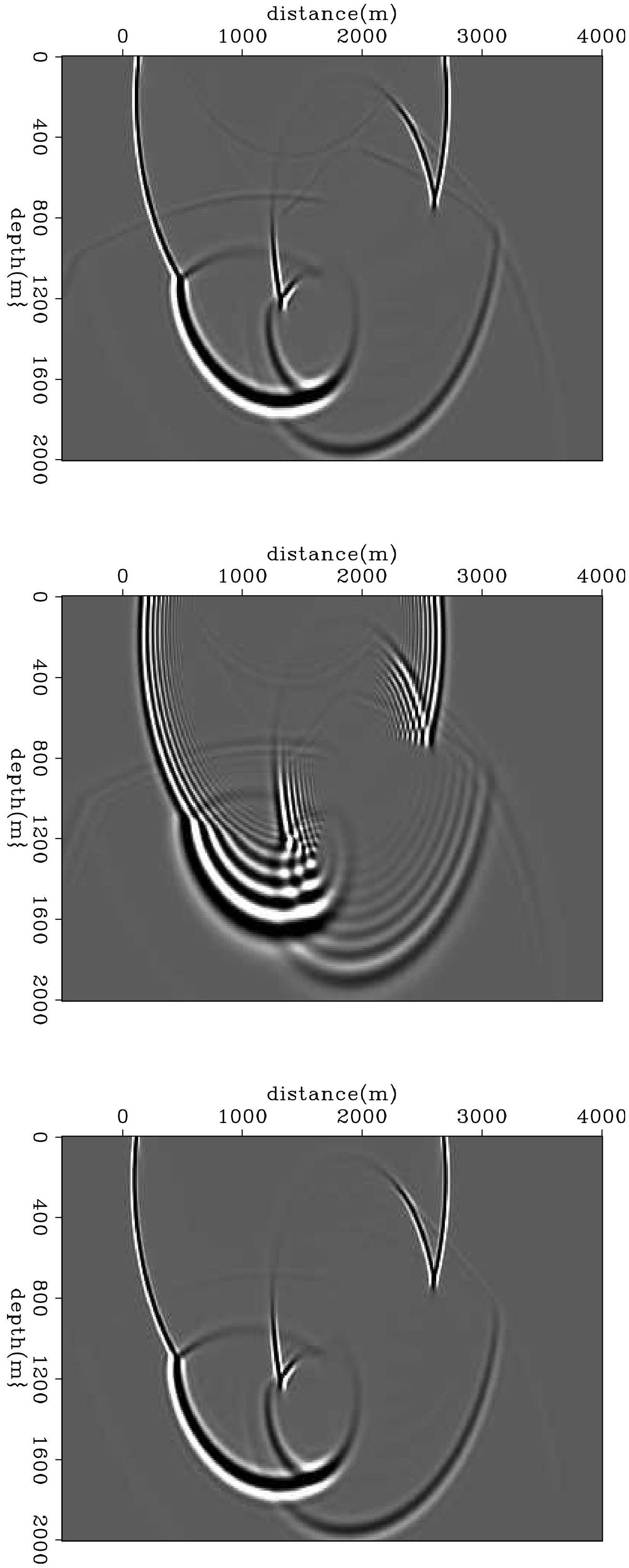


Figure 3: Single-source wavefront propagation using 50Hz Ricker wavelet. *Top*: Lax-Wendroff method; *Middle*: Pseudo-analytical method; *Bottom*: Rapid Expansion method [**CR**] alejandro1/. test2/Dip-Bed-fwdprops2

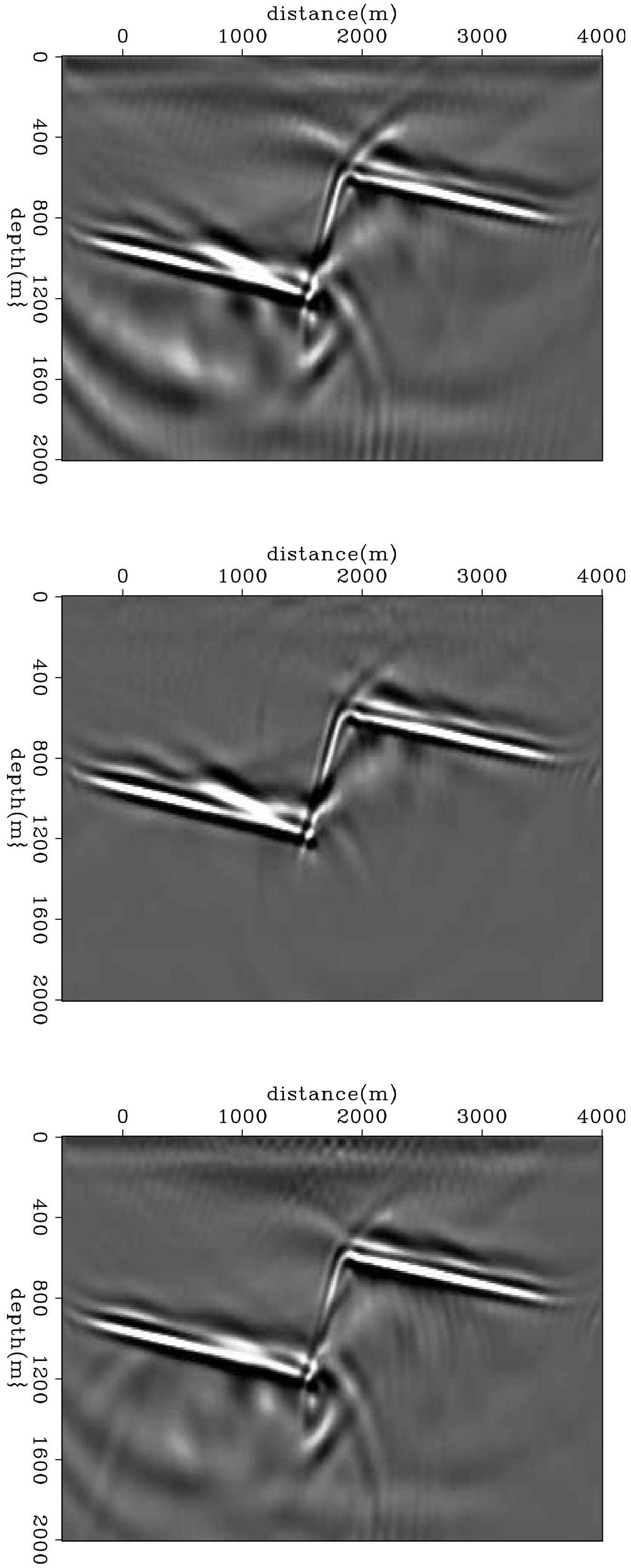


Figure 4: Reverse-time migration stack images using 15Hz Ricker wavelet. *Top*: Lax-Wendroff method; *Middle*: Pseudo-analytical method; *Bottom*: Rapid Expansion method [**CR**] alejandro1/. test1/Dip-Bed-rtm-stacks1

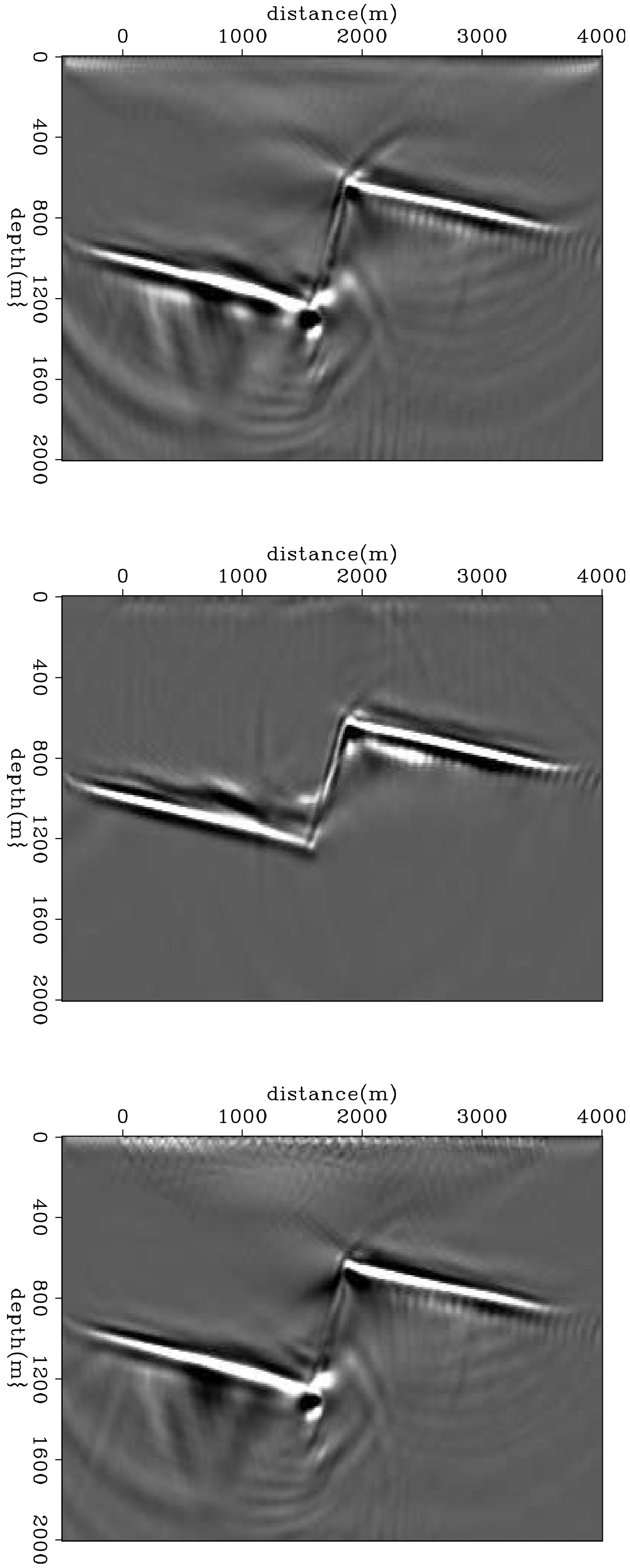


Figure 5: Reverse-time migration stack images using 50Hz Ricker wavelet. *Top*: Lax-Wendroff method; *Middle*: Pseudo-analytical method; *Bottom*: Rapid Expansion method [**CR**] alejandro1/. test2/Dip-Bed-rtm-stacks2

careful optimization strategies should be implemented to make the method fully efficient. Also, it is constrained to frequency dispersion bounds, which can limit its applicability for arbitrary time steps. The Lax-Wendroff method is an attractive alternative easier to implement, and as the rapid expansion method, it offers the flexibility to solve the Laplacian with finite differences. Nevertheless, the tests show that some low-frequency noise and aliasing noise can be present in both methods. The pseudo-analytical method, in the contrary, although is limited to the Fourier domain for the computation of the pseudo-Laplacian, succeeds in constructing a clean and accurate image, in spite of the frequency dispersion of the source wavefield. The better approximation around the Nyquist wavenumbers, and the computation of the pseudo-Laplacian using the constant reference velocity, can explain the reduction of high frequency noise, while the computation of pseudo-Laplacians using constant velocity can explain the reduction of low-frequency artifacts. More experiments are required to shed more light upon this behavior.

ACKNOWLEDGMENTS

I wish to thank Petroleos Mexicanos for the financial support, and the SEP sponsors and crew. Special thanks to Musa Maharramov, who suggested the comparisons presented in this report, and to Reynam Pestana, who kindly helped me clarify many issues about the rapid expansion method.

REFERENCES

Araujo, E., R. Pestana, and A. dos Santos, 2014, Symplectic scheme and the Poynting vector in reverse-time migration: Geophysics, **79**, no. 5, S163–S172.

Baysal, E., D. Kosloff, and J. Sherwood, 1983, Reverse time migration: Geophysics, **48**, 1514–1524.

Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef, 1985, A nonreflecting boundary condition for discrete acoustic and elastic wave equations: Geophysics, **50**, 705–708.

Chen, J., 2007, High-order time discretizations in seismic modeling: Geophysics, **72**, no. 5, SM115–SM122.

Dablain, M., 1986, The application of high-order differencing to the scalar wave equation: Geophysics, **51**, 54–66.

dos Santos, A. and R. Pestana, 2012, A pseudo-analytical method to solve the acoustic wave equation for modeling and RTM: SEG Technical Program Expanded Abstracts, 1–5.

Etgen, J. and S. Brandsberg-Dahl, 2009, The pseudo-analytical method: Application of pseudo-Laplacians to acoustic and acoustic anisotropic wave propagation: SEG Technical Program Expanded Abstracts, 2552–2556.

Gazdag, J. and E. Carrizo, 1986, On reverse-time migration: Geophysical Prospecting, **34**, 822–832.

Jastram, C. and A. Behle, 1991, Elastic modeling by finite difference and the rapid expansion method (REM): SEG Technical Program Expanded Abstracts, 1573–1576.

Kosloff, D. and E. Baysal, 1983, Migration with the full acoustic wave equation: Geophysics, **48**, 677–687.

Kosloff, D., A. Filho, E. Tessmer, and A. Behle, 1989, Numerical solution of the acoustic and elastic wave equation by a new rapid expansion method: Geophysical Prospecting, **37**, 383–394.

Liu, F., Z. Guanquan, S. Morton, and J. Leveille, 2011, An effective imaging condition for reverse-time migration using wavefield decomposition: Geophysics, **76**, no. 1, S29–S39.

McMechan, G., 1983, Migration by extrapolation of time-dependent boundary values: Geophysical Prospecting, **31**, 413–420.

Pestana, R. and P. Stoffa, 2010, Time evolution of the wave equation using rapid expansion method: Geophysics, **75**, no. 4, T121–T131.

Stoffa, P. and R. Pestana, 2009, Numerical solution of the acoustic wave equation by the rapid expansion method (REM)—A one step time evolution algorithm: SEG Technical Program Expanded Abstracts, 2672–2676.

Tal-Ezer, H., D. Kosloff, and Z. Koren, 1987, An accurate scheme for seismic forward modelling: Geophysical Prospecting, **35**, 479–490.

Tessmer, E., 2011, Using the rapid expansion method for accurate time-stepping in modeling and reverse-time migration: Geophysics, **76**, no. 4, S177–S185.

Zhang, Y., J. Sun, and S. Gray, 2007, Reverse-time migration: Amplitude and implementation issues: SEG Technical Program Expanded Abstracts, 2145–2149.

Using Mie scattering theory to debubble seismic airguns

Joseph Jennings and Shuki Ronen

ABSTRACT

Airgun signatures contain a main pulse and then a few bubble oscliations. A process called designature or debubble such signatures into a broad band pulse. We prefer to do as much as possible with deterministic designature and leave as little as possible to statistical deconvolution. Air gun manufacturers provide a library of signatures under various conditions. However, the conditions are not well known. Near field hydrophones record the airguns. However, the near field hydrophones record all airguns in the array, and their data are contaminated by waves that do not radiate to the far field. Current methods that estimate the contribution of each airgun to the far field require inverting for a large number of parameters. In this report, we propose another a deterministic deconvolution method based on theory from Mie scattering. Our method is less sensitive to near field noise and requires only seven parameters. Instead of a linear inversion with thousands of unknowns, we have a non linear inversion with a small number of unknowns. We have encouraging results that demonstrate the potential of using Mie scattering theory for deterministic debubbling.

INTRODUCTION

The most commonly used source in marine reflection seismic surveys is an airgun. In order to transmit acoustic waves into the subsurface, airguns release a pulse of high pressure air into the water. The pulse generates an acoustic wave, similar to a champagne bottle that is uncorked. The air expands to a bubble and loses pressure. When the pressure in the bubble gets to the ambient pressure of the water, the bubble starts slowing down its expansion. Due to the inertia of the water, the bubble continues to expand and the pressure inside the bubble is lower than ambient. At some time, the bubble reaches a maximal size and starts collapsing and the pressure in the bubble starts increasing. When the pressue in the bubble reaches ambient pressure the collapse starts slowing down. Some time later the bubble reaches a minimal size and starts expanding again. Bubbles from airgun oscillate many times as they go up. Typical bubble periods are 50 to 200 milliseconds. Typical depth of airguns under the sea surface is 5-15 meters. The time it takes the bubbles to reach the surface is much longer. The bubbles radiate acoustic waves as they oscillate. As energy is radiated out of the bubble the magnitude of the oscillations diminishes.

The acoustic waves that are radiated by the bubbles go in all directions. The upgoing waves reflect from the surface and follow the waves radiated down. The surface reflection is called the source ghost. The acoustic waves, including main pulse, bubble and ghosts, propagate down through the overburden, are reflected from targets, propagate up through the overburden, are usually ghosted again near the receiver, and eventually are recorded and become data. The convolutional model of a seismic trace is described by Figure 1. To find the reflectivity that best fits the data all filters must be undone. In this paper we focus on the coupling of the Source Force to the acoustic waves which we denote as Bubble. In practice, undoing the filters denoted as Bubble and Ghosts are bundled to what is called deconvolution. To show the effects of Bubble and Source Ghost we present data from a far field hydrophone in Figure 2(a). In this example all other effects are negligible. This figure shows the wavelets and the spectra of two shots at two different depths from the same airgun with the same volume and the same pressure. The Bubble and the Source Ghost filters depend on the depth. Ghosting also depends on the angle (not shown in Figure 2(a) in which both traces have the same angle).

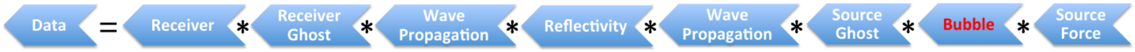


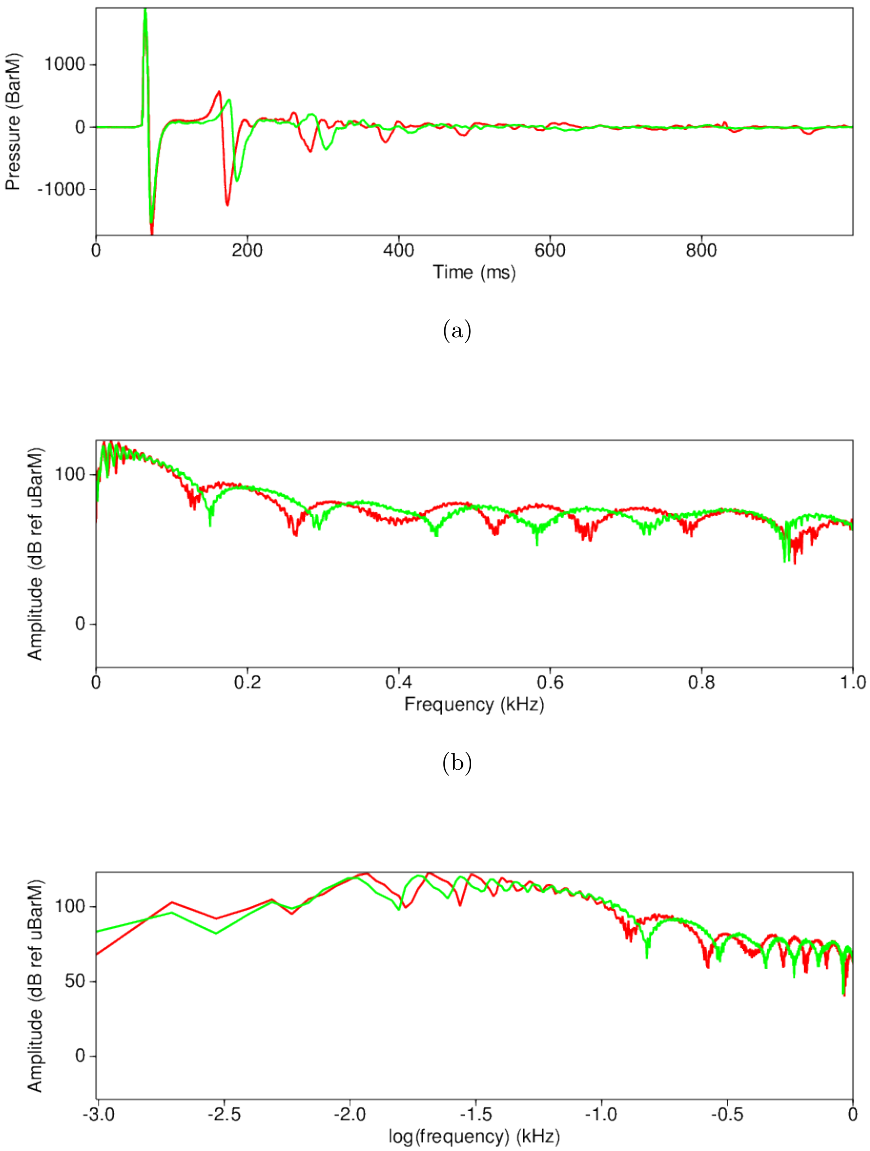
Figure 1: The convolutional model of seismic data. The purpose of the seismic method is to estimate the reflectivity. All other physical filters must be accounted for. In this paper we focus on deterministic methods for debubbling. [**NR**] joseph1/. chart

Deconvolution methods can be grouped as statistical and deterministic. Statistical deconvolution methods assume that the reflectivity is white. The risk is that when it is not white, statistical deconvolution will force it to be white and in the process may remove reflectors that it finds predictable from earlier reflectors. For this reason it is useful to do as little as possible with statistical deconvolution after doing as much as possible with deterministic methods.

Deterministic methods can be grouped into those that are based on library signatures and those that depend on near field hydrophones. Library methods rely on signatures that are based on data similar to those in Figure 2(a) but after deghosting. A problem with library methods is that the same airguns, with the same library signatures, vary from shot to shot because of variable depth, ambiance, and most importantly mechanical issues. In Figure 3 we present data that shows the variations of airgun signatures. There is no way the library signature can account for the variations in the actual signatures.

Near-field hydrophones (NFHs) record data shot by shot. Each NFH is near an airgun, but it also records the other airguns in the array. To estimate the notional signature of each gun, Ziolkowski et al. (1982) developed an inversion method in which unknown notional signatures are fit to NFH data. This is a linear inversion with thousands of equations and as many unknowns, assuming that each airgun has one NFH.

NFH measure variations from shot to shot. However, they also measure other phenomena that are irrelevant to the far field. For example, waves that go up and down the wires and chains that connect the airguns and the NFH to floatation devices. To separate signal from noise in NFH data it is useful to be able to distinguish an airgun bubble signature from additive noise such as waves in the chains and wires. With an ability to reproduce airgun signatures from a small number of key parameters, such as pressure, depth, exact timing, water temperature and salinity, and air temperature we can separate signal and noise in NFH data. To estimate these key parameters, shot by shot, we can use the NFH data. Instead of a linear inversion with thousands of unknowns, we would rather solve a nonlinear inversion with just a few unknowns. While this methodology is our ultimate goal, we have not yet applied the airgun modeling method that we are developing to NFH data. In this progress report, we try to reproduce the library signature in Figure 4. In this report,



(a)

(b)

(c)

Figure 2: Far field hydrophone data (a) and their amplitude spectra in frequency (b) and log frequency (c). The data were acquired in a lake. The hydrophone was about 75 meters below the source. The bottom of the lake was about 120 meters under the hydrophone. The red trace is from an airgun at a depth of about 7 meters below the lake surface. The green trace is from the same airgun deployed at a depth of about 5 meters below the lake surface. The Bubble is dominant up to about 100Hz and the Source Ghost is dominant above 100 Hz. In the time domain (a) the deeper source has a longer delay time of the ghost and a shorter bubble period. In the frequency domain (b and c) the deeper source has a lower ghost notch frequency and a higher bubble frequency. The fundamental bubble frequency is about 9 Hz at 5 meters and about 11 Hz at 7m. The fundamental ghost notch frequency is 150 Hz at 5m and 125 Hz at 7m. Data courtesy of Dolphin Geophysical and Chelminski Associates. [**ER**] joseph1/. lakedata,lakedata-spec,lakedata-spec-log

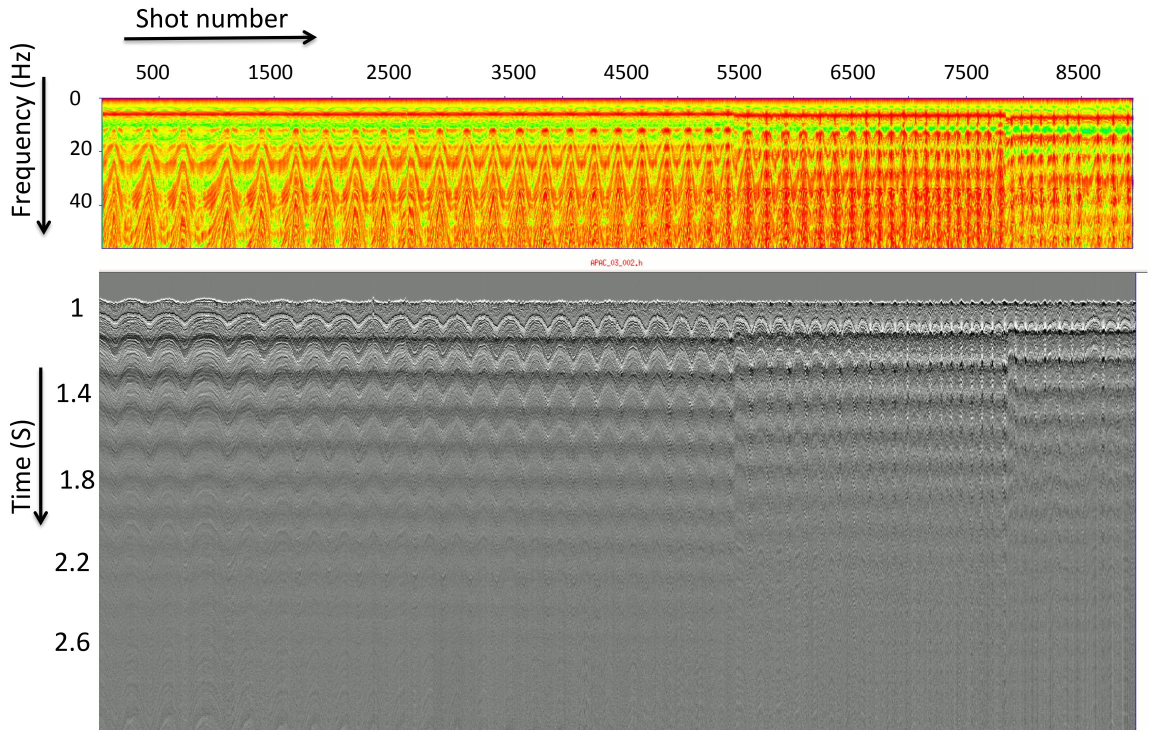
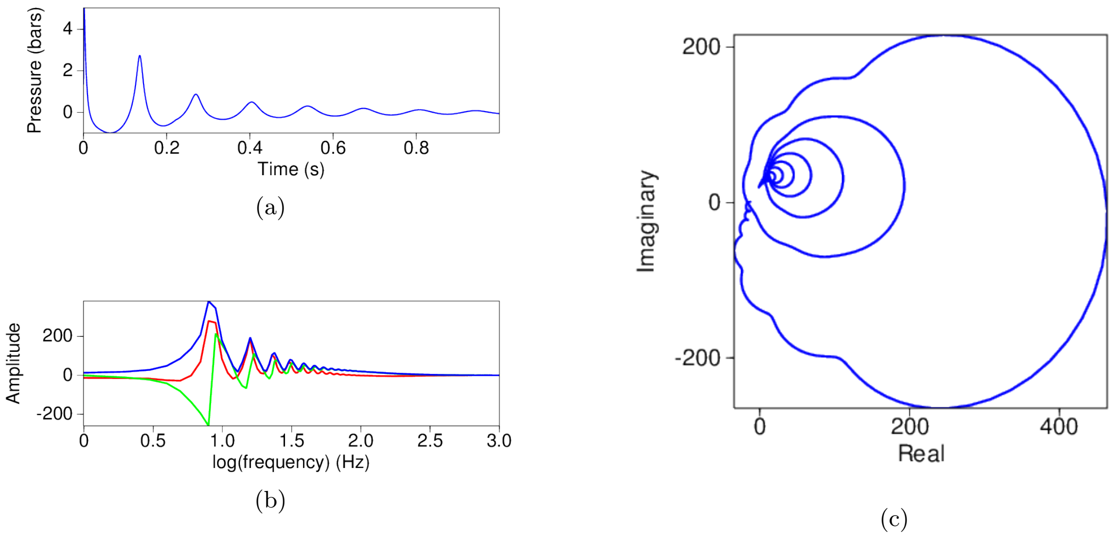


Figure 3: A common receiver gather from an ocean bottom node above the Forties oil field in the North Sea. The gather shows 9000 traces shot over a period of about 24 hours. The horizontal axis is shot number. Amplitude spectrum above and time domain below. The vertical axes are frequency and time. The data have been hyperbolically moved out, so the direct arrivals are approximately flat. The bubble period is about 100 miliseconds. The flat events are the bubble oscilations; the bubble is not affected by the angle. Other events, that are mainly multiples of reflections are affected by angle and are not flat. Note the changes in the bubble signature. One change at about shots 5500 is the start of an airleak. Hence loss of pressure. The second change at about shot 7800 is one airgun that was disconnected. Hence loss of volume. The bubble period decreased with each change. The bubble frequency increased. Data courtesy of Apache North Sea. [**NR**] joseph1/. delta-with-axes



(a)

(b)

(c)

Figure 4: Analysis of the library airgun signature. (a) Time domain wavelet taken from the library. Time gate is limited to one second. (b) its real part (red), imaginary part (green), amplitude spectrum (blue). The frequency band is low-limited to 1 Hz by the time gate and high-limited to 1kH which is the Nyquist frequency of 0.5 millisecond sampling interval. (c) complex hodogram which is a cross plot of the real and imaginary parts. Note from the hodogram and the real part that the signal is primarily positive real. joseph1/. nucleus-raw,reimmag,nucleus-hodo [**ER**]

our goal is to achieve an accurate model of the bubble convolved with the source force in such a manner that requires few modeling parameters.

MIE SCATTERING

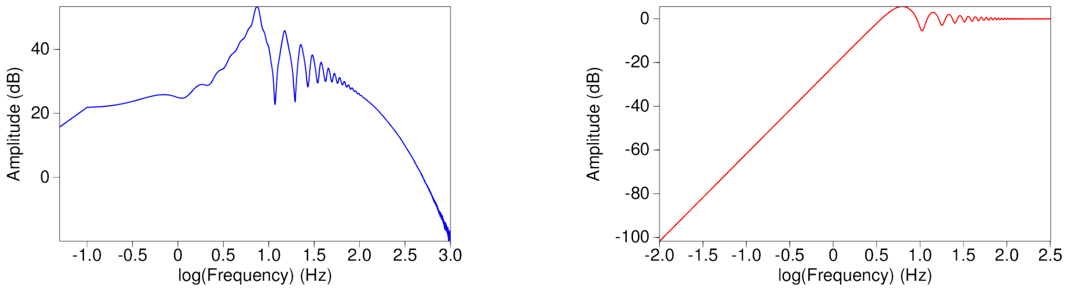
Airgun bubbles are thermodynamic oscillators that radiate acoustic waves. Other oscillators radiate electromagnetic waves. Mie scattering (Mie, 1908) theory predicts scattering of electromagnetic (EM) waves such as radar from metalic objects such as airplanes.

Airgun bubbles and Mie scatterers are both low-cut filters. At low frequencies, there is neither radiation of acoustic waves from an airgun nor scattering of EM waves from small objects. At high frequencies, the airguns fully convert the force of compressed air to acoustic waves and large objects reflect all the EM waves with much smaller wavelength.

The amplitude of the generic Mie scattering plotted as a function of frequency is shown in Figure 5(b). When compared to the spectrum of the library airgun signature in Figure 5(a), it is apparent that while the data spectrum exhibits the shape of a bandpass filter and the modeled spectrum from Mie theory exhibits the shape of low cut filter, the spectra between the low and high frequencies (5-100 Hz) are quite similar in shape. In Figure 6 we observe that Mie scattering spectra is positive real. Positive real wavelets are also minimum phase (Claerbout, 1976). We expect the coupling of source force to acoustic waves to be minimum phase because we expect it to be causal both ways (Kjartansson, 1979). We are not sure about the significance of positive-realness besides insuring minimum-phase. Possibly, positive realness may tell us about the entropy; predict that the (causal) inverse of a (causal) physical process is possible when energy is considered but not possible when entropy is considered. In other words, converting an acoustic wave back to force that would compress air back into an airgun would be causal if it happens, but it would never happen.

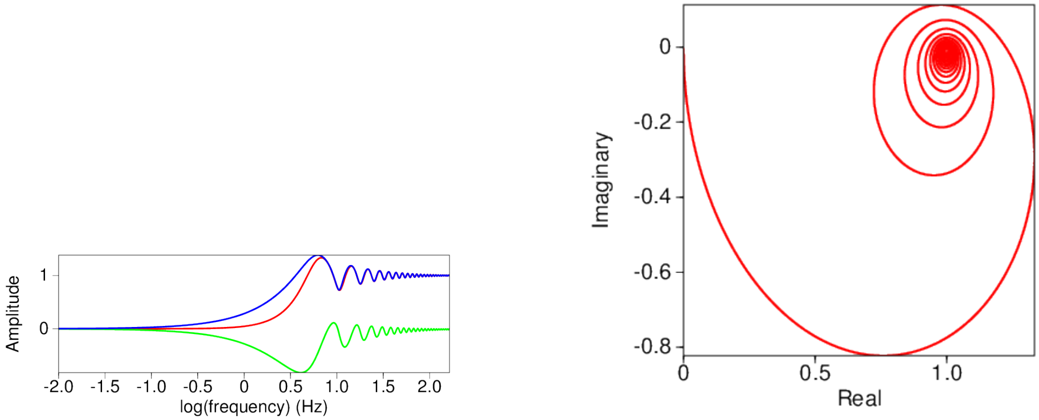
While we do not fully understand the reason for the spectral similarity (Figure 7), we believe that it results from the underlying fact that both Mie scatterers and airgun bubbles are radiating oscillators. There are two questions that require justification. One is that the physics are completely different; the airgun is thermodynamic-acoustic, while Mie is electromagnetic (EM). How can we use one for the other? The other question is how can we use scattering for sourcing. The second question is easier. Babinet's principle (Born and Wolf, 1999) states that scattering theory is applicable to radiation from a source.

The justification of the first question (different physics) is more difficult. The governing equations are Maxwell equations on one side and thermodynamics, Navier Stokes, and acoustics on the other. In one side the scatterer size is comparable to the wavelength of resonance. On the other side, the bubble size is much smaller than the wavelength of the acoustic waves that has the same period. We are aware of this need for justification. We therefore collaborate with researchers in the department of geophysics at Stanford University who are developing a physics-based forward-model of the airgun-bubble coupled system. The plan is to adjust or replace the Mie spectra that we are now using with spectra of wavelets that come from their physics-based model. For now, we hope that the justification may first be because it works and we will later explain why it works. However, we admit that we are not sure that Mie scattering is a workable analogy.



(a) (b)

Figure 5: The library airgun signature spectrum (a) and the amplitude spectrum modeled via Mie scattering theory (b). Note that the signature of the library airgun is a bandpass filter while that of the Mie modeling is a low-cut filter. It means that in addition to coupling which limits the low frequency content of an airgun there is another factor that limits its high frequency content. [**ER**] joseph1/. inspec,genmie



(a) (b)

Figure 6: Analysis of the spectrum modeled with Mie scattering theory. (a) The real part (red), the imaginary part (green) and the amplitude spectrum (blue). (b) The complex hodogram. Note from the hodogram and the real part that the signal is positive real. [**ER**] joseph1/. reimmagmie,miehodo

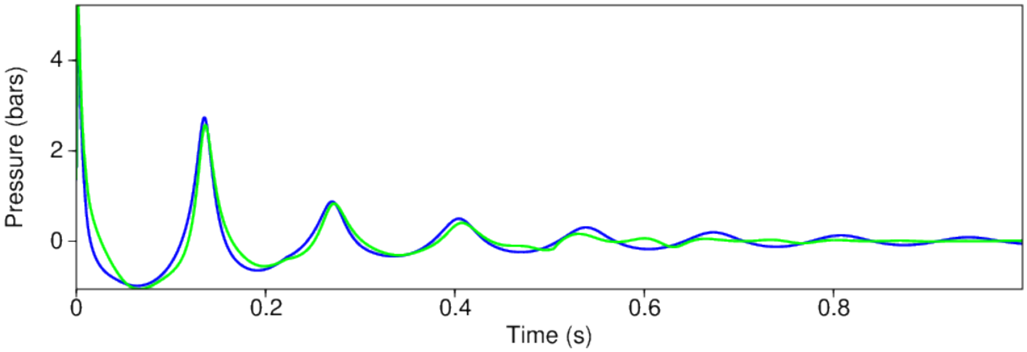
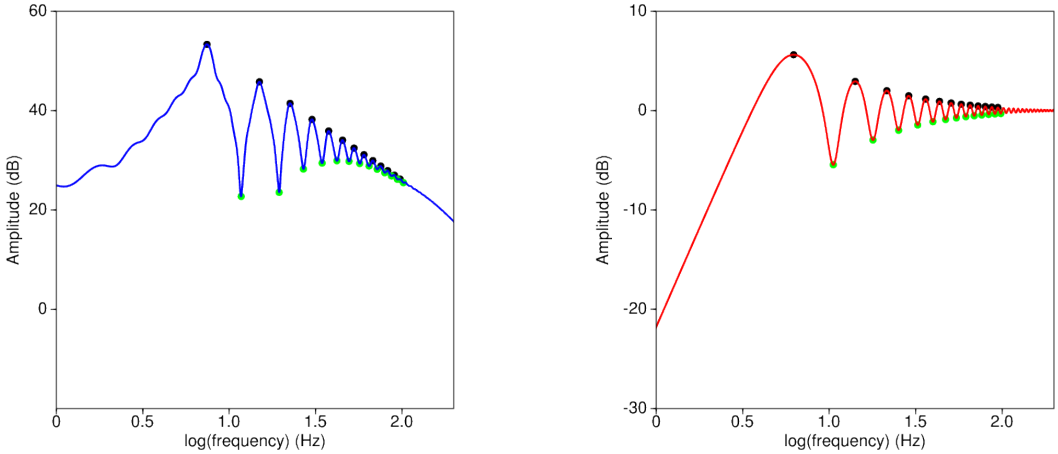
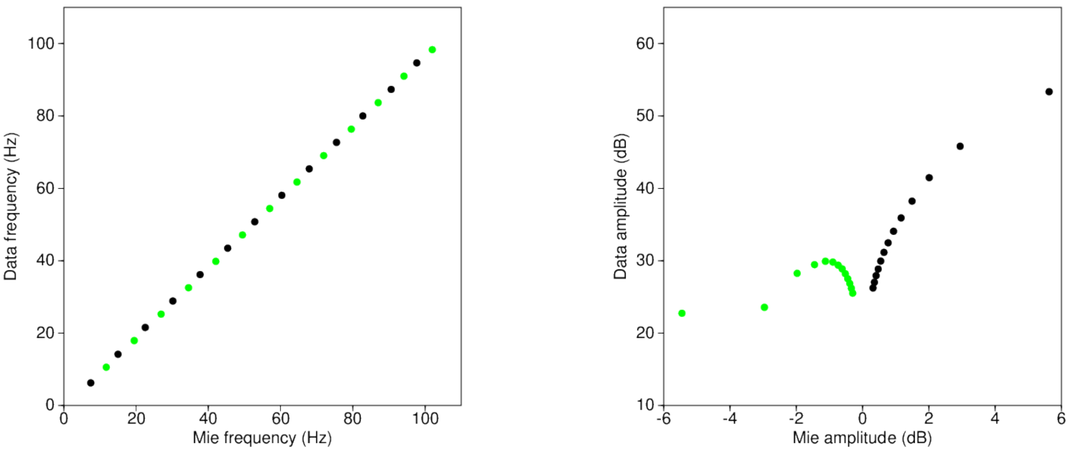


Figure 7: Overlay of the minimum phase equivalent of the library signature (green) and the library signature (blue). The minimum phase equivalent was obtained via Kolmogoroff spectral factorization. The library signature is very close to minimum phase. [**ER**] joseph1/. kolmnuc



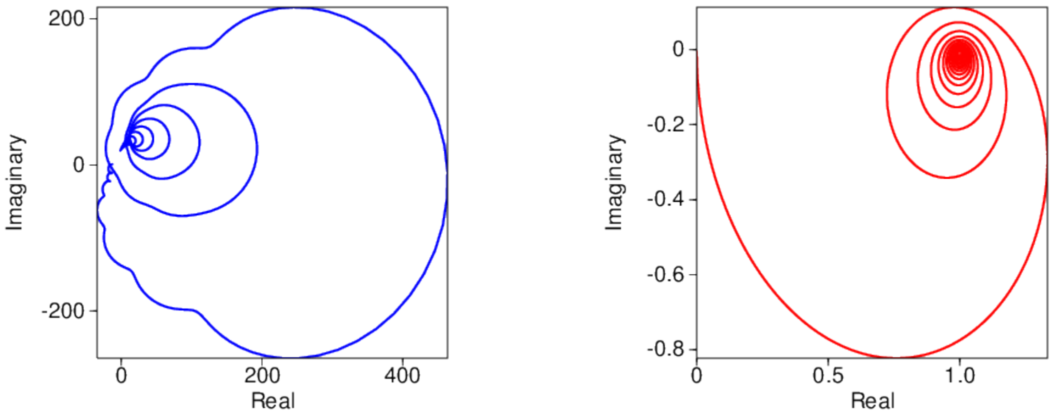
(a) (b)

Figure 8: The control points for the warping of the amplitude spectrum modeled via Mie scattering theory (b) to the library airgun signature spectrum (a). [**ER**] joseph1/. datapick,miepick



(a) (b)

Figure 9: The cross plots of (a) Mie frequency and the library air gun frequency and (b) Mie amplitude and the library air gun amplitude. Note that as in Figure 8, the black points denote the maxima and the green points denote the minima. (a) shows that we can warp the frequencies but not the amplitudes. This is because the amplitudes are affected by the spectra of a bandlimited source force function and the Mie spectra is for an infinte-band impulse. [**ER**] joseph1/. freqqccross,ampqccross



(a)

(b)

Figure 10: The library airgun complex hodogram (a) and the complex hodogram of the spectrum modeled by Mie scattering theory (b). Note that the Mie wavelet is positive real and the library wavelet is almost positive real. We suspect that the deviation of the library wavelet from positive-real as well as the wobbliness of the hodogram in low frequencies (where it starts at zero real and zero imaginary) are due to the limitations of estimating airgun signature from experimental data. Mainly, additive low frequency noise in the experimental data and the fairly short recording time of just one second. [**ER**]
joseph1/. nucleus-hodo,miehodo

METHOD

Note that from this point, we refer to the spectrum modeled via Mie scattering theory as 'Mie' and the library air gun signature as 'library'.

Our general methodology for debubbling the library spectrum is to first warp the spectrum shown in Figure 5(b) (a generic Mie spectrum) to the air gun library spectrum in Figure 5(a). After we compute an adequately warped amplitude spectrum, we then observe the time domain wavelet of this warped Mie spectrum using Kolmogoroff spectral factorization. The justification is that we know that the air gun library trace is minimum phase as shown in Figure 7. With the spectrum and the time domain wavelet, we attempt to debubble the library air gun signature.

Warping Mie to library signature

As stated above, in order to fit the Mie spectrum to the library spectrum, we decide to use a warping approach. That is, we need to find a mapping from a point in the mie space (x, m) to the library space (f, d). More formally, we need to find mappings between the Mie frequency to the library frequency as expressed in equation 1 and Mie amplitude to library amplitude as expressed in equation 2.

$$x \mapsto f \tag{1}$$

$$m \mapsto d \tag{2}$$

While we desire to fit each and every point from the Mie to the library spectrum, we realize that this will only be possible with many fitting terms due to the differences between the two spectra. Therefore, instead of fitting each and every point, we decide to focus on fitting the maxima and minima of the spectra. In other words, we use these points as our "control points". In Figures 8(a) and 8(b) the picked control points are shown where the black points denote the maxima and the green the minima. To get an idea of the mathematics that potentially describe the mappings of the Mie maxima and minima to the library, maxima and minima, we create cross plots of x against f in Figure 9(a) and m against d in Figure 9(b). Once again, in these figures, the green and black points denote the minima and maxima respectively.

Frequency warping

We observe that in Figure 9(a), the mapping is nearly linear and that we we can define the mapping from Mie frequency (x) to the library frequency (f) as:

$$x = bf \tag{3}$$

Using a simple weighted least-squares regression, we can estimate our b parameter. With b, we can then map the Mie frequencies to the data frequencies.

Amplitude warping

Once we have achieved this mapping from Mie frequency (x) to data frequency (f), we can then begin to warp the amplitudes. From Figure 9(b) we observe that the mapping is not linear and therefore may require several parameters. We attribute the complexity of this mapping to the fact that in our Mie modeling we have not taken into account the source force that is present on the library wavelet giving its band pass filter shape. The Mie spectrum, on the other hand, is a low-cut filter and therefore its complex hodogram does not converge to zero as is evident in Figure 10(b). Applying a source force to the Mie spectrum would make this mapping much more linear. The source force function is a convolution operator that multiplies the coupling to generate the library wavelet. Because we warp the amplitudes in the log domain, this multiplication becomes an addition. Therefore, we can describe the mapping between the Mie amplitude and data amplitude as:

$$d(f) \approx m_{warped}(f) = a_m m(x(f)) + s(f) \tag{4}$$

where d is the library airgun signature (data), a_m is a scale factor to be applied to the unwarped Mie spectrum, m is the unwarped Mie spectrum and $s(f)$ is the source force. We parameterize the source force $s(f)$ as

$$s(f) = \sum_{i=-1}^{2} a_i f^i \tag{5}$$

where a_i denotes the coefficients of the $s(f)$ polynomial.

Once again, we can use a linear regression in order to estimate a_m, a_{-1}, a_0, a_1 and a_2 where we define our residual as:

$$r_i = d_i - a_m m(f_i) - \frac{a_{-1}}{f_i} - a_0 - a_1 f_i - a_2 f_i^2 \tag{6}$$

Therefore, our fitting goal is to minimize this residual.

RESULTS

Warping results

After estimating the frequency, amplitude and source force parameters described in the previous section and warping the Mie spectrum to the library spectrum, we obtain the warped Mie spectrum shown in Figure 11. In this figure, it is clear that while we have fit several maxima and minima quite well, we fail to fit the 8Hz low frequency maxima. Furthermore, we can observe that at low frequencies the spectra seem to agree, but then at the higher frequencies they mismatch This is due to a weighting operator that we applied to the residual in estimating our frequency fitting parameter b (3). This is quite evident in Figure 12 where the slope of the red line is the estimated b parameter. The weighting operator was chosen to fit the first four maxima and minima and the remaining elements of the diagonal were left as zeros.

We use Figure 13 as a quality-check for our fit. Instead of showing the relationship between x and f, m and d, these plots show the mapping between the minima and maxima of the warped Mie and the minima and maxima of the library signature. As expected the mapping between x_{warped} and f shown in Figure 13(a) is still linear. In Figure 13(b) we observe that warping the Mie spectrum with the incorporated source force did greatly improve the mapping, but it is not quite linear. This is mostly due to the use of our weighting operator which gives us a better fit of the lower frequencies.

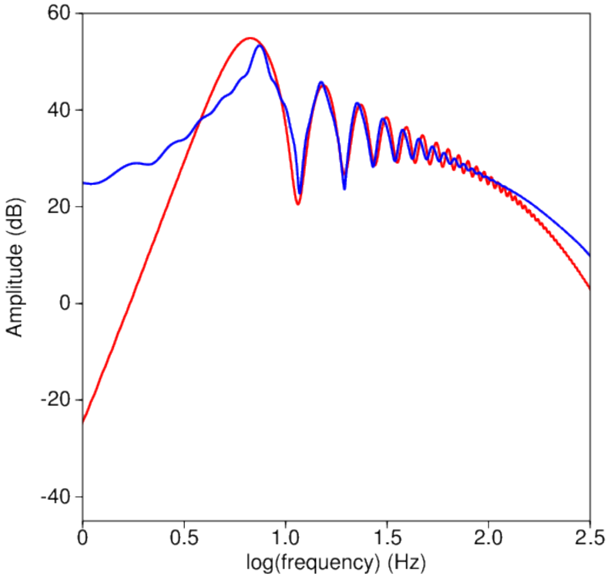


Figure 11: Warping result. [**ER**] joseph1/. warpingoverlay

Time domain wavelets

Now that we have warped the Mie spectrum to the data spectrum, we can find the time-domain minimum-phase equivalent wavelet using Kolmogoroff spectral factorization. Performing this computation gives us the time-domain Mie wavelet shown in Figure 14. Comparing this wavelet to the library wavelet also shown in Figure 14, we see that the Mie wavelet in general resembles that of an airgun signature. Clearly, it shows bubble oscillations and has a peak to bubble ratio comparable to that of the library wavelet. Unfortunately, the deep second bubble oscillation tells us that this result is non-physical as the energy of the bubble should be decreasing as time increases. We consider this result as a preliminary finding on our path to finding a better wavelet.

Designature

With the warped Mie spectrum, we can now deconvolve the bubble from the library airgun signature. We perform this deconvolution on the amplitude spectrum in the dB domain.

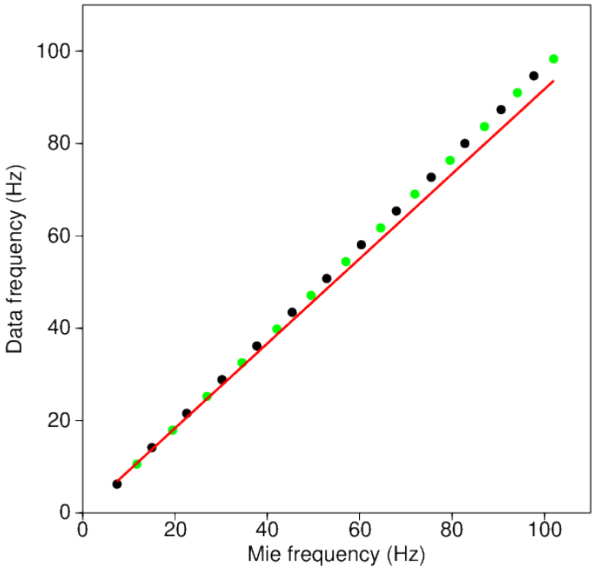
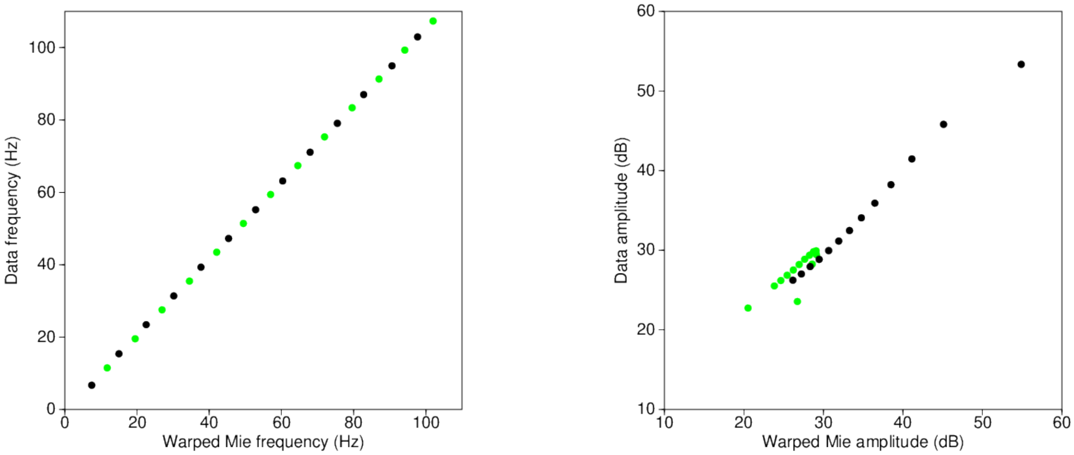


Figure 12: The fitted cross plots of Mie frequency and the library air gun frequency. Note that as in Figure 8, the black points denote the maxima and the green points denote the minima. [**ER**] joseph1/. freqqccross-fit



(a)

(b)

Figure 13: The cross plots of (a) warped Mie frequency and the library air gun frequency and (b) warped Mie amplitude and the library air gun amplitude. Comparing (b) with Figure 9(b), we observe that the mapping is now nearly linear between the amplitudes. [**ER**] joseph1/. mw2ddatfreqqc,mw2ddatampqc

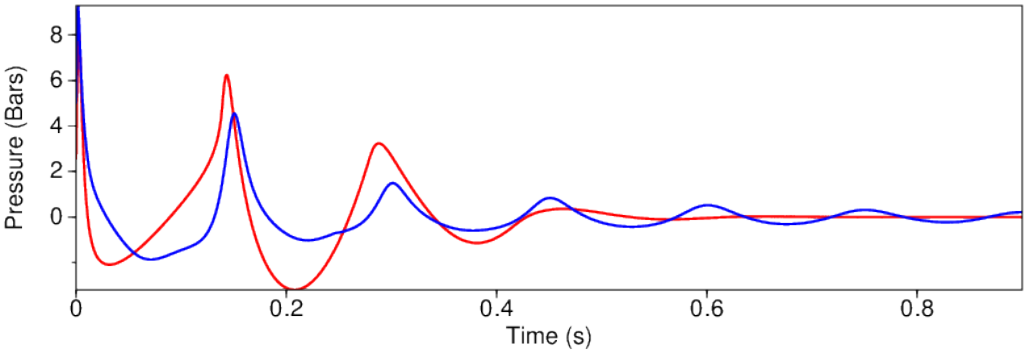


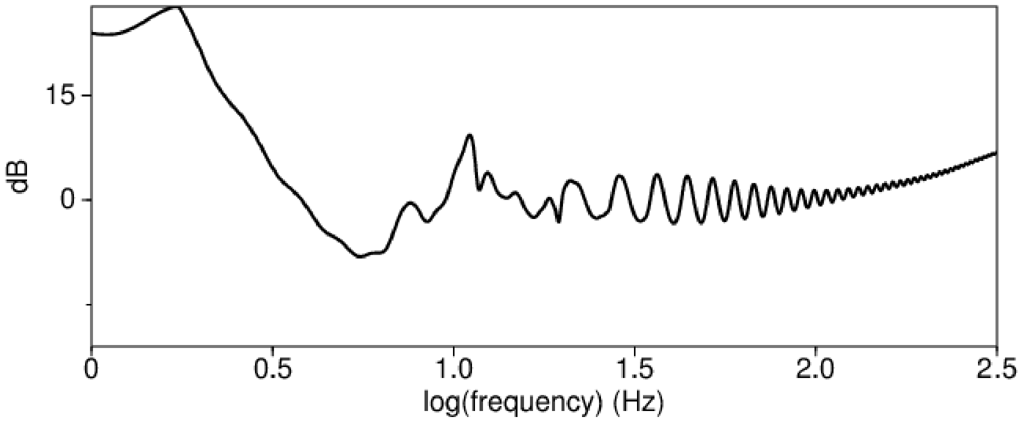
Figure 14: The comparison of the time-domain warped Mie wavelet (red) and the original library wavelet (blue). We are not satisfied with this result; the bubble shape and period fits are not perfect and make the result unsatisfactory. The peak to bubble ratio is under estimated, and the second trough is deeper than the first trough, which is non-physical. Nevertheless, we show this result in this progress report. [**ER**] joseph1/. kolmoverlay

The designatured amplitude spectrum can be seen in Figure 15(a). In performing the deconvolution, a pre-whitening factor of 0.1 was used. From this figure, we observe the remaining bubble spectrum from approximately 8-100 Hz.

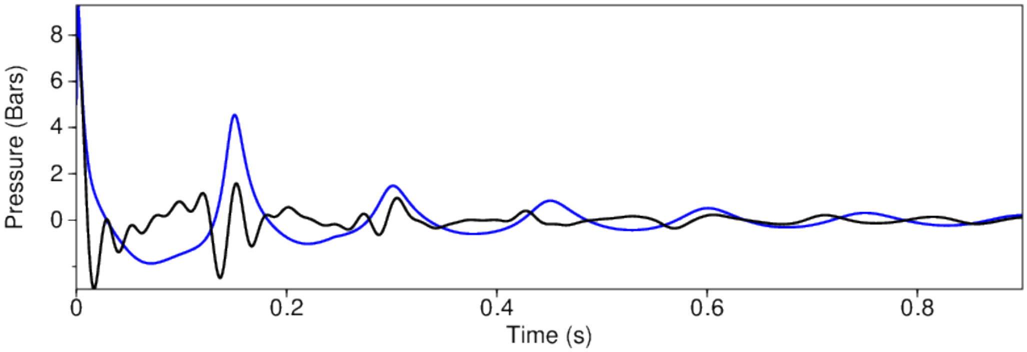
Once again, to find the minimum-phase time-domain equivalent wavelet we use Kolmogoroff spectral factorization. The resulting designatured time-domain wavelet is shown Figure 15(b). While it appears that some of the bubble signature was removed as a result of the deconvolution, most of it remains due to the misfit between the Mie and library wavelets.

DISCUSSION AND CONCLUSION

In this progress report, we have shown that by warping the spectrum computed by Mie scattering theory and then finding its minimum phase equivalent time-domain wavelet by Kolmogoroff spectral factorization we can obtain a time-domain wavelet that resembles an air gun signature. With this wavelet, we have attempted to designature a library airgun signature. The results are encouraging. However, as stated previously, we are not satisfied with the wavelets shown in Figures 14-15(b) as they are non-physical. We believe that the results we have obtained are non-physical due to the fundamental differences between the Mie spectrum and the library spectrum and therefore. These differences force us to rely heavily on warping to get a signature that resembles one of an airgun. To overcome this, in the future we will look into alternative Mie formulations and attempt to warp the spectrum in the complex hodogram domain. This will ensure that the Mie spectrum is positive real and also will have the same amplitude and phase spectrum as the library airgun spectrum. We believe that this will resolve the matches in the bubble period.



(a)



(b)

Figure 15: (a) The designatured amplitude spectrum and (b) the minimum-phase equivalent of the designtured amplitude spectrum computed via Kolmogoroff spectral factorization. In computing the designature, a pre-whitening factor of 0.1 was used. [**ER**] joseph1/. desig,desig-overlay

ACKNOWLEDGEMENTS

The authors would like to thank Dolphin Geophysical and Chelminski associates for the permission to show the lake data. They would also like to thank Avishai Ben David for observing that the airgun source spectra resemble Mie scattering spectra and Ettore Biondi for thoughtful discussions

REFERENCES

Born, M. and E. Wolf, 1999, Principles of optics: Cambridge University Press.

Claerbout, J., 1976, Fundamentals of geophysical data processing with applications to petroleum prospecting: Blackwell Scientific Publications.

Kjartansson, E., 1979, Attenuation of seismic waves in rocks and applications in energy exploration: PhD thesis, Stanford University.

Mie, G., 1908, Beitrge zur optik trber medien, speziell kolloidaler metallsungen: Annalen der Physik, **330**, 337–445.

Ziolkowski, A., G. Parkes, L. Hatton, and T. Haugland, 1982, The signature of an air gun array: Computation from near-field measurements including interactions: Geophysics, **47**, 1413–1421.

Balancing amplitude and phase in TFWI

Ali Almomin and Biondo Biondi

ABSTRACT

Tomographic Full Waveform Inversion (TFWI) provides a robust and accurate method to invert the seismic data by simultaneously inverting all scales of the model using both amplitude and phase information. However, one shortcoming of TFWI is the large number of iteration required to achieve accurate results due to its slow convergence. In this report, we analyze the source of its slow convergence and propose two modifications to mitigate the problem. First by modifying the formulation of the regularization term to focus more on the phase information, and second by using an alternative enhancing operator that is less sensitive to the amplitudes in the extended model. Then, we test the modified TFWI on the marmousi 2 model. The results show a significant improvement in the convergence rate.

INTRODUCTION

TFWI, similar to other data-space inversion method, produces highly accurate results due to matching both the amplitude and phase of the data. This is achieved in two steps: first, extending the wave equation and adding an additional axis to the velocity model, and second, adding a regularization term that drives the solution towards a non-extended model. However, one limitation to TFWI in its earlier formation, as shown in Almomin and Biondi (2013), is the large number of iterations required to achieve accurate results. This large number of iterations causes TFWI to be very expensive, especially for large-scale 3D datasets. In Almomin and Biondi (2014), I proposed a wavelength continuation scheme that preconditioned the gradient and smoothed the updates in the early iterations. While preconditioning did improve the convergence rate, the number of iterations was still in the hundreds.

In this report, we analyze the source of TFWI's slow convergence and find that it is due to the unbalanced effects of amplitudes and phase both in the formulation of the regularization term and the enhancing operator. This imbalance resulted in a strong dependence of the kinematic updates on the amplitude fitting, causing it to take more iterations. To mitigate the problem and speed up convergence, we propose two modifications to TFWI. First by modifying the formulation of the regularization term to focus more on the phase information, and second by using an alternative enhancing operator that is less sensitive to the amplitudes in the extended model. Finally, we test the modified TFWI on the marmousi 2 synthetic model.

MODIFIED REGULARIZATION

The conventional L2 objective function for TFWI can be written as follows:

$$J_{\mathrm{TFWI}}(\widetilde{\mathbf{m}}) = \frac{1}{2}\|\mathbf{f}(\widetilde{\mathbf{m}}) - \mathbf{d}_{\mathrm{obs}}\|_2^2 + \frac{\epsilon}{2}\|\mathbf{A}\widetilde{\mathbf{m}}\|_2^2 = \frac{1}{2}\|\mathbf{r_d}\|_2^2 + \frac{\epsilon}{2}\|\mathbf{r_m}\|_2^2, \tag{1}$$

where $\widetilde{\mathbf{m}}$ is the extended model, $\mathbf{f}$ is the forward modeling operator, $\mathbf{d}_{\mathrm{obs}}$ is the observed surface data, $\mathbf{A}$ is the defocusing operator, $\mathbf{r_d}$ is the data-fitting residual and $\mathbf{r_m}$ is the model regularization residual. The gradient $\mathbf{g}$ can be written as:

$$\mathbf{g}(\widetilde{\mathbf{m}}) = \mathbf{L}^*(\widetilde{\mathbf{m}})\mathbf{r_d} + \epsilon\mathbf{A}^*\mathbf{r_m}, \tag{2}$$

where * denotes an adjoint and $\mathbf{L}$ is the linearized modeling operator. To avoid cycle-skipping, the modeled data needs to be very close to the observed data such that the data residual is basically a 90 degree phase rotation of the observed data. In TFWI the data-fitting term can fit the observed data regardless of the initial model because the model is extended. The regularization term will adjust the model such that the modeled data in the following iterations will slightly shift from the observed data to create a kinematic difference that results in a meaningful tomographic update.

To better understand the process, we can think of the different stages that TFWI cycles through as it iterates. First, the data-fitting term fits the observed data by creating reflectors in the extended model. Then, the regularization term slightly focuses the model which results in a small shift of the modeled data and a decrease in the data-fitting. Finally, the data-fitting term fits the shifted modeled data to the observed data by creating smooth tomographic updates in the model and adjusting the reflectors locations. Notice that, except for the first few iterations, all these stages happen simultaneously in every iteration.

There are a few convergence issues that take place in practice. First, the data-fitting term creates both the reflectors and tomographic updates which makes it more difficult to balance or emphasize either amplitude or phase fitting. Second, the data-fitting term takes several iterations until it becomes sufficiently small mainly due to amplitude differences resulting from using an adjoint instead of an inverse. If the regularization term focuses the model before the model data fits the observed data, then the data residual will not produce tomographic updates. Third, the regularization term will continue to focus the model at a rate that only depends on epsilon, regardless the data-fitting term. This results in a strong sensitivity to epsilon. If epsilon is too small, the focusing will be too slow and the inversion might take thousands of iterations before producing any useful tomographic updates. If epsilon is too large, the model will be focused too fast and will result in cycle-skipping.

To solve the previous issues, we want to modify the regularization term such that it directly produces the tomographic updates instead of indirectly through the data-fitting term. We can start by rewriting the regularization term as follows:

$$\frac{\epsilon}{2}\|\mathbf{A}\widetilde{\mathbf{m}}\|_2^2 = \frac{\epsilon}{2}\|\widetilde{\mathbf{m}} - (\mathbf{I} - \mathbf{A})\widetilde{\mathbf{m}}\|_2^2 = \frac{\epsilon}{2}\|\widetilde{\mathbf{m}} - \mathbf{E}\widetilde{\mathbf{m}}\|_2^2, \tag{3}$$

where $\mathbf{E}$ is an enhancing or focusing operator which is the complement of $\mathbf{A}$. It is easier to see that the regularization term minimizes the difference between the extended model and an enhanced version of the extended model by focusing the model. This formulation makes it similar to the data-fitting term, however, it is still missing the wave-equation operators. Therefore, we propose a new regularization term as written in the following objective function:

$$J_{\mathrm{TFWI}}(\widetilde{\mathbf{m}}) = \frac{1}{2}\|\mathbf{f}(\widetilde{\mathbf{m}}) - \mathbf{d}_{\mathrm{obs}}\|_2^2 + \frac{\epsilon}{2}\|\mathbf{f}(\widetilde{\mathbf{m}}) - \mathbf{f}(\mathbf{E}\widetilde{\mathbf{m}})\|_2^2. \tag{4}$$

This new regularization term completely changes the behavior of TFWI because it does not modify the model. Instead, it directly calculates a residual between the modeled data and the focused modeled data and back-projects it into a tomographic update. This regularization residual is guaranteed to have the correct amount of kinematic shift compared to the modeled data regardless of how well we fit the observed data. Therefore, the data-fitting term only produces reflectors while the regularization term only produces tomographic updates. Moreover, epsilon now balances amplitude fitting and phase fitting without any danger of cycle-skipping at any value, making the inversion process less sensitive to epsilon.

ENHANCING OPERATORS

The enhancing operator is required to create a kinematic shift in the regularization term that can be back-projected into tomographic updates. In the case of DSO, the enhancing operator, i.e. the complement of the focusing operator, is described in equation (3), is:

$$\mathbf{E}_{\mathrm{DSO}}\widetilde{\mathbf{m}}(\tau) = \left(1 - \frac{\tau}{\tau_{\mathrm{max}}}\right)\widetilde{\mathbf{m}}(\tau), \tag{5}$$

where τ is the extended axis value, either in subsurface offset or time lag. As described in the previous section, a proper enhancing operator should result in a residual that is a 90 degree phase rotation of the modeled data. However, DSO operator creates the tomographic update without creating such phase rotated residual. Instead, I scales the amplitude such that the combined effect of all lags (or all offsets in data space) result in the desired tomographic update.

This DSO approach has a few shortcomings. First, DSO assumes the amplitudes will not change significantly along the reflectors or along offsets. This assumption breaks down in many cases such as the presence of AVO effects, complex geometry or irregular acquisition. Second, DSO enhances the image by scaling down the amplitudes along the extended axis. Since the total energy in the data is conserved, this approach assumes that the energy reduction in the image will be converted to tomographic updates in the velocity model, so it indirectly moves focuses the reflectors energy towards the zero-lag. However, this assumption ignores the possibility that the energy can simply be converted reflectors in the null-space of the modeling operator, which are usually present at the edge of the model.

We illustrate the effects of these shortcomings in a simple synthetic example with a single reflector and a constant velocity. Figure 1 shows a shot profile from the modeled data with a slow background velocity using an extended image. The extension of the image preserves all the kinematic information in the observed data. Figure 2(a) shows a shot profile from the modeled data with a slow background velocity using DSO operator regularization on the extended image. Figure 3 compares a trace at 2km offset of the modeled data (bottom) with the DSO regularization residual (middle). We can see that there is no significant change in the phase when we compare the DSO regularization residual to the modeled data and only a small amplitude change.

We propose using a different enhancing operator that shifts the energy towards the zero-lag of the extended axis as follows:

$$\mathbf{E}_{\text{shift}}\widetilde{\mathbf{m}}(\tau) = \widetilde{\mathbf{m}}(\tau + 1 * \text{sign}(\tau)). \tag{6}$$

This shifting operator directly forces the energy to move towards the zero-lag. The main advantage to this enhancing operator compared to the DSO enhancing operator is that it rotates each traces by 90 degrees, therefore, the tomographic update does not depend on how the amplitude of different traces affect each other. In other words, we have no removed the amplitude assumptions. Furthermore, since the energy is directly focused, there are no artifacts at the edge of the model.

We recalculate the regularization residuals with the shifting operator. The shot profile is shown in Figure 2(b) and the trace at 2km is shown in Figure 3 (top). We can see the phase rotation at every offset of the residual.

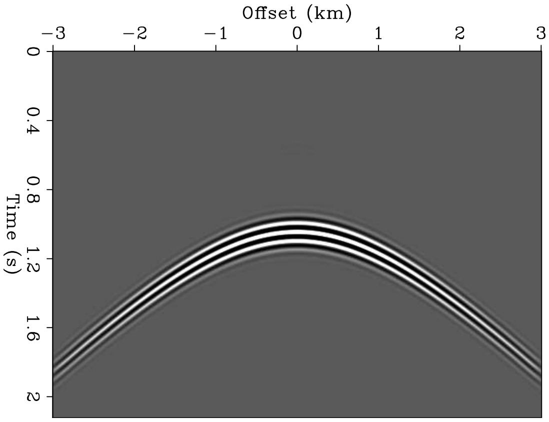
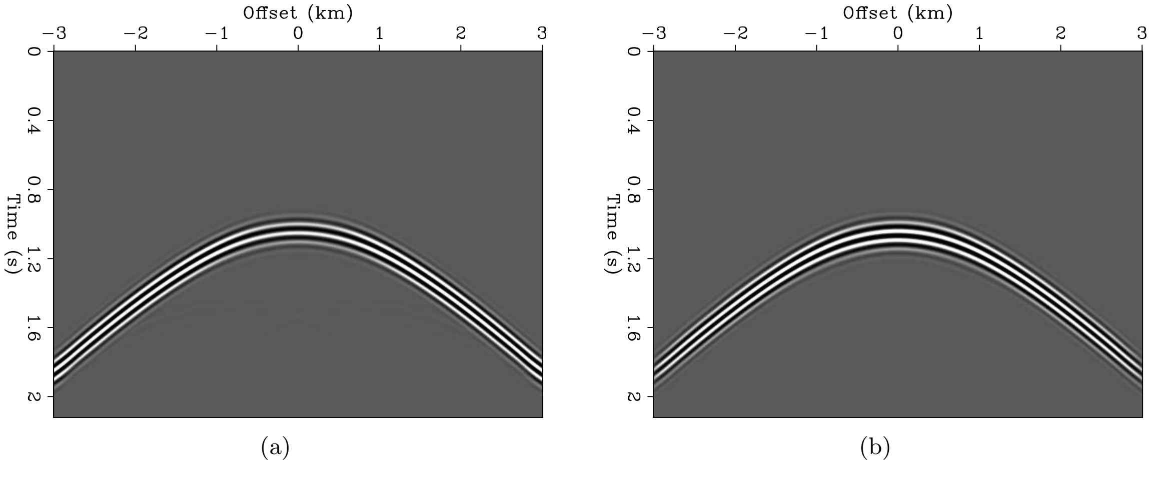


Figure 1: A shot profile from the modeled data with a slow background velocity using an extended image. [**CR**] ali1/. 1



(a) (b)

Figure 2: A shot profile from the modeled data with a slow background velocity using (a) DSO operator regularization on the extended image and (b) shifting operator regularization on the extended image. [**CR**] ali1/. 2,3

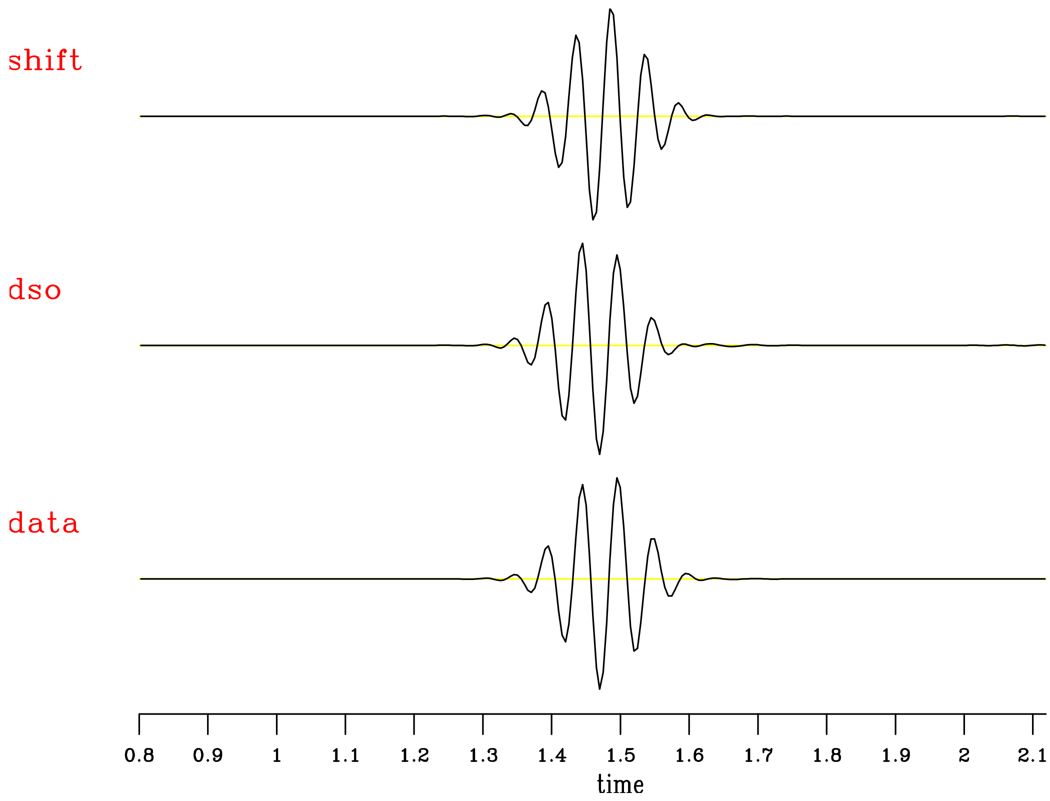


Figure 3: A trace at 2km offset from the modeled data with a slow background velocity using an extended image (bottom), DSO operator regularization on the extended image (middle) and shifting operator regularization on the extended image (top). [**CR**] ali1/. 4

We repeat the previous example but with a fast background velocity instead of a slow background velocity. Figure 4 shows a shot profile from the modeled data with a slow background velocity using an extended image. Figures 5(a) and 5(b) show a shot profile from the modeled data with a slow background velocity using DSO operator regularization and shifting operator regularization, respectively, on the extended image. Figure 6 compares a trace at 2km offset of the modeled data (bottom) with the DSO regularization residual (middle) and shifting operator regularization (top). Again, we can see that there is no significant change in the phase when we compare the DSO regularization residual to the modeled data and only a small amplitude change. On the other hand, the shifting operator shows a clear phase rotation (and in the opposite direction to the previous example).

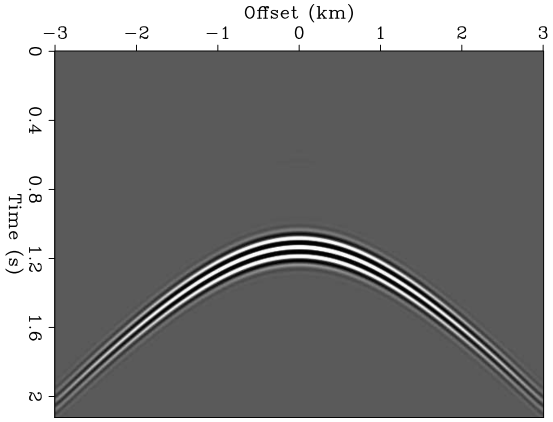
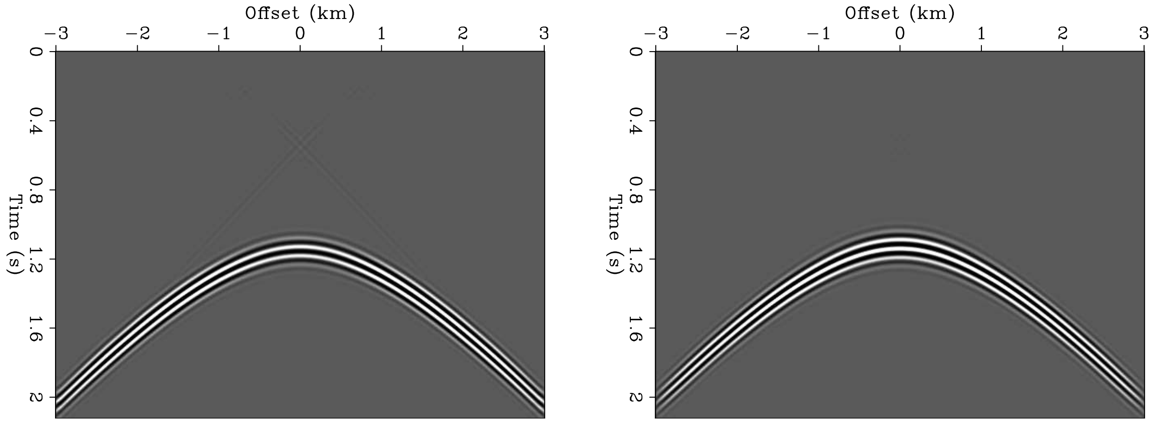


Figure 4: A shot profile from the modeled data with a fast background velocity using an extended image. [**CR**] ali1/. 5



(a) (b)

Figure 5: A shot profile from the modeled data with a fast background velocity using (a) DSO operator regularization on the extended image and (b) shifting operator regularization on the extended image. [**CR**] ali1/. 6,7

Next, we compare the two enhancing operator in model space by calculate the tomographic update of two Gaussian anomaly, one faster than the background velocity and one slow than the background velocity with a flat reflector below. The flat and constant-amplitude reflector is the best possible scenario for DSO. Figures 7(a) and 7(b) show the

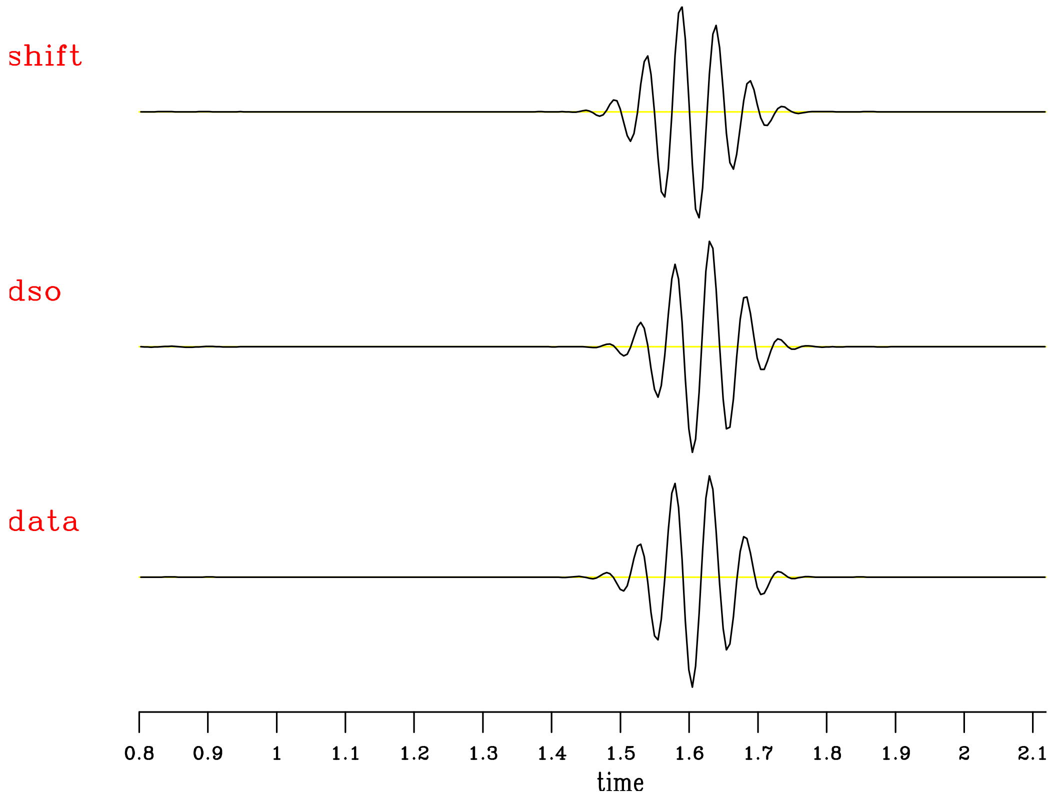
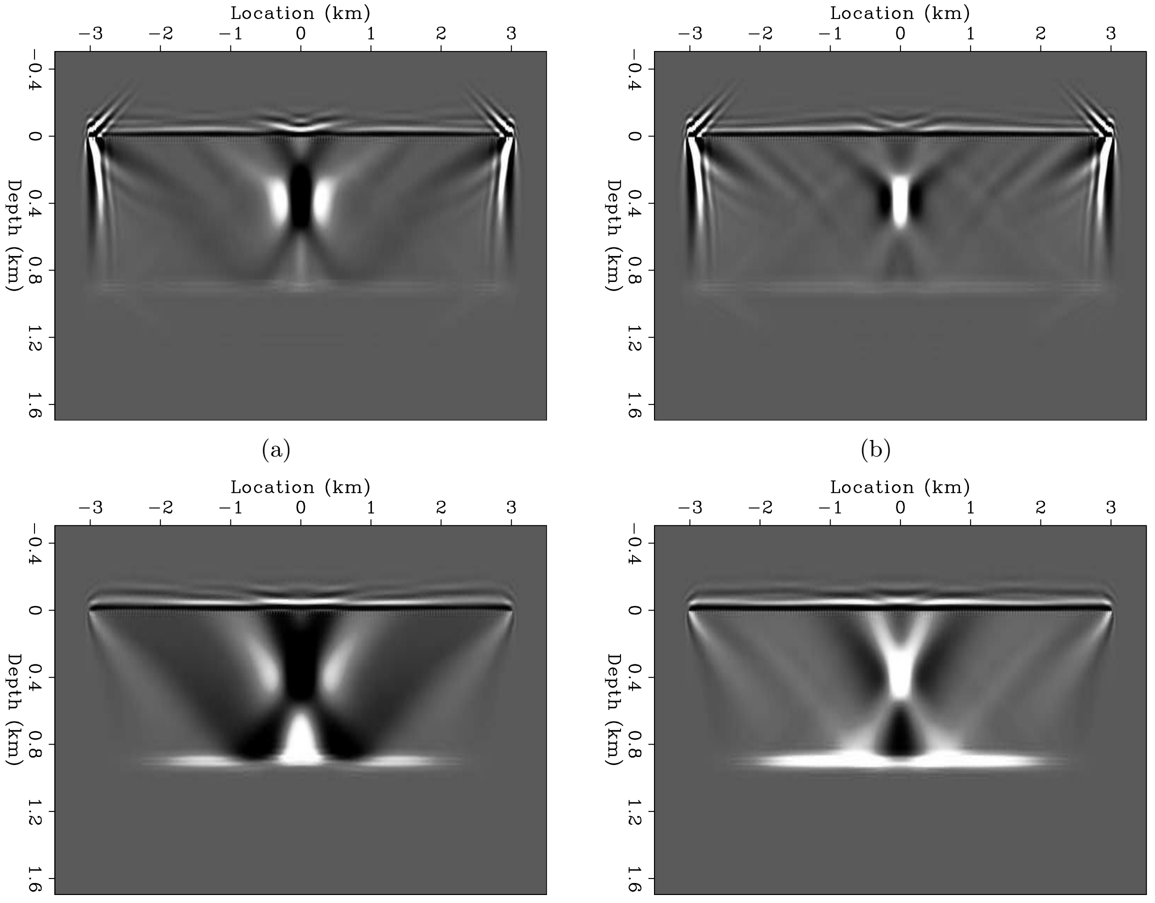


Figure 6: A trace at 2km offset from the modeled data with a fast background velocity using an extended image (bottom), DSO operator regularization on the extended image (middle) and shifting operator regularization on the extended image (top). [**CR**] ali1/. 8

tomographic update of the fast and slow anomalies, respectively, using DSO regularization. Figures 7(c) and 7(d) show the tomographic update of the fast and slow anomalies, respectively, using shifting operator regularization. There is a significant reduction in kinematic artifacts of the shifting operator around the edges of the model. However, since DSO uses the amplitudes of the image, it has a better focusing of the location of the anomaly.



(a)

(b)

(c)

(d)

Figure 7: The tomographic update of (a) fast anomaly using DSO regularization, (b) slow anomaly using DSO regularization, (c) fast anomaly using the shifting operator regularization and (d) slow anomaly using the shifting operator regularization. [**CR**] ali1/. 9,10,11,12

SYNTHETIC EXAMPLES

To test the new algorithm, we run a synthetic TFWI example on the marmousi 2 model. We use a Ricker wavelet with a frequency range between 5 Hz to 25 Hz and a small taper on both ends. Figure 8 shows the correct velocity model. There are 851 fixed receivers with a spacing of 20 m and 171 sources with a spacing of 100 m. The initial 1D model is shown in Figure 9(a) which is obtained by taking the horizontal average of the correct model after removing the high velocity and low velocity anomalies. The RTM image obtained using the initial model is shown in Figure 9(b).

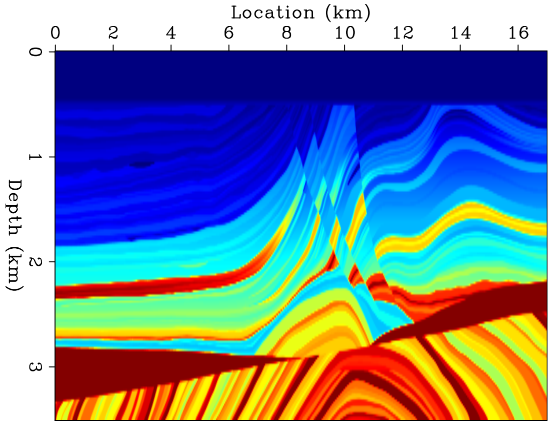
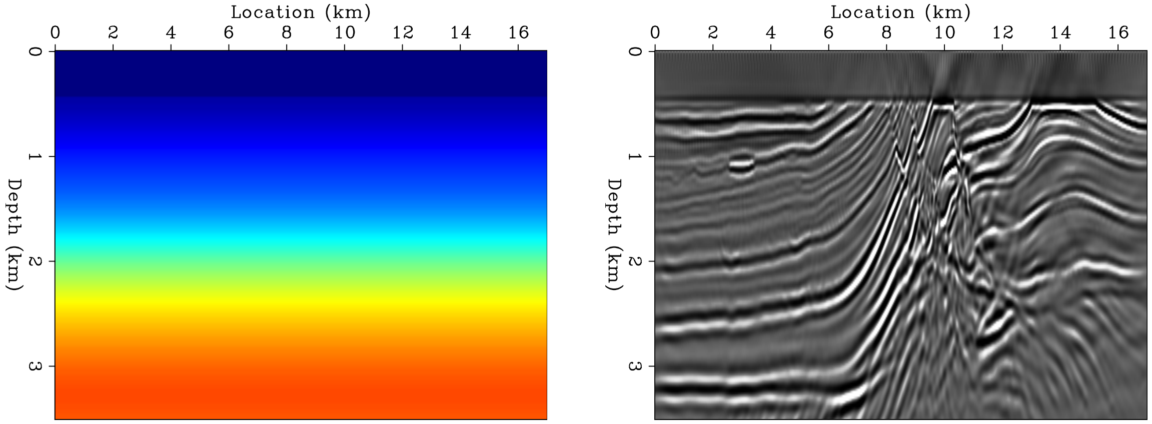


Figure 8: The marmousi 2 velocity model. [**ER**] ali1/. 13

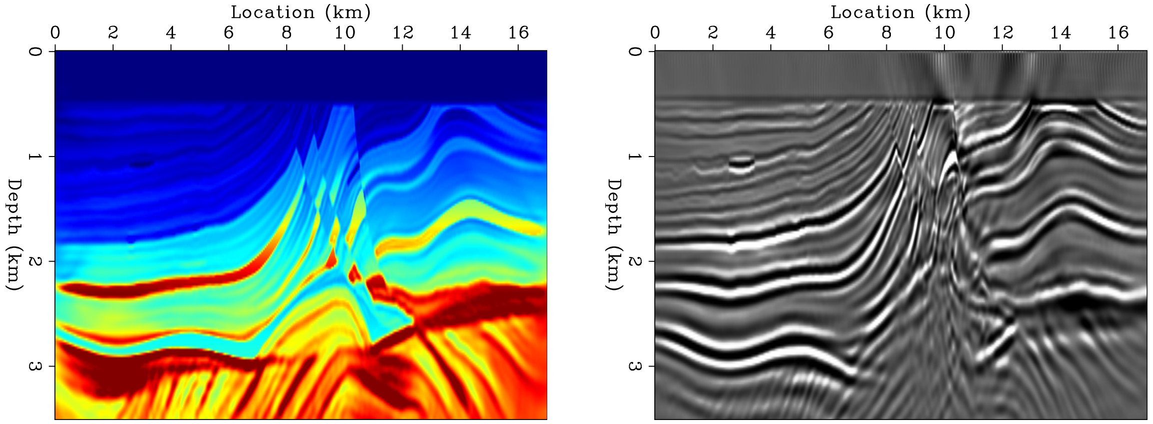


(a)

(b)

Figure 9: (a) The initial velocity model. (b) The RTM image using the initial velocity model. [**CR**] ali1/. 14,16

We followed the workflow proposed by Biondi and Almomin (2014) by first running TFWI on the low frequencies (up to 10Hz) and then running FWI on the higher frequencies. The inverted model is shown in Figure 10(a) after only 15 TFWI iterations and 30 FWI iterations. The RTM image obtained using the inverted model is shown in Figure 10(b). We see that TFWI successfully inverted the velocity model with high accuracy except around the edges where the illumination is insufficient.



(a)

(b)

Figure 10: (a) The inverted velocity model. (b) The RTM image using the inverted velocity model. [**CR**] ali1/. 15,17

CONCLUSIONS

We introduced a modified inversion algorithm that significantly improved the convergence rate of TFWI. This was achieved by introducing a new regularization term that separates the amplitude and phase fitting in the inversion and by changing the enhancing operator to reduce the kinematic artifacts. The synthetic tests show the fast convergence of the new algorithm even when starting from a far initial model.

ACKNOWLEDGMENTS

We would like to thank the Stanford Exploration Project affiliate companies for financial support. Almomin would like to thank Saudi Aramco for supporting his graduate studies at Stanford University.

REFERENCES

Almomin, A. and B. Biondi, 2013, Tomographic full waveform inversion (TFWI) by successive linearizations and scale separations: SEP-Report, **149**, 51–58.

———, 2014, Preconditioned tomographic full waveform inversion by wavelength continuation : SEP-Report, **152**, 11–18.

Biondi, B. and A. Almomin, 2014, Efficient and robust waveform-inversion workflow: Tomographic FWI followed by FWI : SEP-Report, **152**, 1–10.

Full-waveform inversion based on nonlinear conjugate gradient method, Gauss-Newton method and full Newton method

Yinbin Ma, Musa Maharramov, Biondo Biondi

ABSTRACT

Full-waveform inversion (FWI) generates a high-resolution subsurface model. Robust local minimization algorithms are required for FWI because the objective function is highly nonlinear. In this paper we compare the nonlinear conjugate gradient method, Gauss-Newton method and full Newton method for FWI. These methods use the gradient of objective function and application of Hessian on a model perturbation vector, which can be calculated efficiently with the adjoint-state methods. Numerical results suggest Newton-type methods resolve fine structure faster than the nonlinear conjugate gradient method in terms of number of wave propagation.

INTRODUCTION

Full-waveform inversion (Tarantola, 1984; Virieux and Operto, 2009) is a challenging technique that estimates the high-resolution subsurface model by minimizing the mismatch between observed data and synthetic data. The first order derivative is usually needed for FWI, and the Hessian is used in Newton-type methods. It is known that the adjoint-state method is an efficient method to compute the Frechet derivative (Tromp et al., 2005; Plessix, 2006) and the Hessian (Fichtner, 2011; Fichtner and Trampert, 2011) for FWI.

In this paper, we implement and compare FWI using three methods: nonlinear conjugate gradient (CG) method (Nocedal and Wright, 2006; Maharramov and Biondi, 2013), Gauss-Newton method and full Newton method (Pratt et al., 1998). Nonlinear CG method requires the gradient at each iteration, and the model update is computed based on the current gradient and previous gradient. For Gauss-Newton method and full Newton method, the model update is calculated by applying the approximated inverse of the Hessian to the gradient.

Simple synthetic models are used to test our implementation of FWI algorithms. A model with pinch-out structures is constructed to test the vertical resolution of different methods. A model with wells is used to test the horizontal resolution and the ability to recover vertical structures. Numerical results suggest that Newton-type methods converge faster than nonlinear CG method in terms of number of wave propagation.

We create synthetic model with fine structures based on the Society of Exploration Geophysicists/European Association of Geoscientists and Engineers (SEG/EAGE) model. Preliminary results suggest that Newton-type methods recover deeper structure better than the nonlinear CG method after the same number of wave propagation.

We briefly discuss the different rate of convergence in the last section.

METHOD

Full-waveform inversion for acoustic media

We use the least-squares misfit function for FWI in the time domain, as follows:

$$J(m) = \frac{1}{2}\sum_r \int_0^T ||S_r\mathbf{u} - d_r||_2^2 dt, \tag{1}$$

where S_r is the sampling operator for the receivers, d_r is the observed data at the receiver $\mathbf{r}$, and $\mathbf{u}$ is the synthetic pressure wavefield.

The pressure field $\mathbf{u}$ is computed using the acoustic approximation of wave equation with a non-constant density, as follows:

$$\begin{cases} [\frac{1}{K}\partial_t^2 - \tilde{\nabla}\cdot(\frac{1}{\rho}\nabla)]\mathbf{u} & = \mathbf{f} \\ \mathbf{u}(\mathbf{r}, t=0) & = 0 \\ \partial_t \mathbf{u}(\mathbf{r}, t=0) & = 0, \end{cases} \tag{2}$$

where K is the bulk modulus, ρ is the density, and $\mathbf{f}$ is the source wavefield. Numerically, we solve equation 2 in the time domain using staggered-grid finite difference method, starting from $t = 0$ to maximum recording time $t = \mathrm{T}$.

We formulate FWI as a nonlinear optimization problem and solve it iteratively. The Frechet derivate is estimated at each iteration based on the adjoint-state methods, which requires the correlation of wavefield $\mathbf{u}$ with the receiver wavefield λ, which is defined as follows:

$$\begin{cases} [\frac{1}{K}\partial_t^2 - \nabla\cdot(\frac{1}{\rho}\tilde{\nabla})]\lambda & = \mathbf{d_{res}} \\ \lambda(\mathbf{r}, t=T) & = 0 \\ \partial_t \lambda(\mathbf{r}, t=T) & = 0, \end{cases} \tag{3}$$

where $\mathbf{d_{res}}$ is the difference between synthetic data and observed data. The receiver wavefield is computed backward in time, starting from the maximum recording time $t = \mathbf{T}$ to $t = 0$.

The acoustic wave equation 2 is used to model non-constant density media. We implemented multi-parameter FWI in acoustic media. However, in this paper ρ is assumed to be constant, and we estimate bulk modulus K, which is related to velocity by, as follows:

$$v = \sqrt{\frac{K}{\rho}}. \tag{4}$$

Nonlinear conjugate gradient method

We implement the conjugate gradient method to minimize the objective function in equation 1. A nonlinear CG method generates a sequence of estimated modulus K_i, $i \geq 0$, starting from initial guess K_0.

Assume at iteration i, we have obtained the estimated model K_i. For the next iteration, the gradient is calculated at $\mathbf{g_i} = \mathbf{g}(K_i) \equiv \frac{\partial J}{\partial K_i}$. From the current gradient $\mathbf{g_i}$ and previous gradient $\mathbf{g_{i-1}}$, we get the search direction ΔK_i using the Fletcher Reeves formula and Polak Ribire formula(Nocedal and Wright, 2006; Maharramov and Biondi, 2013).

The local minimum of objective $J(K)$ in the vicinity of K_i is estimated with the line search approach, as follows:

$$\alpha_i = \mathrm{argmin}_\alpha J(K_i + \alpha \Delta K_i), \tag{5}$$

and we update the model with,

$$K_{i+1} = K_i + \alpha_i \Delta K_i. \tag{6}$$

We do not solve the line search problem in equation 5 exactly because the evaluation of $J(K_i + \alpha\Delta K_i)$ is expensive. We want an approximated solution with a few iterations. The line search process is terminated once the Wolfe condition (Nocedal and Wright, 2006) is satisfied, as follows:

$$J(K_i + \alpha\Delta K_i) \leq J(K_i) + c_1 \alpha \mathbf{g_i}^T \Delta K_i \tag{7}$$

$$|\nabla J(K_i + \alpha \Delta K_i)^T \Delta K_i| \leq c_2 |\mathbf{g_i}^T \Delta K_i|, \tag{8}$$

where $0 < c_1 < c_2 < 1$, and ∇J is the gradient of objective function.

We continue this process, until the value of the objective function or the norm of gradient is below a certain threshold.

The gradient used in the nonlinear CG method is computed with the adjoint-state methods (Fichtner, 2011; Fichtner and Trampert, 2011), shown in the following:

$$\mathbf{g}(\mathbf{K}) = \frac{\partial J}{\partial K} = -\int_0^T \lambda \frac{1}{K^2} \partial_t^2 \mathbf{u} dt, \tag{9}$$

where the source wavefield $\mathbf{u}$ satisfies equation 2, and receiver wavefield λ is obtained from equation 2. The details of derivation are shown in the appendix, with multiparameter model:

$$\mathbf{m} = \begin{bmatrix} 1/K \\ 1/\rho \end{bmatrix}. \tag{10}$$

Full Newton method and Gauss-Newton method

Consider the second order expansion of objective function, as follows:

$$J(K + \Delta K) = J(K) + \left(\frac{\partial J}{\partial K}\right)^T \Delta K + \frac{1}{2} \Delta K \mathbf{H}(K) \Delta K + O(\Delta K^3), \tag{11}$$

where the full Hessian $\mathbf{H}$ is the second order derivative of the objective function,

$$\begin{aligned}\mathbf{H} &= \frac{\partial^2 J}{\partial K^2}\\ &= \left(\frac{\partial J}{\partial K}\right)^T \left(\frac{\partial J}{\partial K}\right) + \mathbf{d_{res}}^T \frac{\partial^2 (S_r \mathbf{u})}{\partial K^2}.\end{aligned}$$

The Hessian is reduced to Gauss-Newton Hessian $\mathbf{H}_{\mathrm{GN}}$ by dropping the second term, as follows:

$$\mathbf{H}_{\mathrm{GN}} = \left(\frac{\partial J}{\partial K}\right)^T \left(\frac{\partial J}{\partial K}\right) \tag{12}$$

For the full Newton method and Gauss-Newton method, suppose at iteration $i > 0$, we have obtained modulus K_i. For the next iteration, the gradient $\mathbf{g}(K_i)$ is calculated with equation 9. We scale the gradient by applying the inverse of Hessian, as follows:

$$\Delta K_i = -\mathbf{H}^{-1}\mathbf{g}(K_i). \tag{13}$$

Then we use the line search approach to find the estimate model for the next iteration,

$$\begin{aligned}\alpha_i &= \mathrm{argmin}_\alpha J(K_i + \alpha \Delta K_i) &(14)\\ K_{i+1} &= K_i + \alpha_i \Delta K_i. &(15)\end{aligned}$$

One advantage of the full Newton method is that when K_i is close to the solution, we will have $\alpha_i \to 1$.

The inverse of Hessian $-\mathbf{H}^{-1}g(K_i)$ is approximated iteratively, which requires the action of Hessian on a model perturbation vector $\mathbf{H}\Delta K$. The action of Hessian can be computed efficiently using the adjoint-state method, as follows:

$$\mathbf{H}\Delta K = -\frac{\Delta K}{K^2}\int_0^T (\mu_a(x,t) + \mu_b(x,t))\partial_t^2 \mathbf{u}(x,t)dt - \frac{\Delta K}{K^2}\int_0^T \lambda(x,t)\partial_t^2 \delta\mathbf{u}_1(x,t)dt, \tag{16}$$

with source wavefield $\mathbf{u}$, λ from the forward wave propagation, and receiver wavefield μ_a, μ_b and λ from the backward wave propagation, as follows:

$$\begin{cases} [m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]\mathbf{u} & = \mathbf{f}\\ [\frac{1}{K}\partial_t^2 - \tilde{\nabla}\cdot(\frac{1}{\rho}\nabla)]^*\lambda & = \sum_r S_r^*(S_r\mathbf{u} - d_r)\\ [m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]\delta\mathbf{u_1} & = [\Delta m_K\partial_t^2 - \tilde{\nabla}\cdot(\Delta m_\rho\nabla)]\mathbf{u}\\ [m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]^*\mu_{\mathbf{a}} & = S_r^* S_r \delta\mathbf{u_1}\\ [m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]^*\mu_{\mathbf{b}} & = [\Delta m_K\partial_t^2 - \tilde{\nabla}\cdot(\Delta m_\rho\nabla)]^*\lambda, \end{cases} \tag{17}$$

where we use $m_K \equiv \frac{1}{K}$ and $m_\rho \equiv \frac{1}{\rho}$. The corresponding model perturbations are $\Delta m_K = -\frac{\Delta K}{K^2}$ and $\Delta m_\rho = -\frac{\Delta\rho}{\rho^2}$.

For the Gauss-Newton Hessian, the action can be expressed as the following:

$$\mathbf{H}_{\mathrm{GN}}\Delta K = -\frac{\Delta K}{K^2}\int_0^T \mu_{\mathbf{a}}(\mathbf{r},t)\partial_t^2\mathbf{u}(\mathbf{r},t)dt. \tag{18}$$

We put the detailed derivation of the Hessian in the appendix with multiparameter model $\mathbf{m} = (1/K, 1/\rho)$. The approximation of the inverse of the Hessian is expensive both in computation time and memory.

RESULTS

Simple synthetic models

We initially test the nonlinear CG method, Gauss-Newton method and full Newton method on simple synthetic models. We use 10 sources with $300\,m$ spacing between neighboring sources. The depth of the sources is $650\,m$, and the receivers are at the same depth as the sources.

Each model is solved with nonlinear CG method for 50 iterations (150 wave propagation for pinch-out model and 166 wave propagation for model contains vertical wells). Four iterations of the full Newton method (160 wave propagation) and Gauss-Newton method (120 wave propagation) are applied, and at each iteration we solve $\Delta K = H^{-1}(-g)$ for 10 CG steps.

The first model Figure 1 contains 5 thin layers with $60 -$ meter(m) thickness, and pinch-out structure. The model has velocity $2400\,m/s$ with source wavelet centered at approximately 30 Hz. The layers are within the resolution of FWI, while the thin part of the pinch-out is below the resolution. In Figure 3, we plot the objective function versus the number of wave propagation for different methods. Newton-type methods converge faster than the nonlinear CG method in terms of computational cost. We can see the inversion results for the pinch-out model in Figure 2, after 120 wave propagation. Newton-type methods reconstruct sharper boundary comparing with the nonlinear CG method.

The second model shown in Figure 4 contains two vertical wells with $100\,m$ and $30\,m$ width. The objective function versus the number of wave propagation for different methods is shown in Figure 6. We can see the inversion results in Figure 5. For the Newton-type methods, the location of the vertical well is identified properly, while the parameter within the well cannot be properly estimated. On the other hand, the nonlinear CG method has not resolved the deeper part of the $100\,m$ well and has not clearly identified the $30\,m$ well.

SEG/SEAM model

The full Newton method and Gauss-Newton method use the quadratic approximation of the objective function, and ideally they should have better resolution comparing with the nonlinear CG method. They should converge faster when we are close to the true model. In this subsection, we create a synthetic model based on SEG/EAGE model as in Figure 7. We use 10 sources with $300\,m$ spacing between neighboring sources. The depth of the sources is $650\,m$, and the receivers are at the same depth as the sources.

We run the nonlinear CG method for 100 iterations. We apply Full Newton method and Gauss-Newton method for 4 iterations; and at each iteration, we solve $\Delta K = H^{-1}(-g)$ for 15 CG steps. The objective versus the number of wave propagation is shown in Figure 9. We can see that the Newton-type methods converge faster than the nonlinear CG method,

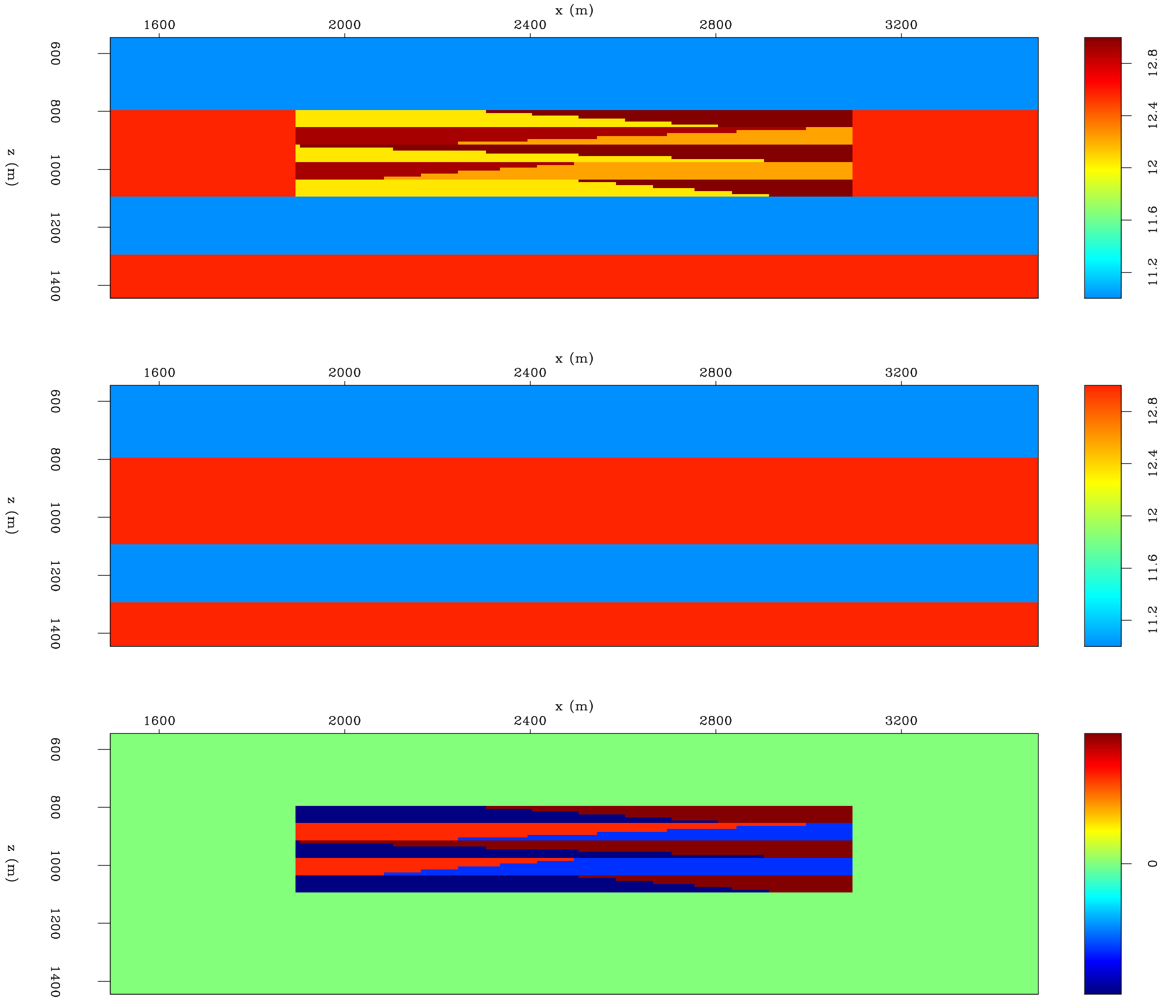
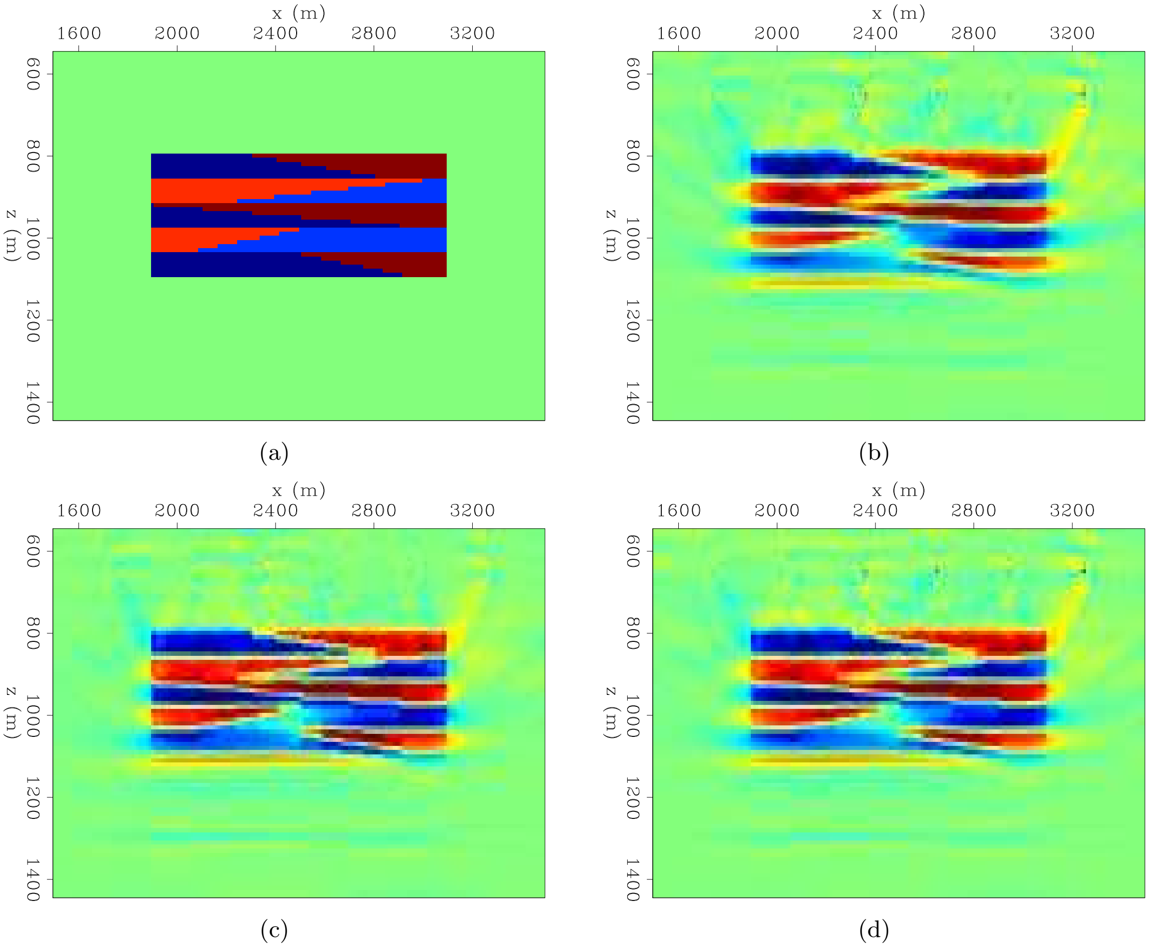


Figure 1: The top panel shows the true modulus model. The middle panel shows the starting modulus model. The bottom panel shows the difference between the true model and starting model.[**ER**] yinbin1/. Pinch



(a)

(b)

(c)

(d)

Figure 2: Panel (a) shows the difference between the true model and starting model. Panel (b) is the model update with the nonlinear CG method. Panel (c) is the model update with the Gauss-Newton method. Panel (d) is the model update with the full Newton method.[**CR**] yinbin1/. PinchDiff,NCGPinch,GNewtonPinch,NewtonPinch

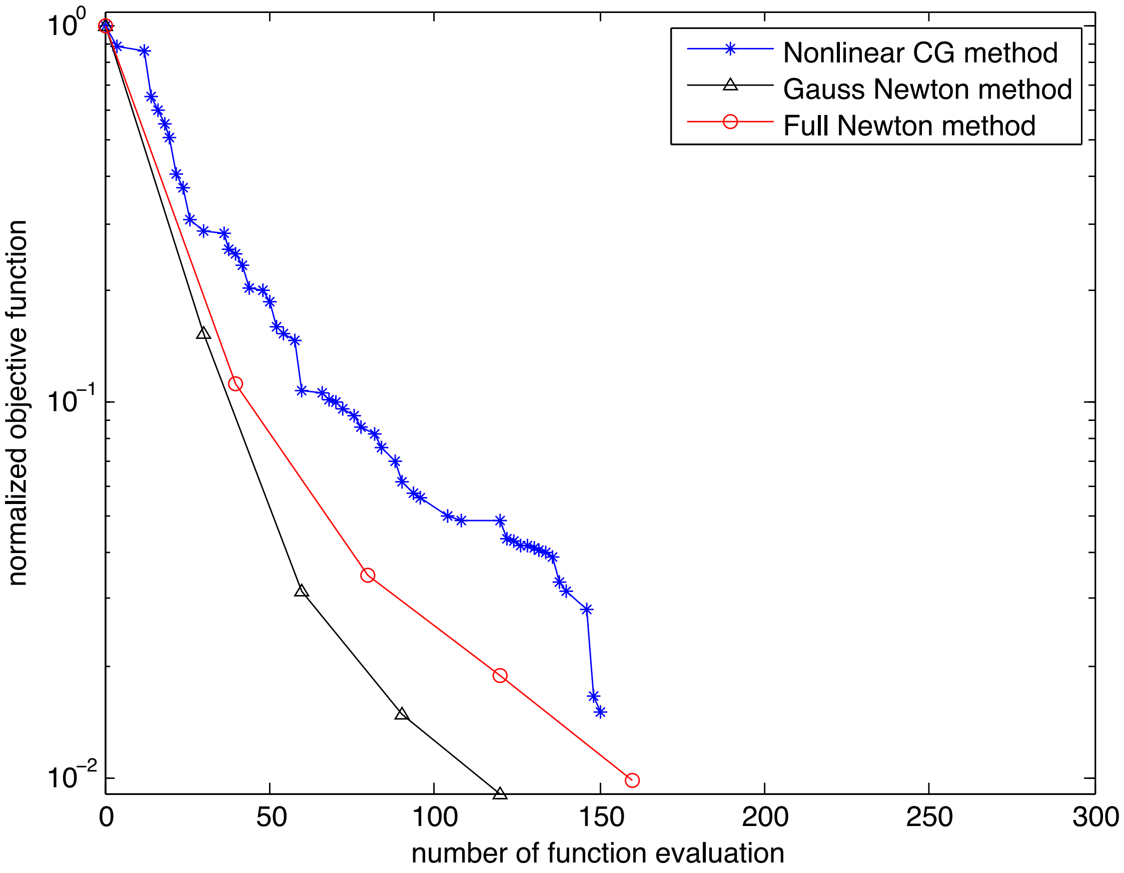


Figure 3: The normalized objective function versus the number of wave propagation. The blue curve represents the nonlinear CG method, black curve represents the Gauss-Newton method and red curve represents the full Newton method.[**CR**] yinbin1/. OBJPinch

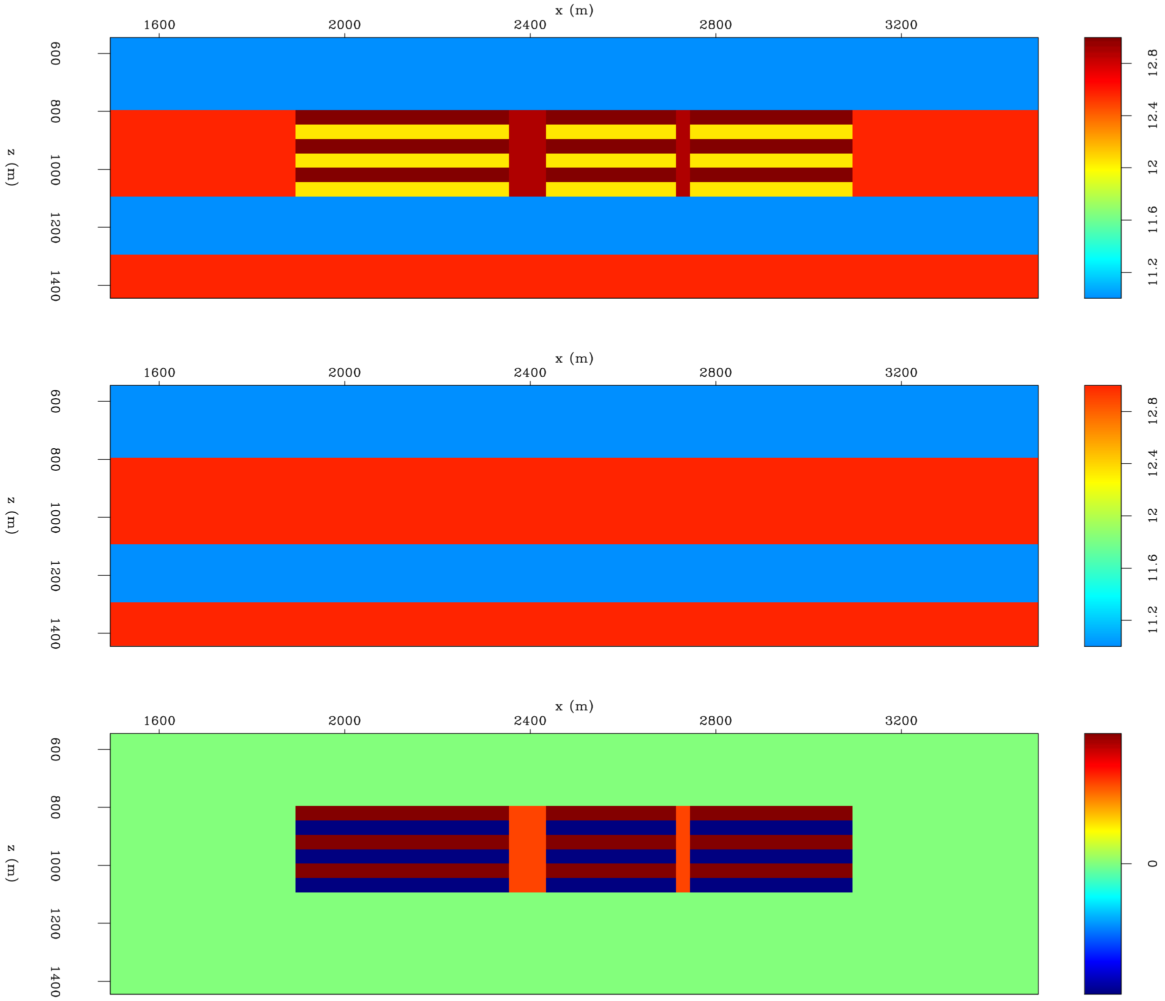


Figure 4: The top panel shows the true modulus model. The middle panel shows the starting modulus model. The bottom panel shows the difference between the true model and starting model.[**CR**] yinbin1/. Pipe

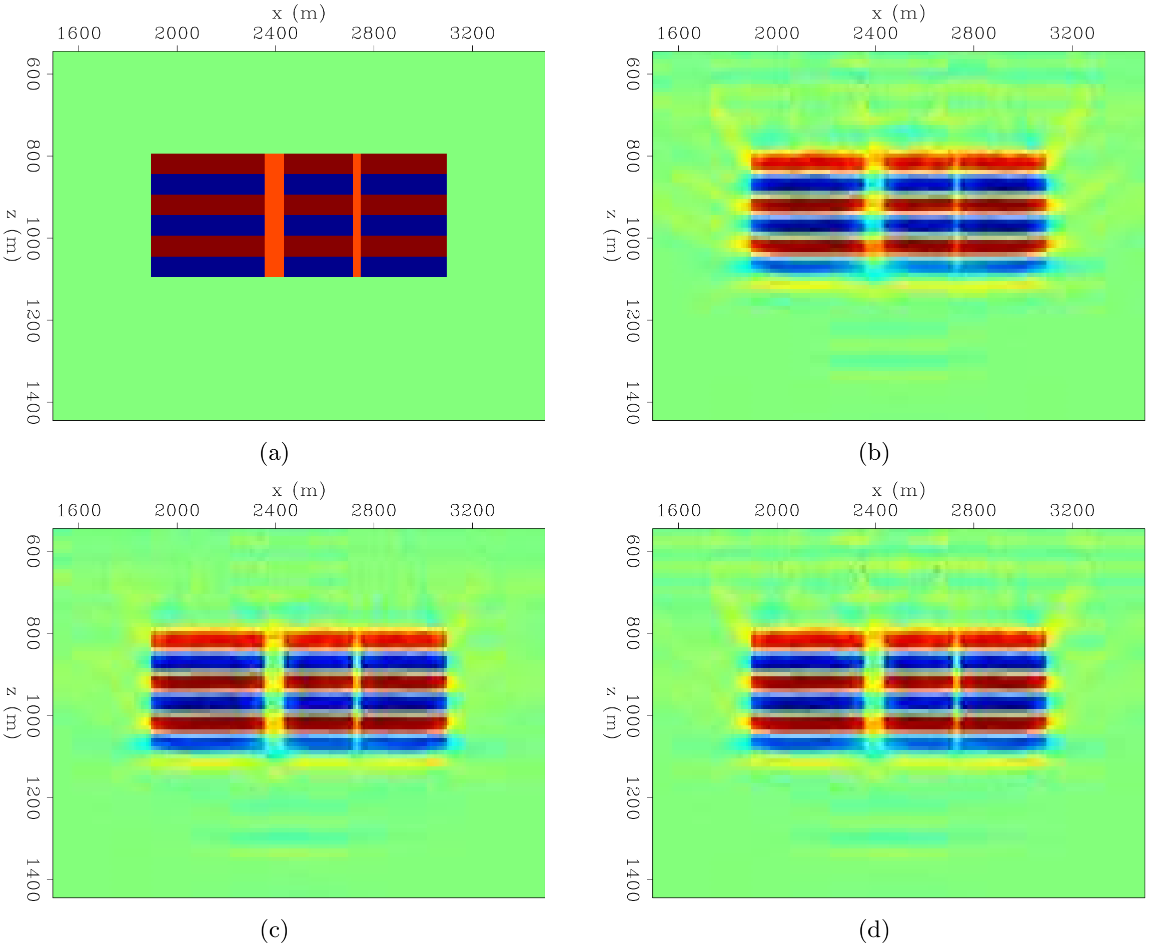


Figure 5: Panel (a) shows the difference between the true model and starting model. Panel (b) is the model update with the nonlinear CG method. Panel (c) is the model update with the Gauss-Newton method. Panel (d) is the model update with the full Newton method.[**CR**] yinbin1/. PipeDiff,NCGPipe,GNewtonPipe,NewtonPipe

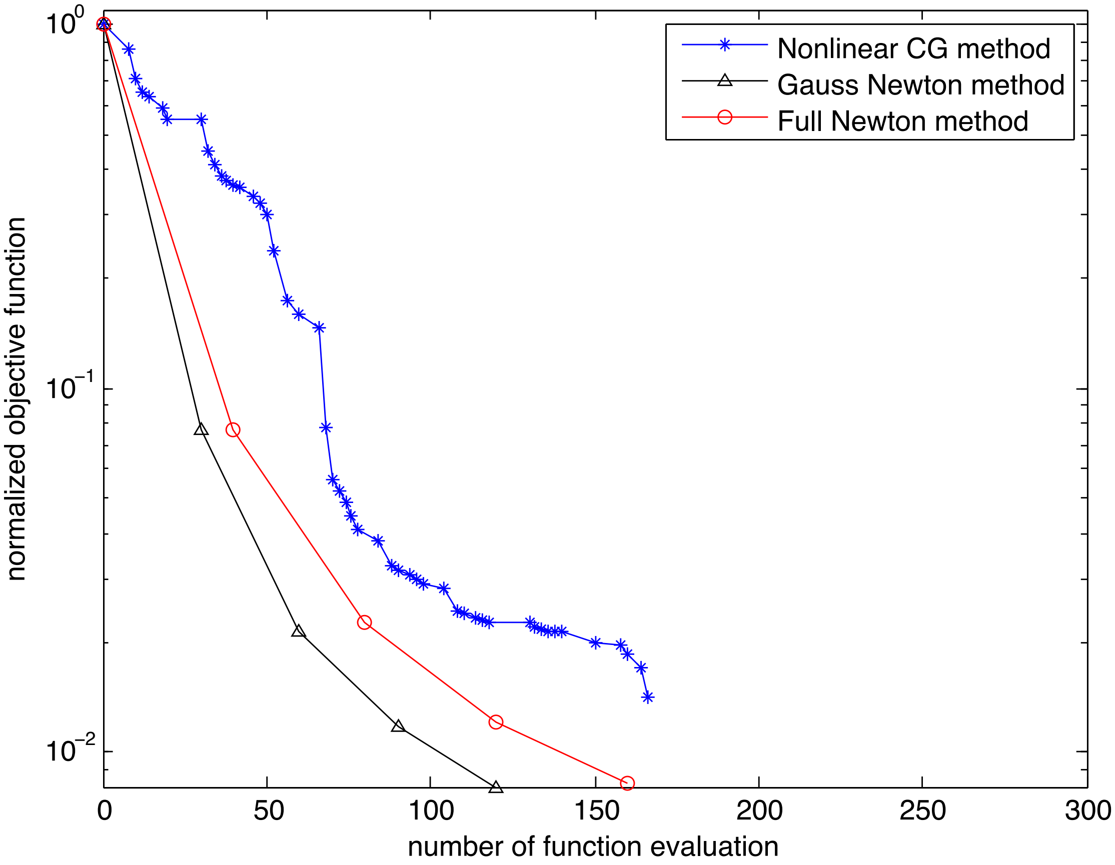


Figure 6: The normalized objective function versus the number of wave propagation. The blue curve represents the nonlinear CG method, black curve represents the Gauss-Newton method and red curve represents the full Newton method.[**CR**] yinbin1/. OBJPipe

at the same computational cost.

We plot the inversion results after 180 wave propagation (56 iterations for the nonlinear CG method, 4 iterations of Gauss-Newton method, and 3 iterations of full Newton method) in Figure 8. We can see significant difference below $1100m$. The nonlinear CG method has not resolved the structure, and the Newton-type methods have reconstructed the fine structure. The Gauss-Newton method has smaller residual comparing with the full Newton method, and we will discuss it in the next section.

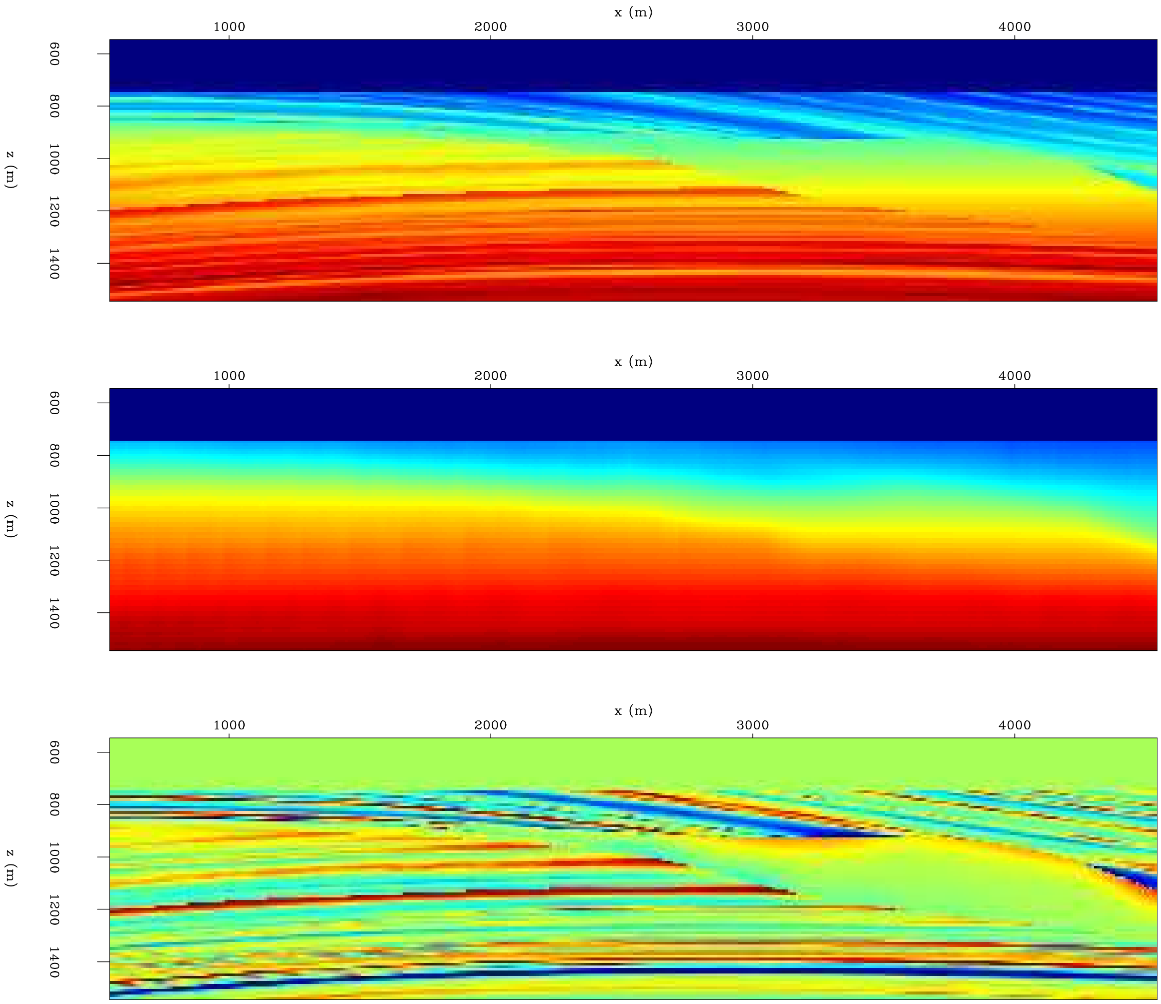


Figure 7: The top panel shows the true modulus model. The middle panel shows the starting modulus model. The bottom panel shows the difference between the true model and starting model.[**ER**] yinbin1/. ModSEAM

DISCUSSION ON RATE OF CONVERGENCE

We show that Newton-type methods converge faster than nonlinear CG method, in terms of number of the wave propagation.

The nonlinear CG method and the Newton-type methods scale the gradient to get the

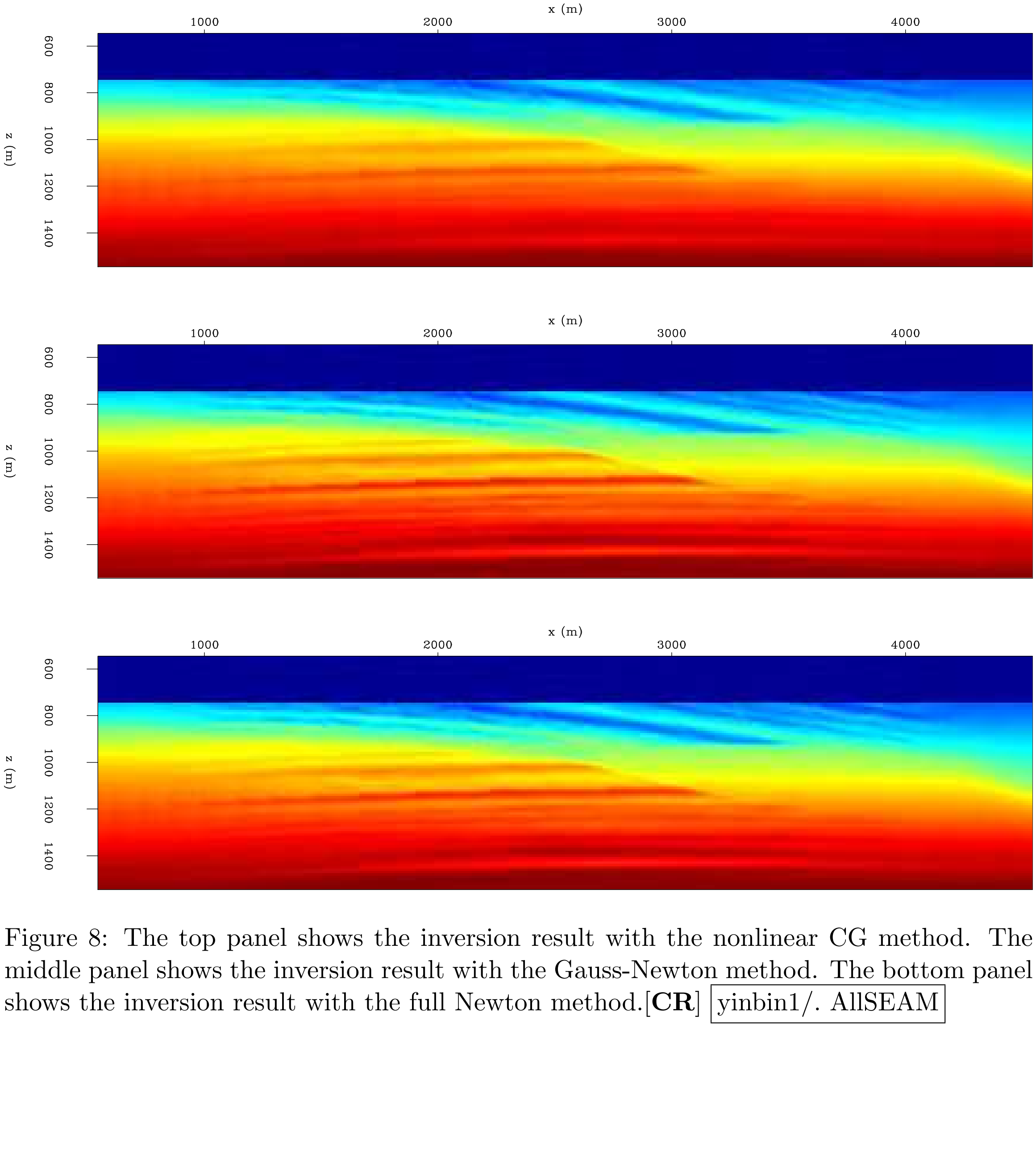


Figure 8: The top panel shows the inversion result with the nonlinear CG method. The middle panel shows the inversion result with the Gauss-Newton method. The bottom panel shows the inversion result with the full Newton method.[**CR**] yinbin1/. AllSEAM

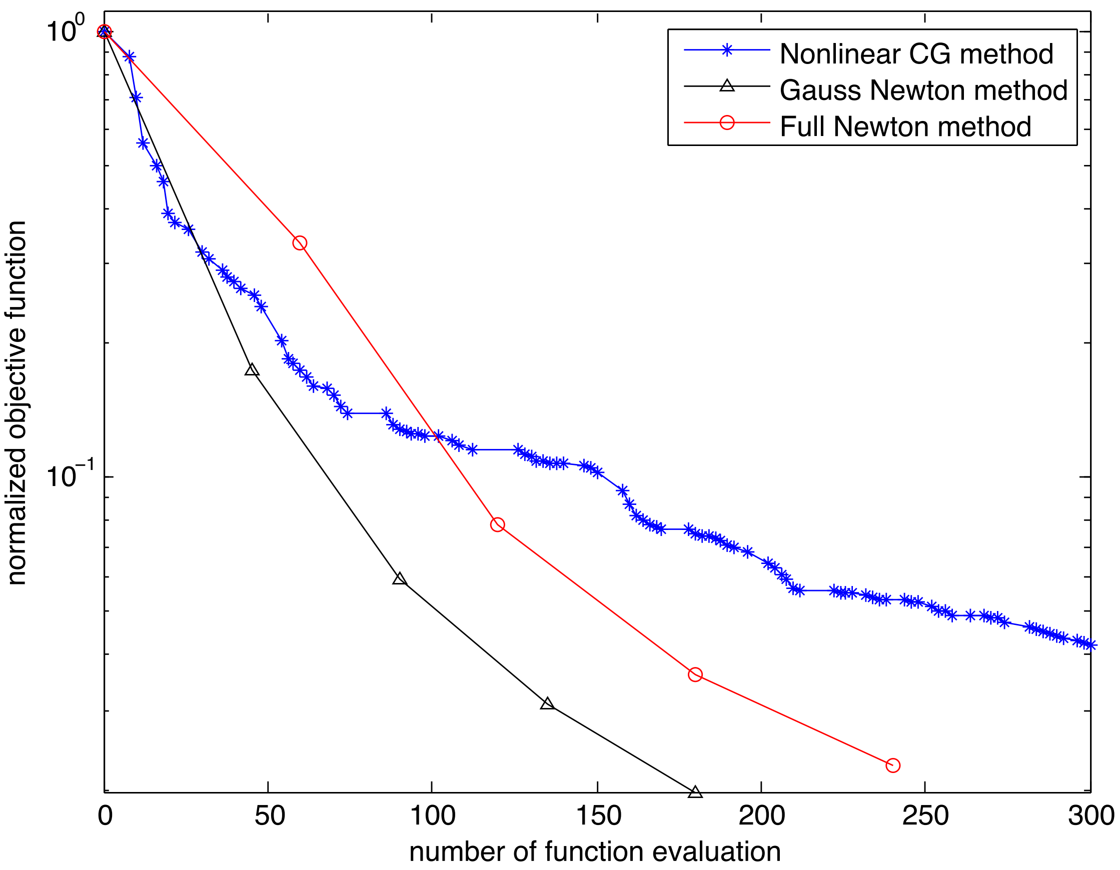


Figure 9: The normalized objective function versus the number of wave propagation. The blue curve represents the nonlinear CG method, black curve represents the Gauss-Newton method and red curve represents the full Newton method.[**CR**] yinbin1/. OBJSEAM

search direction. It is interesting to examine how the gradient is scaled.

We take the estimated model K_i from the nonlinear CG method at $i = 30$ iteration. The difference between the true model and K_i, which is the ideal search direction, is shown in Figure 10(e). The gradient $\mathbf{g}_i$ is shown in Figure 10(a). The gradient can not be directly used as updating direction because the shallow component is much stronger than the deep component. This property of gradient result in the failure of the steepest descent method in FWI.

In Figure 10(b), we show the correction with nonlinear CG method,

$$\Delta K_i = -\mathbf{g}_i + \beta_i \mathbf{g}_{i-1}, \tag{19}$$

where β_i is obtained from Fletcher Reeves formula. After we scale the gradient in CG method, the search direction has more weight on the deep part. However, the component of search direction in the shallow part are not corrected properly. As the objective function is sensitive to shallow perturbation, we would not expect much decrease in the objective function.

For the Newton-type methods, we apply $\Delta K_{i,\mathrm{N}ewton} = \mathbf{H}^{-1}(-\mathbf{g}_i)$ for 15 CG steps, and the results are shown in Figure 10(c) and 10(d). The shallow component of the search direction is significantly different for the nonlinear CG method and the Newton-type methods as shown in Figure 10. The search direction for Newton-type methods approximate the ideal search direction reasonably well.

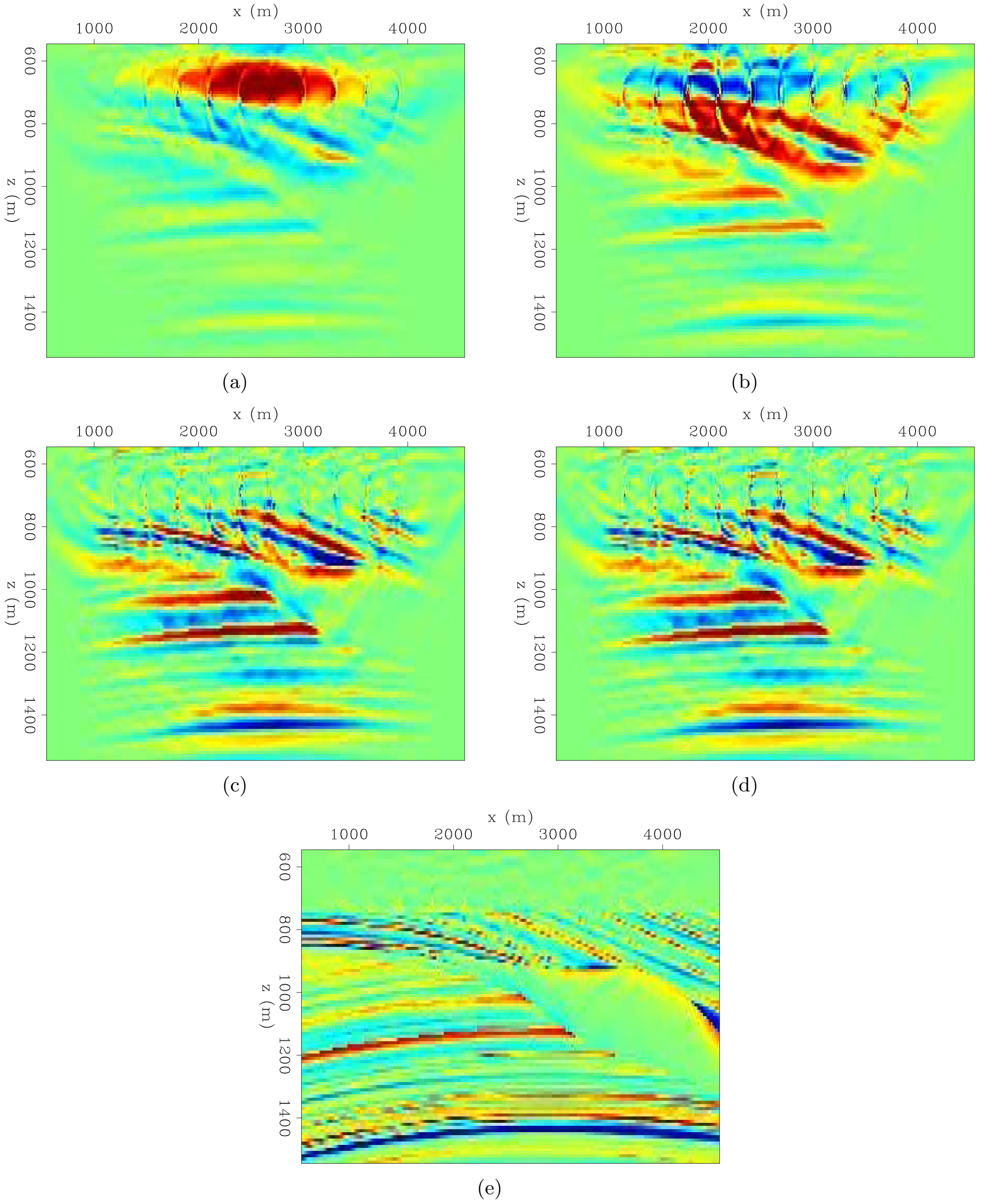
We have pointed out in the previous section, that the Gauss-Newton method converges faster than the Newton method at equivalent number of wave propagation. One reason is that each application full Hessian uses 4 wave propagation, and each application of the Gauss-Newton Hessian uses 3 wave propagation. Solver $\mathbf{H}^{-1}(-\mathbf{g})$ is therefore more expensive for the full Newton method. Another reason might be that because we use single precision float number for wave propagation, the CG algorithm for $\mathbf{H}^{-1}(-\mathbf{g})$ converges slower than the ${\mathbf{H}_{\mathbf{GN}}}^{-1}(-\mathbf{g})$ in terms of CG steps.

CONCLUSION

In this paper, we implemented nonlinear CG method, Gauss-Newton method and full Newton method for solving FWI. We computed the gradient and Hessian with the adjoint-state method. Numerical results suggest Newton-type methods resolve fine structures better than the nonlinear CG method when the computational cost is the same.

ACKNOWLEDGEMENT

We would like to thank Ali Almomin, Robert Clapp for their help on the implementation of full-waveform inversion. We would also like to thank Ettor Biondi, Gustavo Alves, Ossian O'Reilly for useful suggestions on staggered-grid finite difference.



(a)

(b)

(c)

(d)

(e)

Figure 10: Panel (a) shows the gradient $\mathbf{g}(K_i)$, with K_i estimated from the nonlinear CG method at $i = 30$. Panel (b) is the model update ΔK_i with the nonlinear CG method. Panel (c) is the model update with the Gauss-Newton method after applying $\mathbf{H}_{\mathrm{GN}}^{-1}(-\mathbf{g}(K_i))$. Panel (d) is the model update with the full Newton method after applying $\tilde{\mathbf{H}}^{-1}(-\mathbf{g}(K_i))$. Panel (e) is the ideal model updating direction, which is the difference between true modulus model and current modulus model.[**CR**] yinbin1/. SEAMGK,SEAMdK,SEAMGNdK,SEAMNdK,SEAMTruedK

APPENDIX A: FULL-WAVEFORM INVERSION GRADIENT AND HESSIAN

In this appendix, we derive the Frechet derivative (Tromp et al., 2005; Plessix, 2006) and the Hessian (Fichtner, 2011) for the FWI objective function. We use acoustic approximation of wave equation with non-constant density. For simplicity, we define our model parameter as:

$$m_K \equiv \frac{1}{K}, \tag{20}$$

$$m_\rho \equiv \frac{1}{\rho}, \tag{21}$$

$$m \equiv \begin{bmatrix} m_K \\ m_\rho \end{bmatrix}. \tag{22}$$

The corresponding model perturbation can be expressed as follows:

$$\Delta m_K = -\frac{\Delta K}{K^2}, \tag{23}$$

$$\Delta m_\rho = -\frac{\Delta \rho}{\rho^2}, \tag{24}$$

The wave equation in acoustic media with non-constant density is shown in the following:

$$\begin{cases} [\frac{1}{K}\partial_t^2 - \tilde{\nabla}\cdot(\frac{1}{\rho}\nabla)]\mathbf{u} & = \mathbf{f} \\ \mathbf{u}(\mathbf{r}, t=0) & = 0 \\ \partial_t \mathbf{u}(\mathbf{r}, t=0) & = 0, \end{cases} \tag{25}$$

where K is the bulk modulus, ρ is the density, and $\mathbf{f}$ is the source wavefield. We use operator L to represent the operator:

$$L \equiv [\frac{1}{K}\partial_t^2 - \tilde{\nabla}\cdot(\frac{1}{\rho}\nabla)]. \tag{26}$$

The adjoint of wave equation used to compute receiver wavefield can be written as follows:

$$\begin{cases} [\frac{1}{K}\partial_t^2 - \nabla\cdot(\frac{1}{\rho}\tilde{\nabla})]\lambda & = \mathbf{d_{res}} \\ \lambda(\mathbf{r}, t=T) & = 0 \\ \partial_t \lambda(\mathbf{r}, t=T) & = 0, \end{cases} \tag{27}$$

where $\mathbf{d_{res}}$ is the difference between synthetic data and observed data. The receiver wavefield is computed backward in time, starting from maximum recording time $t = \mathbf{T}$ to $t = 0$.

Frechet derivative

For arbitrary model perturbation, we have

$$\left\langle \Delta m, \frac{\partial J}{\partial m} \right\rangle_{\mathbf{M}} = \left\langle \frac{\partial u}{\partial m}\Delta m, \frac{\partial J}{\partial u} \right\rangle_{\mathbf{U}}, \tag{28}$$

where the subscript **M** indicates the inner product is done in the model space, and and subscript **U** indicates the wavefield space.

Perturb the wave equation, we have

$$\left(\frac{\partial L}{\partial m}\Delta m\right)u + L\frac{\partial u}{\partial m}\Delta, m = 0 \tag{29}$$

Substitute equation (29) into equation (28), we get

$$\begin{aligned}\left\langle \Delta m, \frac{\partial J}{\partial m}\right\rangle_{\mathbf{M}} &= -\left\langle L^{-1}(\frac{\partial L}{\partial m}\Delta m)u, \frac{\partial J}{\partial u}\right\rangle_{\mathbf{U}} && (30)\\ &= -\left\langle (\frac{\partial L}{\partial m}\Delta m)u, L^{-*}\frac{\partial J}{\partial u}\right\rangle_{\mathbf{U}}. && (31)\\ & && (32)\end{aligned}$$

Introducing the receiver wavefield,

$$L^*\lambda = \frac{\partial J}{\partial u}, \tag{33}$$

with the explicit formula shown in equation (27). We have,

$$\left\langle \Delta m, \frac{\partial J}{\partial m}\right\rangle_{\mathbf{M}} = -\left\langle (\frac{\partial L}{\partial m}\Delta m)u, \lambda\right\rangle_{\mathbf{U}}. \tag{34}$$

For acoustic wave equation,

$$\begin{aligned}\left\langle \Delta m, \frac{\partial J}{\partial m}\right\rangle_{\mathbf{M}} &= -\left\langle [\delta m_K \partial_t^2 - \tilde{\nabla}\cdot(\delta m_\rho \nabla)]u, \lambda\right\rangle_{\mathbf{U}} && (35)\\ &= -\left\langle \delta m_K, \int_0^T \lambda \partial_2^t u dt\right\rangle_{\mathbf{M}} + \left\langle \Delta m_\rho, \int_0^T (\tilde{\nabla}^*\lambda)\cdot(\nabla u)dt\right\rangle_{\mathbf{M}}. && (36)\end{aligned}$$

Thus,

$$\frac{\partial J}{\partial m} = \begin{bmatrix} \frac{\partial J}{\partial m_K} \\ \\ \frac{\partial J}{\partial m_\rho}\end{bmatrix} = \begin{bmatrix} -\int_0^T \lambda \partial_t^2 u dt \\ \\ \int_0^T (\tilde{\nabla}^*\lambda)\cdot(\nabla u)dt\end{bmatrix}. \tag{37}$$

The Frechet derivative with respect to (K, ρ) is,

$$\frac{\partial J}{\partial(K,\rho)} = \begin{bmatrix} \frac{\partial J}{\partial K} \\ \\ \frac{\partial J}{\partial \rho}\end{bmatrix} = \begin{bmatrix} \frac{1}{K^2}\int_0^T \lambda \partial_t^2 u dt \\ \\ -\frac{1}{\rho^2}\int_0^T (\tilde{\nabla}^*\lambda)\cdot(\nabla u)dt\end{bmatrix} \tag{38}$$

Full Hessian and Gauss-Newton Hessian

We are interested in the action of Hessian:

$$H(\Delta m_1, \Delta m_2) = \langle \Delta m_2, H\Delta m_1\rangle_{\mathbf{M}}, \tag{39}$$

where Δm_1 and Δm_2 are two model perturbation. We use shorthand notation for the bilinear form in this subsection:

$$F(x_1, x_2) \equiv \langle x_1, F x_2 \rangle . \tag{40}$$

Our goal is to find an expression,

$$\langle \Delta m_2, H \Delta m_1 \rangle_{\mathbf{M}} = \langle \delta m_2, A(\Delta m) \rangle_{\mathbf{M}} , \tag{41}$$

for arbitrary Δm_2, where $A(\Delta m_1)$ does not depend on Δm_2 and can be computed efficiently. Then we claim $A(\Delta m_1)$ is equivalent to the action of Hessian applied to Δm_1.

Expand the bilinear form in equation (39),

$$\langle \Delta m_2, H \Delta m_1 \rangle_{\mathbf{M}} \quad = \quad \left\langle \frac{\partial u}{\partial m} \Delta m_2, \frac{\partial^2 J}{\partial u^2} \frac{\partial u}{\partial m} \Delta m_1 \right\rangle_{\mathbf{U}} \tag{42}$$

$$+ \left\langle \frac{\partial J}{\partial u}, \frac{\partial^2 u}{\partial m^2} (\Delta m_1, \Delta m_2) \right\rangle_{\mathbf{U}} . \tag{43}$$

The bilinear form for Gauss-Newton Hessian can be obtained by neglecting line (43),

$$\langle \Delta m_2, H_{GN} \Delta m_1 \rangle_{\mathbf{M}} = \left\langle \frac{\partial u}{\partial m} \Delta m_2, \frac{\partial^2 J}{\partial u^2} \frac{\partial u}{\partial m} \Delta m_1 \right\rangle_{\mathbf{U}} . \tag{44}$$

where H_{GN} represent the Gauss-Newton Hessian.

In order to eliminate $\frac{\partial^2 u}{\partial m^2}(\Delta m_1, \Delta m_2)$ in equation (42 and 43) which depends on Δm_2, we need the first order and second order perturbation of the wave equation:

$$(\frac{\partial L}{\partial m} \Delta m_1) u + L \frac{\partial u}{\partial m} \Delta m_1 = 0, \tag{45}$$

$$(\frac{\partial L}{\partial m} \Delta m_2) u + L \frac{\partial u}{\partial m} \Delta m_2 = 0, \tag{46}$$

and,

$$(\frac{\partial^2 L}{\partial m^2}(\Delta m_1, \Delta m_2))u + (\frac{\partial L}{\partial m} \Delta m_1)(\frac{\partial u}{\partial m} \Delta m_2) + (\frac{\partial L}{\partial m} \Delta m_2) \frac{\partial u}{\partial m} \Delta m_1 + L \frac{\partial^2 u}{\partial m^2}(\Delta m_1, \Delta m_2) = 0. \tag{47}$$

Substitute equation (47) into equation (42), we get,

$$\langle \Delta m_2, H \Delta m_1 \rangle_{\mathbf{M}} \quad = \quad \left\langle \frac{\partial u}{\partial m} \Delta m_2, \frac{\partial^2 J}{\partial u^2} \frac{\partial u}{\partial m} \Delta m_1 \right\rangle_{\mathbf{U}} \tag{48}$$

$$+ \left\langle \frac{\partial J}{\partial u}, \frac{\partial^2 u}{\partial m^2} (\Delta m_1, \Delta m_2) \right\rangle_{\mathbf{U}} . \tag{49}$$

Define $\delta u_1 \equiv \frac{\partial u}{\partial m} \Delta m_1$ which can be computed from Born approximation, as follows:

$$L(m) \Delta u_1 = -(\frac{\partial L}{\partial m} \delta m_1) u, \tag{50}$$

Use equation (33 and 50), and after some algebra,

$$\begin{aligned}
\langle \Delta m_2, H\Delta m_1\rangle_{\mathbf{M}} &= \left\langle \frac{\partial u}{\partial m}\Delta m_2, -(\frac{\partial L}{\partial m}\Delta m_1)^*\lambda + \frac{\partial^2 J}{\partial u^2}\delta u_1 \right\rangle_{\mathbf{U}} \\
&- \left\langle \lambda, (\frac{\partial^2 L}{\partial m^2}(\Delta m_1, \Delta m_2))u \right\rangle_{\mathbf{U}} \\
&- \left\langle \lambda, (\frac{\partial L}{\partial m}\Delta m_2)\Delta u_1 \right\rangle_{\mathbf{U}}.
\end{aligned}$$

We then eliminate $\frac{\partial u}{\partial m}\Delta m_2$ term using equation (46), and get,

$$\begin{aligned}
\langle \Delta m_2, H\Delta m_1\rangle_{\mathbf{M}} &= -\left\langle (\frac{\partial L}{\partial m}\Delta m_2)u, L^{-*}\{-(\frac{\partial L}{\partial m}\Delta m_1)^*\lambda + \frac{\partial^2 J}{\partial u^2}\Delta u_1\} \right\rangle_{\mathbf{U}} && (51)\\
&- \left\langle \lambda, (\frac{\partial^2 L}{\partial m^2}(\Delta m_1, \Delta m_2))u \right\rangle_{\mathbf{U}} && (52)\\
&- \left\langle \lambda, (\frac{\partial L}{\partial m}\Delta m_2)\Delta u_1 \right\rangle_{\mathbf{U}} && (53)\\
&= -\left\langle \mu_a + \mu_b, (\frac{\partial L}{\partial m}\Delta m_2)u \right\rangle_{\mathbf{U}} && (54)\\
&- \left\langle \lambda, (\frac{\partial^2 L}{\partial m^2}(\Delta m_1, \Delta m_2))u \right\rangle_{\mathbf{U}} && (55)\\
&- \left\langle \lambda, (\frac{\partial L}{\partial m}\Delta m_2)\Delta u_1 \right\rangle_{\mathbf{U}}, && (56)
\end{aligned}$$

where we define,

$$\begin{aligned}
L(m)^*\mu_a &= -(\frac{\partial L}{\partial m}\Delta m_1)^*\lambda, && (57)\\
L(m)^*\mu_b &= \frac{\partial^2 J}{\partial u^2}\Delta u_1. && (58)
\end{aligned}$$

Thus, our solution for Hessian in the general case:

$$\begin{aligned}
\langle \Delta m_2, H\Delta m_1\rangle_{\mathbf{M}} &= \left\langle \mu, (\frac{\partial L}{\partial m}\Delta m_2)u \right\rangle_{\mathbf{U}} && (59)\\
&+ \left\langle \lambda, (\frac{\partial^2 L}{\partial m^2}(\Delta m_1, \Delta m_2))u \right\rangle_{\mathbf{U}} && (60)\\
&+ \left\langle \lambda, (\frac{\partial L}{\partial m}\Delta m_2)\delta u_1 \right\rangle_{\mathbf{U}}, && (61)
\end{aligned}$$

with each wavefield computed as follows:

$$
\begin{aligned}
L(m)u &= f, && (62)\\
L(m)^*\lambda &= \frac{\partial J}{\partial u}, && (63)\\
L(m)\delta u_1 &= -(\frac{\partial L}{\partial m}\delta m_1)u, && (64)\\
L(m)^*\mu_a &= \frac{\partial^2 J}{\partial u^2}\delta u_1, && (65)\\
L(m)^*\mu_b &= -(\frac{\partial L}{\partial m}\delta m_1)^*\lambda, && (66)
\end{aligned}
$$

For acoustic wave equation, we get,

$$
H\Delta m = \begin{bmatrix} -\int_0^T(\mu_a(x,t)+\mu_b(x,t))\partial_t^2 u(x,t)dt - \int_0^T \lambda(x,t)\partial_t^2\Delta u_1(x,t)dt \\ \\ \int_0^T \tilde{\nabla}^*(\mu_a(x,t)+\mu_b(x,t)\cdot\nabla u(x,t)dt + \int_0^T \tilde{\nabla}^*\lambda(x,t)\nabla\Delta u_1(x,t)dt, \end{bmatrix}. \tag{67}
$$

with the 2 forward wave propagation and 2 backward wave propagation,

$$
\begin{aligned}
[m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]u &= f, && (68)\\
[\frac{1}{K}\partial_t^2 - \tilde{\nabla}\cdot(\frac{1}{\rho}\nabla)]^*\lambda &= \sum_r S_r^*(S_r u - d_r), && (69)\\
[m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]\delta u_1 &= -[\Delta m_K\partial_t^2 - \tilde{\nabla}\cdot(\Delta m_\rho\nabla)]u, && (70)\\
[m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]^*\mu_a &= S_r^* S_r\delta u_1, && (71)\\
[m_K\partial_t^2 - \tilde{\nabla}\cdot(m_\rho\nabla)]^*\mu_b &= -[\Delta m_K\partial_t^2 - \tilde{\nabla}\cdot(\Delta m_\rho\nabla)]^*\lambda. && (72)
\end{aligned}
$$

The Hessian with respect to (K,ρ) therefore can be written as:

$$
H\begin{bmatrix} \Delta K \\ \\ \Delta\rho \end{bmatrix} = \begin{bmatrix} \frac{\Delta K}{K^2}\int_0^T(\mu_a(x,t)+\mu_b(x,t))\partial_t^2 u(x,t)dt + \frac{\Delta K}{K^2}\int_0^T \lambda(x,t)\partial_t^2\Delta u_1(x,t)dt \\ \\ -\frac{\Delta\rho}{\rho^2}\int_0^T \tilde{\nabla}^*(\mu_a(x,t)+\mu_b(x,t)\cdot\nabla u(x,t)dt - \frac{\Delta\rho}{\rho^2}\int_0^T \tilde{\nabla}^*\lambda(x,t)\nabla\Delta u_1(x,t)dt \end{bmatrix}. \tag{73}
$$

For Gauss-Newton Hessian, we can follow similar derivation, and write:

$$
H_{GN}\Delta m = \begin{bmatrix} -\int_0^T \mu_a(x,t)\partial_t^2 u(x,t)dt \\ \\ \int_0^T \tilde{\nabla}^*\mu_a(x,t)\cdot\nabla u(x,t)dt \end{bmatrix}. \tag{74}
$$

The difference between the full Hessian and the Gauss-Newton Hessian can be expressed as follows:

$$
(H - H_{GN})\Delta m = \begin{bmatrix} -\int_0^T \mu_b(x,t)\partial_t^2 u(x,t)dt - \int_0^T \lambda(x,t)\partial_t^2\Delta u_1(x,t)dt \\ \\ \int_0^T \tilde{\nabla}^*\mu_a(x,t)dt + \int_0^T \tilde{\nabla}^*\lambda(x,t)\nabla\Delta u_1(x,t)dt \end{bmatrix}. \tag{75}
$$

REFERENCES

Fichtner, A., 2011, Full seismic waveform modeling and inversion: Springer.

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

Maharramov, M. and B. Biondi, 2013, Simultaneous time-lapse full waveform inversion: SEP 150, 1–8.

Nocedal, J. and S. J. Wright, 2006, Numerical Optimization: Springer.

Plessix, R. E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Pratt, R. G., C. Shin, and G. J. Hicks, 1998, Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

Tromp, J., C. Tape, and Q. Liu, 2005, Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels: Geophysical Journal International, **160**, 195–216.

Virieux, J. and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1.

Processing of continuous data at Apache Forties for seismic interferometry

Jason P. Chang

ABSTRACT

I process continuous data from the Apache Forties data set for use in studies relating to passive seismic imaging. Spectrograms from multiple nodes indicate the presence of strong, naturally-occurring seismic energy at frequencies below 1 Hz, as well as the presence of seismic energy above 4 Hz that originates from the oil platform. After ignoring times of active seismic shooting, I perform passive seismic interferometry on over 2 days of hydrophone and vertical geophone data. For frequencies between 1.00 and 1.25 Hz, there are indications of Scholte-wave energy. However, the expected wavelengths (approximately 150 m/s) at these frequencies are likely comparable to the aperture of the array (approximately 400 m), which compromises the reliability of the arrivals. For frequencies between 3.00 and 7.00 Hz, there are no clear indications of interface waves. Asymmetry of seismic energy in the virtual source gathers suggests that platform vibrations dominate the ambient noise field at high frequencies.

INTRODUCTION

Passive seismic interferometry suggests that cross-correlating simultaneous recordings of ambient noise at two receivers can estimate the Green's function between them (Lobkis and Weaver, 2001; Wapenaar, 2004; Snieder, 2004). These estimated Green's functions are typically dominated by low-frequency interface waves, which have in turn been used for tomographic imaging at primarily the regional and continental scales (e.g., Shapiro et al., 2005; Sabra et al., 2005; Lin et al., 2007). Given the success of these studies, efforts have shifted to applying this technique to recordings from exploration-scale seismic arrays. Focusing on frequencies in the microseism band (between 0.1 and 2.0 Hz), de Ridder and Dellinger (2011) and Mordret et al. (2013) processed ocean-bottom cable recordings of ambient noise to recover interstation Scholte waves at the Valhall oil field. Energy in the microseism band is generated by the natural interaction of water waves in seas and oceans. It is different from microseismic energy, which originates from small earthquakes typically induced by production and drilling activity. The microseism waves are used to tomographically image the reservoir overburden. Additionally, the dense node spacing associated with exploration-scale arrays (approximately 100 m) allows for seismic interferometry to be applied to ambient noise at higher frequencies. For instance, Mordret et al. (2013) showed that platform energy dominates the ambient noise field at frequencies above 2 Hz at the Valhall field and used that energy to estimate shear-wave velocity along a 2D line.

In this study, we apply the seismic interferometry technique to recordings of ambient noise from a small array of ocean-bottom nodes (OBNs) in the North Sea. Although this is four-component data, I only examine recordings from the hydrophone and vertical-component geophone. First, I provide an overview of the OBN array and its continuous recordings. Second, I compute spectrograms to examine how the ambient seismic noise field changes over time. Finally, I cross-correlate recordings to estimate interstation Green's functions over two frequency ranges. For the frequency range focused on the microseism band, there are indications of coherent seismic events. However, they are likely not reliable enough for use in a tomographic procedure. For frequencies between 3.00 and 7.00 Hz, there are no clear seismic events. However, the asymmetry of energy in the correlations suggests that the platform dominates the ambient noise field at these frequencies.

CONTINUOUS RECORDINGS FROM FORTIES

The Forties data set, provided to SEP by the Apache Corporation, consists of three groups of OBNs centered at three different platforms in the North Sea. The nodes were deployed as part of an active seismic survey aimed at imaging shallow gas pockets that could pose potential drilling hazards. Unlike some other reservoirs in the North Sea, the Forties reservoir is not easily compacted and is unlikely to generate significant activity of this type. The nodes were continuously recording, and because active seismic shooting had to be suspended for a couple of days due to rough weather conditions, there is enough ambient noise data to attempt to perform seismic interferometry. Furthermore, stormy weather conditions tend to increase the upper limit of the microseism energy (de Ridder and Dellinger, 2011), which is ideal in this case given the limited aperture of the OBN clusters.

For this study, we focus on hydrophone and vertical geophone components of one of the three groups of OBNs. Referred to as Bravo, this cluster of OBNs consists of 52 4-component nodes arranged in a hexagonal shape around a platform (Figure 1). The average node spacing is 50 m, and the array aperture is approximately 400 m. Each node continuously records for approximately 4 days at 2 ms sampling and is located roughly 120 m below the sea surface. To prepare the recordings for ambient noise processing, I round the start times up to the nearest hour and round the end times to the nearest hour. These times will not be the same for all nodes, as they were not all deployed simultaneously.

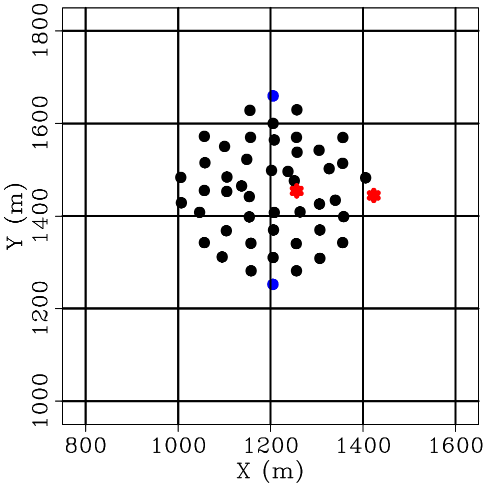


Figure 1: Map of OBNs in the Bravo array. The platform is in the center of the array. Red dots indicate virtual source locations. They are also the nodes at which I compute spectrograms. Blue dots indicate receiver locations for the traces shown later in the study. [**ER**] jason1/. map-total

SPECTROGRAMS

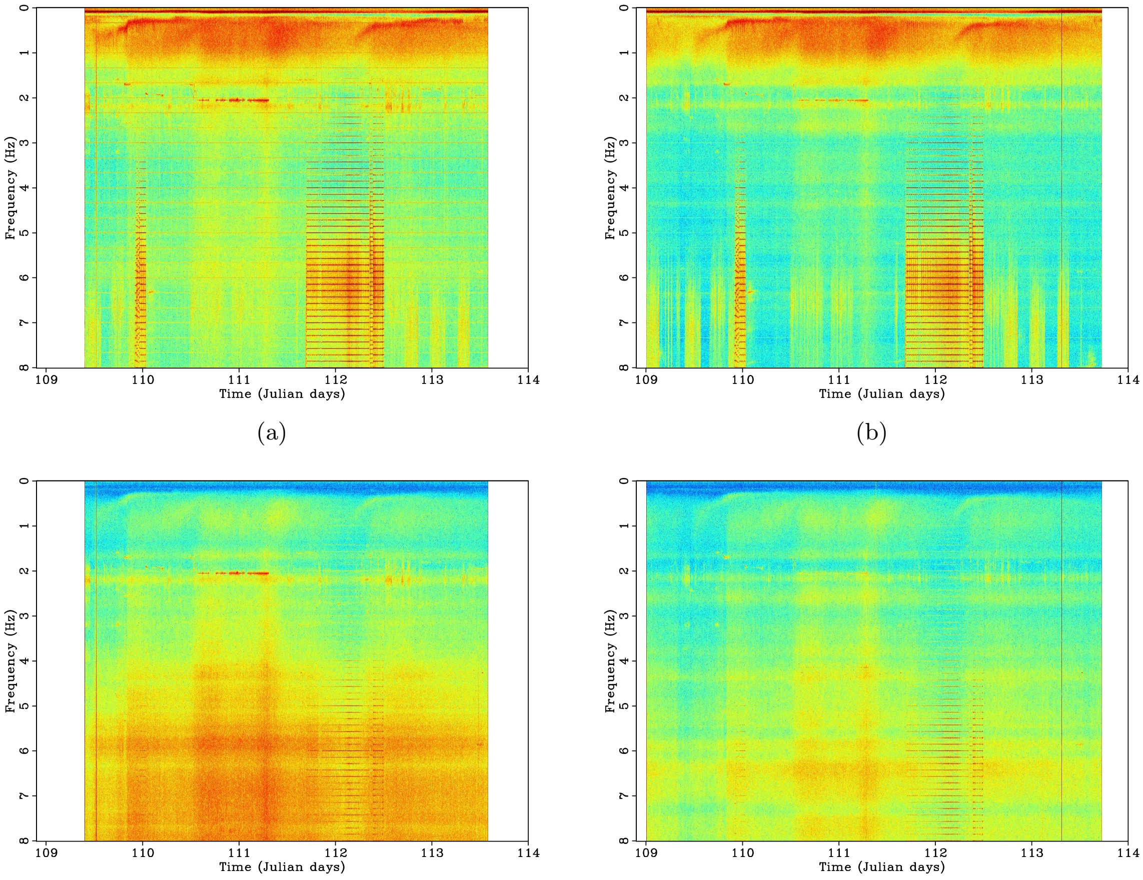
To investigate the change in frequency content over time in the continuous recordings, I compute spectrograms of the hydrophone and vertical-component geophone at two nodes (indicated by red dots in Figure 1). I divide the recordings into 5-minute time segments with 50% overlap, compute the power spectrum of each segment, then plot them side by side. To enhance important features, I plot the spectrograms on a log scale for frequencies up to 8 Hz.

Figures 2(a) and 2(b) reveal the spectrograms for the hydrophone component at the western and eastern nodes (Figure 1), respectively. Colors in both plots have the same clip. The repeating horizontal patterns at regular frequency intervals on day 110 and days 111.5 to 112.5 indicate periods of active seismic shooting. Of more interest to this study is the clear microseism energy found at frequencies below 1 Hz. As mentioned in the previous section, active seismic shooting was suspended due to rough weather conditions. This is visible starting around day 110, where the microseism energy increases dramatically right before active shooting is halted. The microseism energy remains strong for a couple of days before active shooting resumes and appears to be nearly the same strength at both node locations, which is expected for low-frequency energy in a small array. At frequencies above 2 Hz, there tends to be more energy when the node is closer to the platform. Similar patterns are observed in the spectrograms of the vertical-component geophones (Figures 2(c) and 2(d), corresponding to the western and eastern nodes, respectively). As before, the microseism energy at both node locations is nearly the same strength. It is also apparent that there is more energy at higher frequencies when the node is near the platform than when the node is further away from the platform. Therefore, it appears that the energy source is well-distributed in space at low frequencies and localized at the platform at higher frequencies.

In all the spectrograms, there is a streak of energy at 2 Hz that persists for roughly 1 day, starting at day 110.5. The cause of high energy at this specific frequency is unknown. It is observed in all spectrograms, suggesting it is due to a physical phenomenon, perhaps related to the platform, and not due to instrument malfunction. Overall, there appears to be significant microseism and platform energy in the data, which is promising for passive seismic interferometry.

SEISMIC INTERFEROMETRY

To perform seismic interferometry on ambient noise, the data records must be synchronized in time. Therefore, all node recordings are truncated to the latest start time and the earliest end time amongst the group of nodes. Because three of the nodes had anomalously short recording periods, I work with 49 out of the 52 possible nodes in Bravo. I then divide the remaining continuous recordings into 2-hour time windows with 50% overlap for two reasons. First, cross-correlating shorter, overlapping records and then summing them leads to more rapid convergence than when cross-correlating the full recordings themselves (Seats et al., 2012). This is particularly advantageous in this case, where there is only about 4 days of data. Second, dividing the long recordings into shorter ones makes it easier to ignore times when active seismic shooting is occurring. If the maximum absolute value of a trace (bandpassed for frequencies between 4 and 50 Hz) is above a certain threshold, all traces for that time segment are avoided when summing the correlation results. I end up using 52



(a)

(b)

(c)

(d)

Figure 2: Spectrograms computed at two nodes (indicated by red dots in Figure 1). Hydrophone component for (a) western node and (b) eastern node. Vertical component for (c) western node and (d) eastern node. The western node is close to the platform. Warmer colors indicate higher power. Colors in all plots have the same clip. [**ER**] jason1/. sp1H,sp2H,sp1V,sp2V

out of the 74 possible 2-hour segments, leading to over 2 days of ambient noise data.

Passive seismic interferometry is performed by cross-correlating the recordings of ambient seismic noise at two receivers. Under ideal conditions, the result is an estimate of the Green's function between the two receivers, and it is typically dominated by interface waves because they are the strongest wave modes in ambient seismic noise (Wapenaar et al., 2010). Here, I apply a correlation procedure adapted from Bensen et al. (2007). To compensate for expected variations in source amplitude over time and to broaden the bandwidth of the correlation result without changing the phase information, I perform spectral whitening on the traces prior to cross-correlation. In the frequency domain, the procedure is expressed as:

$$[G(x_B, x_A, \omega) + G^*(x_B, x_A, \omega)] = \left\langle \left(\frac{U(x_B, \omega)}{\{|U(x_B, \omega)|\}} \right) \left(\frac{U^*(x_A, \omega)}{\{|U(x_A, \omega)|\}} \right) \right\rangle, \qquad (1)$$

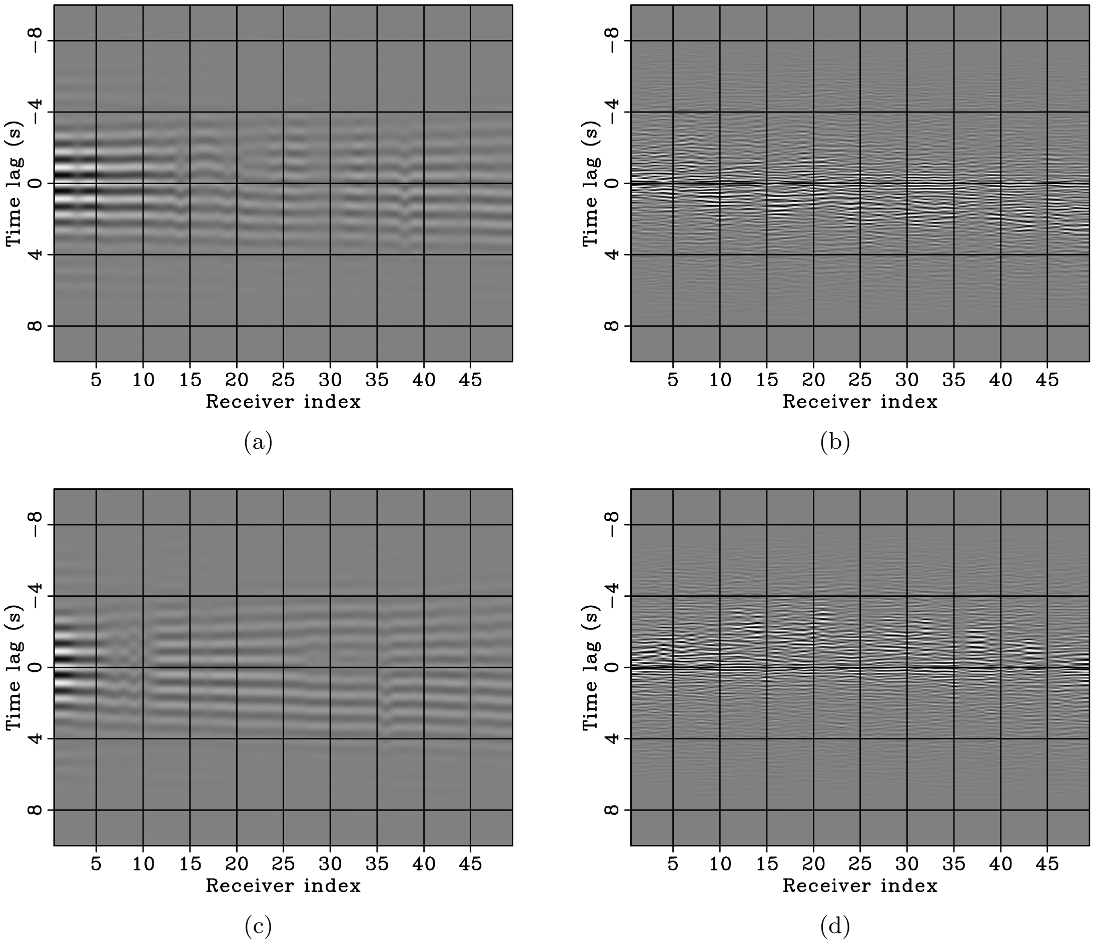
where G is the Green's function between two receiver locations (x_A, x_B), $U(x, \omega)$ is the spectrum of the wavefield at a given receiver location x, $*$ is the complex conjugate, $\langle \cdot \rangle$ is the time-averaged ensemble, $| \cdot |$ is the magnitude of the spectrum, and $\{\cdot\}$ is a 0.003 Hz running window average used for normalizing the signal. This procedure is equivalent to calculating the cross-coherence between two traces.

By cross-correlating one node with all other nodes, I am able to create a virtual source gather. Here, I create virtual source gathers centered at the two node locations indicated by red dots in Figure 1. For both the hydrophone and vertical-geophone components, I investigate two frequency ranges: 1.00–1.25 Hz (for the microseism energy) and 3.00–7.00 Hz (for the platform energy). Gathers are sorted by distance between the virtual source and receiver. Figure 3 shows the hydrophone component, while Figure 4 shows the vertical geophone component. For the lower frequency range, there are clear coherent events from both virtual source locations that are potentially Scholte waves (Figures 3(a) and 3(c) for the hydrophone; Figures 4(a) and 4(c) for the vertical geophone). However, for the higher frequency range, there appears to be no clear coherent seismic energy (Figures 3(b) and 3(d) for the hydrophone; Figures 4(b) and 4(d) for the vertical geophone). The gathers are asymmetric, though, which indicates an uneven source distribution.

For a closer look at the lower-frequency waveforms, Figure 5 plots the correlation results between the virtual sources and receivers shown by colored dots in Figure 1. I only show the hydrophone component because the vertical geophone component looks very similar. When the distance between the virtual source and receiver is larger (Figures 5(c) and 5(d)), apparent Scholte-wave arrivals at negative and positive time lags are clear. When the distance between the virtual source and receiver is smaller (Figures 5(a) and 5(b)), then there appears to only be one Scholte-wave arrival near zero time lag.

DISCUSSION

Results of passive seismic interferometry indicate the shortcomings of the array when attempting to extract Scholte waves from ambient noise in the microseism band. Given the investigation frequencies and general North Sea Scholte-wave phase velocities between 200 and 400 m/s at those frequencies (de Ridder and Dellinger, 2011; Mordret et al., 2013), it is very likely that the wavelengths of the estimated Scholte waves (optimistically around 150 m) are comparable to the aperture of the Bravo array (approximately 400 m). This



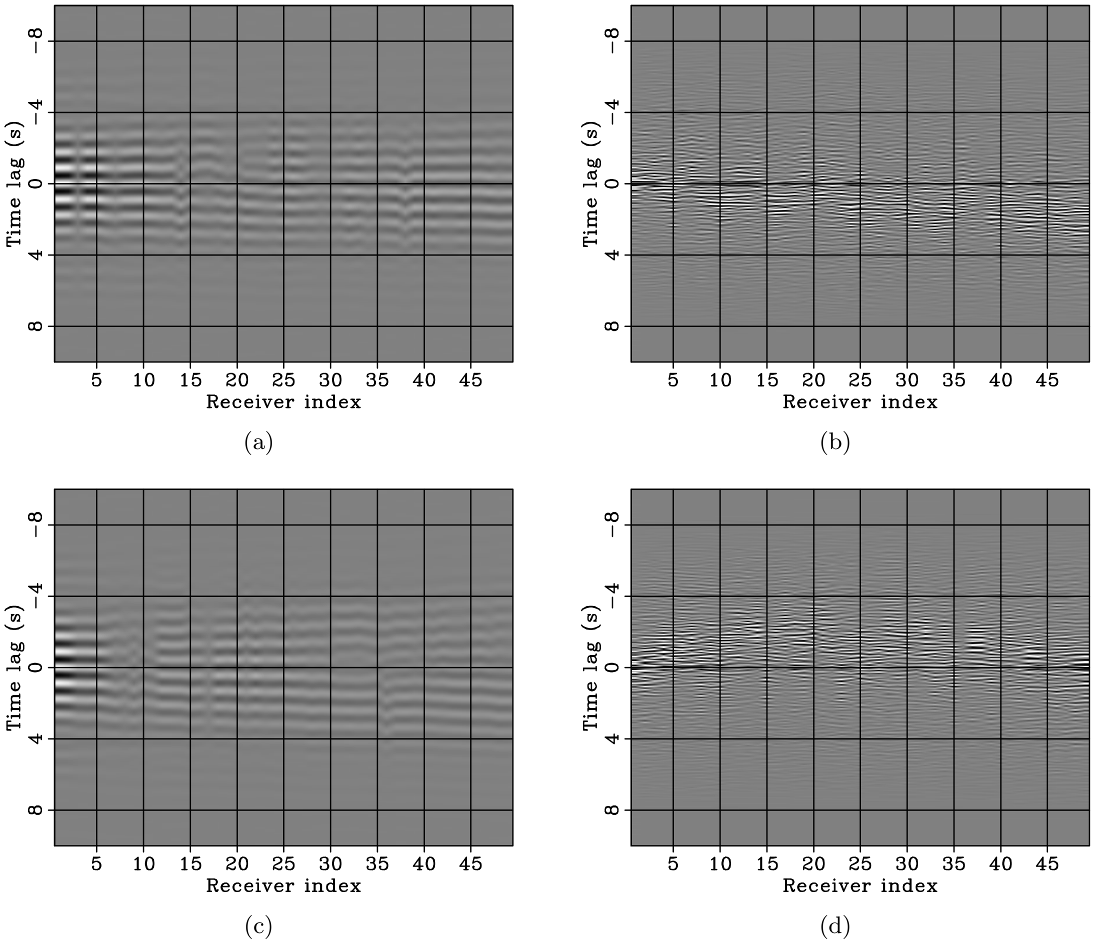
(a)

(b)

(c)

(d)

Figure 3: Virtual source gathers of the hydrophone component. Virtual source locations are indicated by red dots in Figure 1. Western source location (near platform): (a) 1.00–1.25 Hz and (b) 3.00–7.00 Hz. Eastern source location: (c) 1.00–1.25 Hz and (d) 3.00–7.00 Hz. [**CR**] jason1/. g32-Hlo,g32-Hhi,g05-Hlo,g05-Hhi



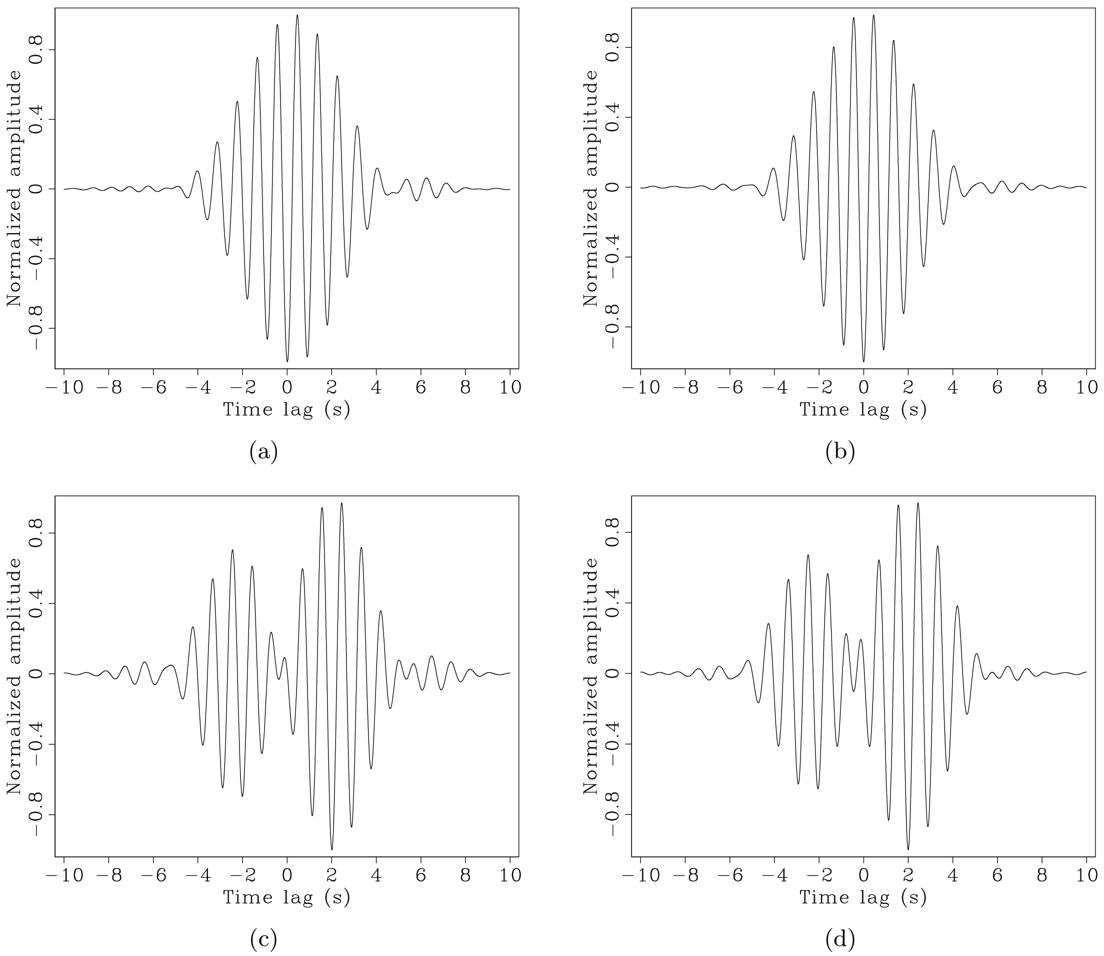
(a)

(b)

(c)

(d)

Figure 4: Virtual source gathers of the vertical geophone component. Virtual source locations are indicated by red dots in Figure 1. Western source location (near platform): (a) 1.00–1.25 Hz and (b) 3.00–7.00 Hz. Eastern source location: (c) 1.00–1.25 Hz and (d) 3.00–7.00 Hz. [**CR**] jason1/. g32-Vlo,g32-Vhi,g05-Vlo,g05-Vhi



(a)

(b)

(c)

(d)

Figure 5: Correlations of hydrophone data at 1.00–1.25 Hz from virtual sources and receivers shown in Figure 1. (a) Between the western virtual source and northern receiver. (b) Between the western virtual source and southern receiver. (c) Between the eastern virtual source and northern receiver. (d) Between the eastern virtual source and southern receiver. The western virtual source is near the center of the array and is thus closer to the receiver locations. [**CR**] jason1/. t32-03,t32-43,t05-03,t05-43

would violate the far-field approximation for reliable travel times, where the distance between the virtual source and receiver should be at least three times the wavelength (Lin et al., 2008). Thus, while the virtual source gathers appear to show Scholte-wave-related arrivals, their travel times are unlikely to be reliable. This includes not only the correlations between nearby nodes (Figures 5(a) and 5(b)), where there is only one apparent arrival with a travel time near 0 s, but also correlations between more distant nodes (Figures 5(c) and 5(d)). Even though these distant nodes display strong events at both negative and positive time lags, their apparent travel times of 3 s are unrealistic given the station spacings of 300 m and the expected velocities between 200 and 400 m/s seen in other regions of the North Sea.

Ideally, there would be coherent energy at higher frequencies, which would correspond to shorter wavelengths. From the spectrograms, it appears that the oil platform is the dominant source of energy at frequencies higher than the microseism band. From the high-frequency virtual source gathers, there is a clear asymmetry in the correlations that supports this conjecture. Events at negative time lags correspond to energy traveling from the receiver to the virtual source, while positive time lags correspond to energy traveling from the virtual source to the receiver. Energy is found primarily at negative time lags when the virtual source is east of the platform (Figures 3(d) and 4(d)). This is expected, as the majority of the energy travels east toward the virtual source location from the platform. Furthermore, energy appears to arrive at near-zero travel time at far offsets, which is an indication of an active source in the region (Chang et al., 2014). Alternatively, energy is found primarily at positive time lags when the virtual source location is near the platform. Again, this is expected, as the energy is propagating away from the virtual source location (the platform) to the other nodes. Therefore, there is substantial evidence that the platform is the dominant source of energy at high frequencies. Unfortunately, the platform does not seem to be energetic enough to stimulate strong interface waves that can be recovered through standard seismic interferometry given the amount of ambient noise data available.

CONCLUSIONS

The Apache Forties OBN survey provided an opportunity to apply passive seismic interferometry to hydrophone and vertical-geophone data from a small, ocean-bottom array. Spectrograms revealed that the microseism energy is nearly equally energetic across the array, while energy at higher frequencies is dependent on distance from the platform, which is the dominant source of noise at these frequencies. I then applied seismic interferometry to times without active seismic shooting. At frequencies between 1.00 and 1.25 Hz, there appears to be interface-wave-related energy. However, because the expected wavelength of these waves are comparable to the maximum aperture of the array of 400 m, it is unlikely the travel times associated with these events are reliable. At frequencies between 3.00 and 7.00 Hz, there are no indications of clear interface-wave arrivals. However, the asymmetry in the correlations reaffirm that the platform is the primary source of noise at high frequencies.

Although the ambient noise data at the Forties is not immediately useful for seismic interferometry and ambient noise tomography, it still contains potential for passive fathometry. Rough weather conditions and more violent breaking waves at the sea surface are ideal for the technique, which attempts to create shallow subsurface reflection images by correlating passive up- and down-going energy. Understanding of the noise sources in and around the array is useful for creating filters to dampen horizontally-traveling waves, including interface waves.

ACKNOWLEDGMENTS

The author would like to thank Apache North Sea Limited for access to the data set and permission to publish. The author also thanks Shuki Ronen and Stewart Levin for help processing the data and useful discussions.

REFERENCES

Bensen, G., M. Ritzwoller, M. Barmin, A. Levshin, F. Lin, M. Moschetti, N. Shapiro, and Y. Yang, 2007, Processing seismic ambient noise data to obtain reliable broad-band surface wave dispersion measurements: Geophysical Journal International, **169**, 1239–1260.

Chang, J. P., N. Nakata, R. G. Clapp, B. Biondi, and S. de Ridder, 2014, High-frequency surface and body waves from ambient noise cross-correlations at Long Beach, CA: 84th Annual International Meeting, SEG, Expanded Abstracts, 2235–2239.

de Ridder, S. and J. Dellinger, 2011, Ambient seismic noise eikonal tomography for near-surface imaging at Valhall: The Leading Edge, **30**, 506–512.

Lin, F.-C., M. P. Moschetti, and M. H. Ritzwoller, 2008, Surface wave tomography of the western United States from ambient seismic noise: Rayleigh and Love wave phase velocity maps: Geophysical Journal International, **173**, 281–298.

Lin, F.-C., M. H. Ritzwoller, J. Townend, S. Bannister, and M. K. Savage, 2007, Ambient noise Rayleigh wave tomography of New Zealand: Geophysical Journal International, **170**, 649–666.

Lobkis, O. I. and R. L. Weaver, 2001, On the emergence of the Green's function in the correlations of a diffuse field: The Journal of the Acoustical Society of America, **110**, 3011–3017.

Mordret, A., M. Landès, N. Shapiro, S. Singh, P. Roux, and O. Barkved, 2013, Near-surface study at the Valhall oil field from ambient noise surface wave tomography: Geophysical Journal International, ggt061.

Sabra, K. G., P. Gerstoft, P. Roux, W. Kuperman, and M. C. Fehler, 2005, Extracting time-domain Green's function estimates from ambient seismic noise: Geophysical Research Letters, **32**, L03310.

Seats, K. J., J. F. Lawrence, and G. A. Prieto, 2012, Improved ambient noise correlation functions using Welch's method: Geophysical Journal International, **188**, 513–523.

Shapiro, N. M., M. Campillo, L. Stehly, and M. H. Ritzwoller, 2005, High-resolution surface-wave tomography from ambient seismic noise: Science, **307**, 1615–1618.

Snieder, R., 2004, Extracting the Green's function from the correlation of coda waves: A derivation based on stationary phase: Physical Review E, **69**, 046610.

Wapenaar, K., 2004, Retrieving the elastodynamic Green's function of an arbitrary inhomogeneous medium by cross correlation: Physical Review Letters, **93**, 254301.

Wapenaar, K., D. Draganov, R. Snieder, X. Campman, and A. Verdel, 2010, Tutorial on seismic interferometry: Part 1—Basic principles and applications: Geophysics, **75**, no. 5, 75A195–75A209.

Overview of the Apache Forties data set

Gustavo Alves

ABSTRACT

I present a dataset acquired in 2010 by Apache North Sea and made available to the Stanford Exploration Project (SEP). The dataset is composed of three surveys acquired with four-component ocean-bottom nodes placed under production platforms in a close pattern. Active shooting was performed in a spiral geometry around the platforms at short offsets. The goal of this acquisition was to identify gas pockets at shallow depths and possible geo-hazards. This acquistion configuration can be advantageous for multi-component data processing such as elastic Reverse Time Migration (RTM).

INTRODUCTION

Seismic acquistion is an essential tool in both the exploration and exploitation phases in the oil & gas industry. During production, it can be used for 4D monitoring, by observing seismic response changes in the reservoir and overburden, in order to improve the production strategy. Seismic data can also be used to identify geo hazards at shallow depths, which create a risk for drilling and other operations during production (Heggland, 2004).

ACQUISITION GEOMETRY

The data provided to SEP by Apache North Sea were recorded using four-component ocean bottom nodes (4C OBNs). The nodes were arranged at each of the production platforms in the survey area in a hexagonal array. Sources were shot in a coil pattern around the platform. The total survey time for each field resulted in a few days of passive data. From the passive data, active data was extracted in 9-second traces.

DATASETS

Bravo data

Data for the Bravo survey was acquired using 10,108 shots. There were 52 nodes laid out in a hexagonal pattern. Figure 1 shows the source and receiver configuration.

Charlie data

Data for the Charlie survey was acquired using 10,494 shots. There were 53 nodes laid out in a hexagonal pattern. Figure 2 shows the source and receiver configuration.

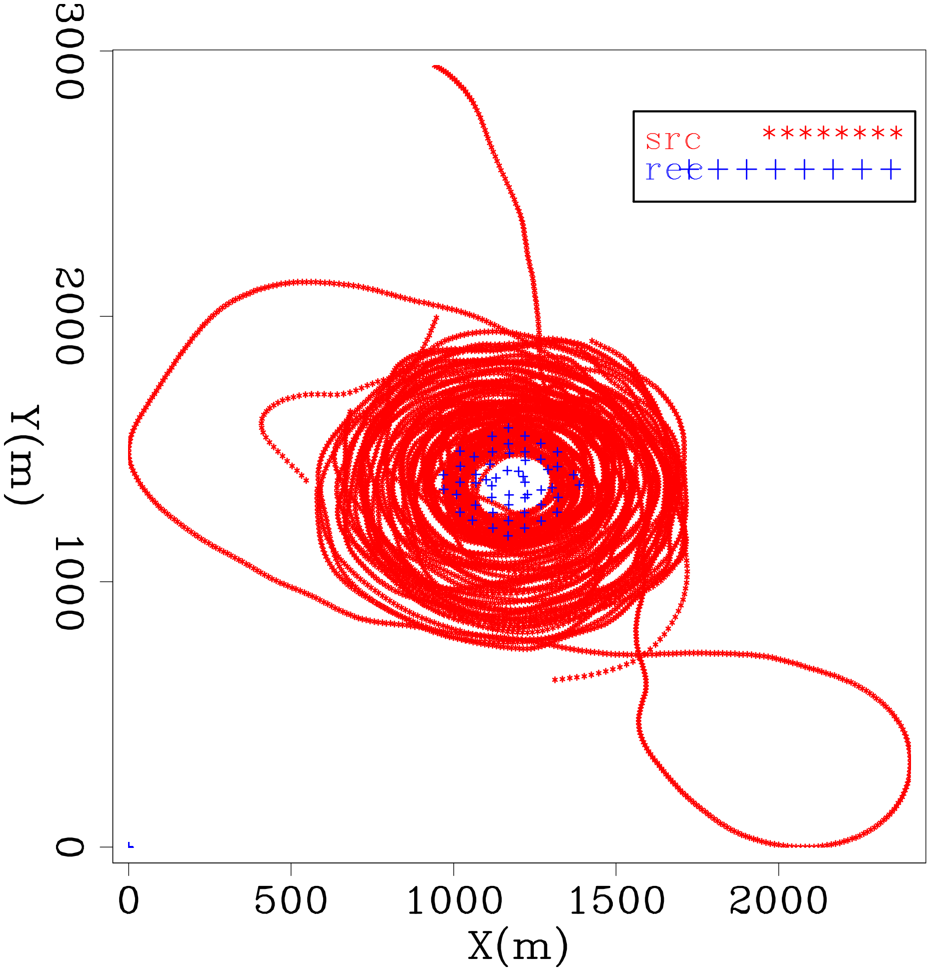


Figure 1: Shot and receiver configuration for the Bravo survey. [**NR**] gcalves1/. recbravo

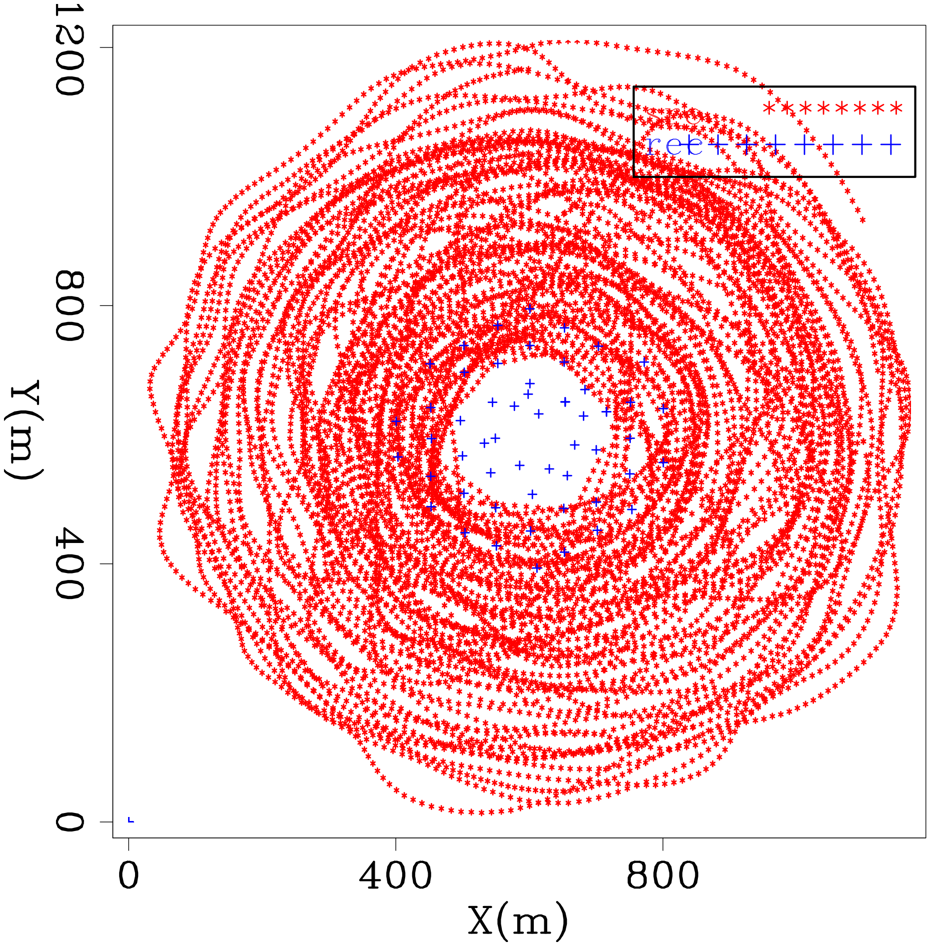


Figure 2: Shot and receiver configuration for the Charlie survey. [**NR**] gcalves1/. reccharlie

Delta data

Data for the Delta survey was acquired using 14,485 shots. There were 48 nodes laid out in a hexagonal pattern. Figure 3 shows the source and receiver configuration.

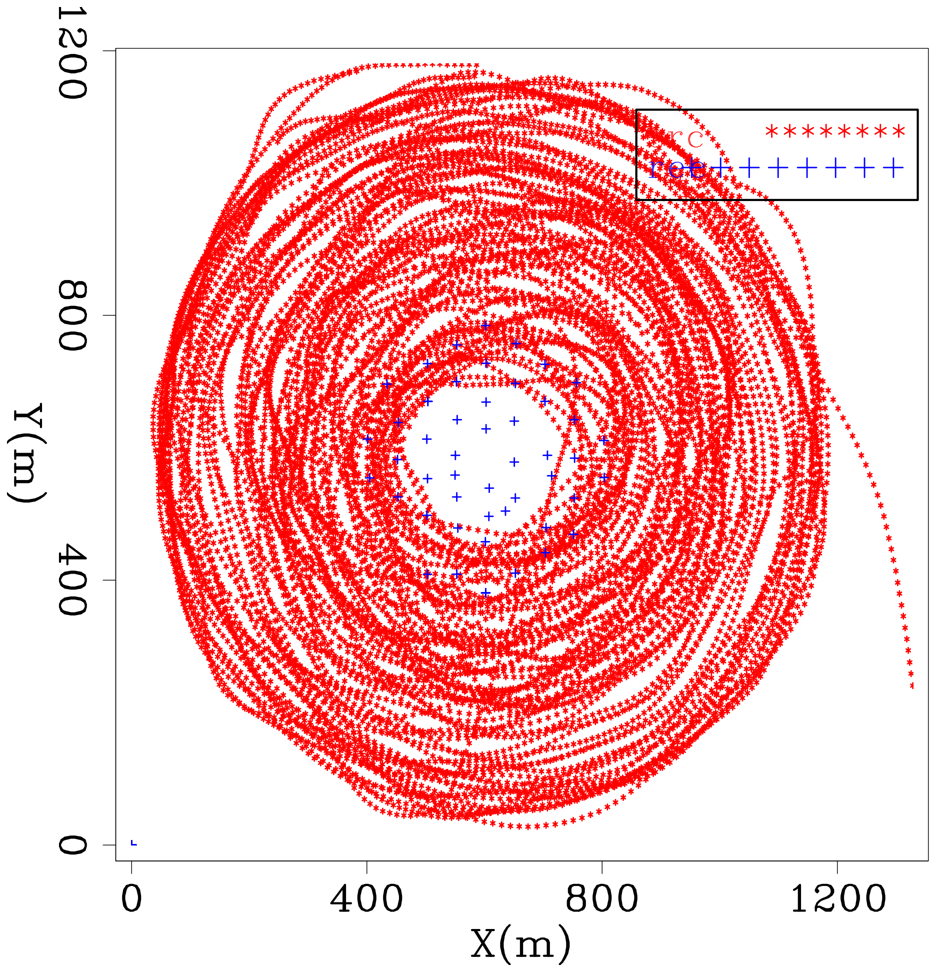


Figure 3: Shot and receiver configuration for the Delta survey. [**NR**] gcalves1/. recdelta

ACKNOWLEDGMENT

The author would like to thank Apache North Sea for releasing this field data and for their support to the Stanford Exploration Project (SEP). The author would also like to thank Petrobras for their financial support.

REFERENCES

Heggland, R., 2004, Definition of geohazards in exploration 3-D seismic data using attributes and neural-network analysis: AAPG Bulletin, **88**, 857–868.

Industrial scale high-performance adaptive filtering with PEF applications

Kaixi Ruan, Joseph Jennings, Ettore Biondi, Robert G. Clapp, Stewart A. Levin, and Jon Claerbout

ABSTRACT

In most areas of SEP research, neither data nor models are truly stationary. As a consequence, global fitting, regridding, and inversion rarely approach the ideal of independent, identically-distributed residuals. Here we design and develop parallel multidimensional adaptive filter routines parsimonious in both memory and computation, suitable for industrial scale applications. We applied them to examples of three classes of time and space variable adaptive Prediction Error Filters (PEF) applications: (1) blind decon, (2) filling gaps in data, and (3) making random realizations of data; and then, outline other areas where these tools will have a wide application.

INTRODUCTION

Estimating data covariance, or in the case of PEFs the inverse covariance, is a powerful tool with a wide range of applications. Its uses include decon, missing data interpolation, signal and noise separation, and fault detection. Because both seismic data and the subsurface properties we want to infer from them are nonstationary, we need time- and space-variable PEFs.

Time-variable filtering has a long history with space-variable filtering only a few steps behind. In the context of very large, e.g., terabyte, datasets, big challenges must be overcome to be practical, as follows:

- Constraining memory usage
- Minimizing arithmetic complexity
- Simplifying the parameter space
- Enabling an efficient parallel implementation
- Avoiding instability
- Constructing a tool that can easily be shared in a wider context with many other people.

Over the last twenty-five years several attempts of capturing non-stationary covariance have been attempted at SEP. Schwab (1998) solved a series of stationary problems using overlapping patches. The downside of using patching is the large parameter space (size of patch, amount of overlap) that must be searched to find an acceptable result. Crawley

(2000) solved for a global non-stationary covariance by creating micro-patches and requiring them to vary smoothly. The micro-patch approach adds significant memory requirements (for each data point you must store which micro-patch with which it is associated) and the resulting indirection reduces performance. A third approach is to linear interpolate from a sparse grid of filter locations. The downside of this approach is that it requires interpolating the filter at every location. The interpolation cost can be minimized by being careful on how you interpolate between the sparse grid of filters Hale (2006). Here we use this sparse grid of filters approach.

In this progress report we describe our method and show a small number of preliminary results on 2-D seismic data, 2-D images, and on 3-D synthetic data. We have not yet done enough tests to recognize all its opportunities and limitations. We have not yet documented a clean user interface. But we think the code is working and look forward to providing more examples and documentation for general use.

First we summarize the basic principles of multi-dimensional prediction-error filtering (PEF). Then we describe our Fortran 2003 implementation of non-stationary filter estimation. Finally, we show examples that support our claim to have built an important basic tool. However, we have not yet tested use of this tool in general inversion problems other than deconvolution and interpolation.

PREDICTION ERROR FILTERS

Before diving into the enticing details, recall the textbook definition of a multidimensional prediction error filter (PEF) which underlies our time-variable IID (TV-IID) applications.

In one dimension, it vanishes before $t = 0$; it takes a unit value at $t = 0$; and it has coefficients determined by least squares for $t > 0$. In 2-D, the unit value is along one side of a rectangle. Analogous to the 1-D PEF vanishing before $t = 0$, in 2-D there are zero values along one edge of the rectangle before (or after) the unit value. In 3-D the unit value is along one face of a box, as depicted in Figure 1. The enticing mathematical feature of PEFs is that their filter outputs are white. More precisely, the output tends to whiteness as the filter size increases. This fact is shown in GIEE Claerbout (2014) chapter 7 (based on chapter 4).[1]

In multidimensional PEFs, once their coefficients have been learned (by least squares), contain the inverse spectrum of the input data. The PEFs are valuable because they have many uses that include:

1. Conditioning residuals for inverse problems
2. Blind deconvolution
3. Creating random realizations
4. Filling missing data

[1] While that source only proves whiteness in a 1-D sense, its helix interpretation of the convolution of multidimensional data and filter as an equivalent 1-D convolution of unwrapped data and unwrapped filter imply the N-D result.

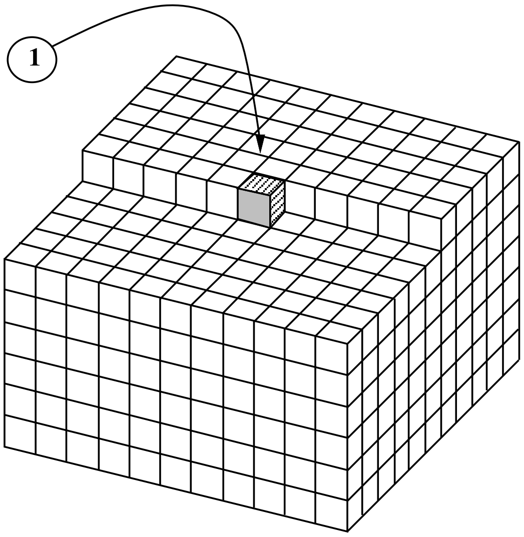


Figure 1: A 3-D causal filter at the starting end of a 3-D helix. [**NR**] kaixi1/. 3dpef

Figure 2 shows the boundary effects that arise when the PEF is not allowed to access data outside a rectangular mesh. (It is left-right reversed compared with Figures 4-6.) Zeros around three edges of our Figures show the filter size.

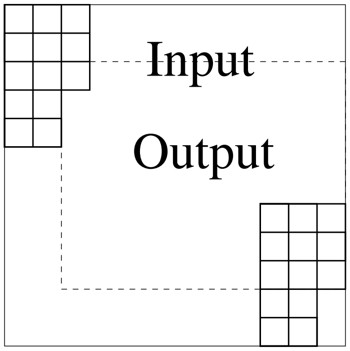


Figure 2: Domain of inputs and outputs of a two-dimensional filter like a PEF. [**NR**] kaixi1/. rabdomain

DESIGN CONSIDERATIONS: THREE BOXES

Confusion may arise because distance scales arise in three contexts, (1) data fabric, (2) filter, and (3) blending. In 2-D these distance scales define boxes. In 3-D cubes.

1. Fabric box. Data being investigated will have characteristic lengths relating to how fast its spectrum changes. This box subjectively defines the size of a region where the spectrum is roughly constant. For example, you might presume on one side of a CMP gather that hyperbolic moveout roughly allows interpolation along 3 to 4 straight lines.

2. Filter box. The application programmer or end user must choose the filter size. It is given in points on the axis, e.g., (20, 2) might mean a filter 10 points long on the time axis acting on 2 neighboring traces.

3. Blending box. The spacing on the coarse mesh that filters are defined. They are linearly interpolated to the data mesh. For example, a blending box size of (1/3, 1/3) means we count 3 boxes along each axis meaning 4 points along each axis at which filters are defined.

Having a smaller blending box than the fabric box allows the filter field to adapt to the local degree of nonstationarity. On the other hand, too small a blending box introduces the danger of overfitting. Examples in our immediate future should provide us a better understanding of appropriate ratios among these three box sizes. Unfortunately, the fabric box size is intrinsically space variable so ultimately we appear to be facing deeper issues sometimes glibly called "deep learning." Hopefully, by limiting our examples to realistic seismic data we can provide users with useful guidance for parameter setting.

METHOD AND IMPLEMENTATION

Our TV-IID operator applies a linear interpolation to filter coefficients defined on a coarse grid to calculate filter coefficients for each point in the denser data mesh. These interpolated filter coefficients are then applied at that data mesh location. Although it is theoretically equivalent to the chain of two well-known operators that we already have in the current library, we did not adopt this approach because it is memory intensive. We concoct a new operator that blends the two in a computationally efficient way to make it practical.

Filter design

We designed our linearly varying filter in an object-oriented manner using Fortran 2003. The main filter class contains a 2-D `lag` array that contains the coordinates describing the shape of the filter and also two arrays that contain locations of the filter coefficients on the coarse grid. The class requires that the user provide the filter spacing they desire on each axis from which it computes and builds bounds for the blending boxes `b`{egin} and `e`{nd} arrays.

This filter type also contains an array containing the estimated filter coefficients. We have designed this array for the first dimension to contain the filter coefficients, and the remaining dimensions to be the number of filters along each axis present in the data. With this design, the number of dimensions of this array will grow with the size of the problem. To account for this growth, we extend our filter class, to 1-D, 2-D, 3-D, . . . types.

Code parallelization

Because the filter coefficients on a data point only depend on the patch inside which it resides (they are interpolated from those defined on the corners of the patch), each patch can perform the TV-IID operation independently. In our code the outermost loop is over patches and we can easily use tools such as OpenMP for parallelization.

Fast linear interpolation in one dimension

Given an n points 1-D grid and any filter coefficient `f_1` at the beginning and `f_n` at the end of the grid, the traditional linear interpolation looks like the following:

```
real :: g,h
integer :: i,n
do i=2,n-1
   g=(i-1)/(n-1)
   h=1-g
   f_i= h*f_1+g*f_n
end do
```

Fast interpolation removes a division, subtraction and two multiplications from the inner loop, as follows:

```
real :: delta
integer :: i,n
delta=(f_n-f_1)/(n-1)
f_i = f_1
do i=2,n-1
   f_i= f_i+delta
end do
```

We can trace the history of this algebraic reorganization back to Newton's method of divided differences (Newton (1687); Stirling (1730)). It is the simplest example of interpolation by numerical solution of differential equations.

Design discussion

We deliberately did not use an explicit memory array for the values intermediate between `f_1` and `f_n` because they can, and usually would, be computed and applied on the fly.

Our "fast interpolation" is *recursive* in the sense that filter coefficient updates are tied to local shifts of the filter by 1 grid point. It is not a significant impediment for this method because: (a) Our main parallelism is over an outer loop of patches, (b) If we start from the center of a patch, we can simultaneously work both left and right, up and down, etc., i.e., $2n$ directions in n dimensions[2], and (c) Even these inner loops could be unrolled to compute interleaved entries with a multiple of `delta` should the need arise.

While the difference in arithmetic cost is an attractive factor of approximately three per dimension, an equally important benefit in higher dimensions is that we avoid repeatedly accessing coarse grid filter coefficient arrays (`f_1` and `f_n` in the above example) that are widely separated in memory.

Code samples

Using the above example, the 1-D forward operator is shown here. The reader may easily identify the part of fast linear interpolation and the part of convolution.

[2] Snaking back and forth columns and rows helps preserve locality and reduce cache misses. More sophisticated "space-filling paths" (see, e.g., Savage (1997); Knuth (2005); Flahive (2008)) that tend to optimize parallel locality are also possible.

```
subroutine forward1D(m,f,d) !output d=f*m
      real, dimension(:) :: m
      type(lv_filter1d)  :: f
      real, dimension(:) :: d
      integer            :: i,j,id
      real,allocatable   :: b0(:),d0(:)

      allocate (b0(size(f%coef1,1)),d0(size(f%coef1,1)))

      do i = 1,size(f%b,2)         ! Loop over patches
        b0=f%coef1(:,i)
        d0=(f%coef1(:,i+1)-b0)/(f%e(1,i)-f%b(1,i)+1)
        do id=f%b(1,i),f%e(1,i)    ! Loop over each point in the patch
          do j = 1,size(f%coef1,1) ! Loop over the filter coefficients
            d(id+f%lag(1,j))=d(id+f%lag(1,j))+b0(j)*m(id)
          end do
          b0=b0+d0
        end do
      end do
      deallocate(b0,d0)
end subroutine forward1D
```

The above example implements the concept of convolving a filter and vector to produce another vector. We can write this in matrix form in two different ways. The first approach is

$$\mathbf{d} = \mathbf{M}\mathbf{f}, \tag{1}$$

where **M** is convolving with our model, **f** is our filter, and **d** is our data. The second approach is

$$\mathbf{d} = \mathbf{F}\mathbf{m}, \tag{2}$$

where **F** is convolving with our filter, and **m** is our model. The two approaches imply two different adjoints. The first approach, equation 1 is needed to estimate the filter. The second approach, equation 2 is needed to apply the inverse covariance to a model. We have implemented both types of adjoints for up to three dimensional models.

EXAMPLES

We tested our library on several different datasets with different objectives. In our first example we looked to see how our non-stationary filter performed when confronted with an image with sharp changes in covariance. In our second example we attempted to capture only smooth changes in our covariance, and use what was not captured to find anomalies. Our final example shows a 3-D adaptive prediction error output for a synthetic with fine layering, faulting, and a wide range of dips.

Basket fabric data

Our first example was to check make sure our filter estimation was producing reasonable results. We chose the basket fabric from Claerbout (2014) because it had significant spatial variation that confused stationary filtering. We did not expect to perfectly capture the covariance because the abrupt change in dips in the fabric breaks our fundamental assumption of smooth change implied by our filter interpolation. We first estimated the covariance by solving in a least-squares sense the objective

$$\mathbf{0} \approx \mathbf{D}\mathbf{f}, \tag{3}$$

where $\mathbf{0}$ is a vector of zeros, $\mathbf{D}$ is convolution with our data, $\mathbf{f}$ is our inverse covariance PEF estimate. Viewing $\mathbf{f}$ is uninformative. To get a sense of how effective our representation of the covariance was, we attempted to see the inverse of $\mathbf{f}$ by estimating the least squares objective function represented by

$$\mathbf{n} \approx \mathbf{F}\mathbf{m}, \tag{4}$$

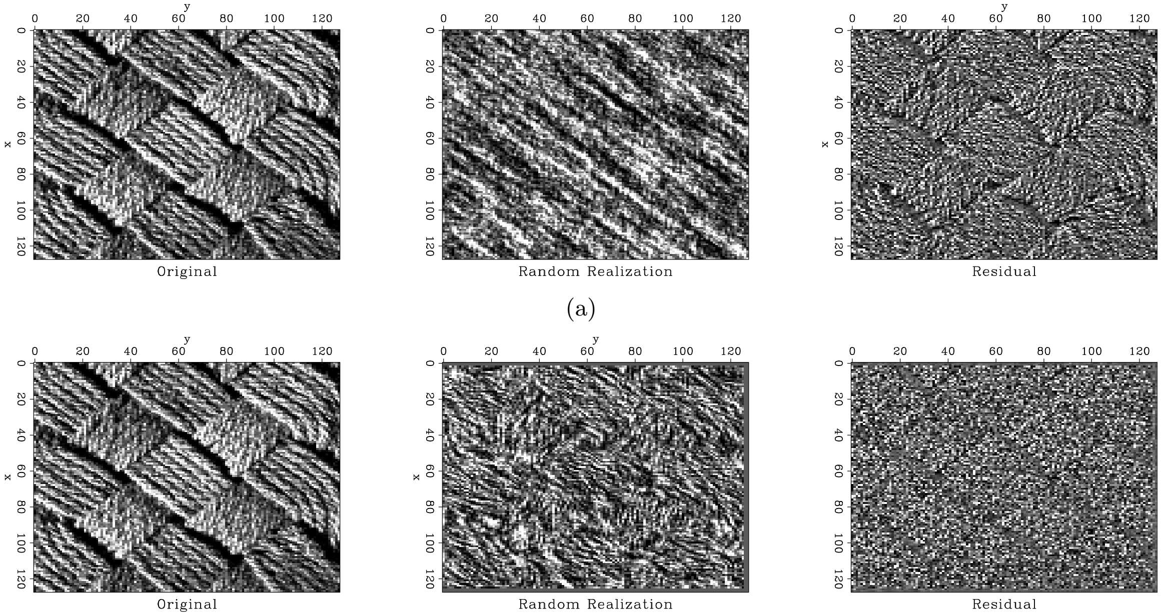
where $\mathbf{n}$ contains random noise. Any random noise vector in equation 4 gives a random realization of data. $\mathbf{F}$ is convolving with our PEF representation of the inverse covariance, and $\mathbf{m}$ is our model that should now have the same covariance (the inverse of the inverse covariance) as our original data. In addition, we inspected the result of filtering our data $\mathbf{d}$ with our PEF representation,

$$\mathbf{r} = \mathbf{F}\mathbf{d}, \tag{5}$$

where $\mathbf{r}$ should be IID if we perfectly captured the model's spectrum. We estimated the fabric's spectrum using both a stationary (Figure 3(a)) and non-stationary (Figure 3(b)) filter. In each figure the left panel shows the original fabric, the center panel shows $\mathbf{m}$ from equation 4, and the right panel shows the residual from equation 5. Note how for the stationary case we see the same dips everywhere in the center panel of Figure 3(a), while we see changing dip pattern in the non-stationary case (Figure 3(b)). For the stationary case in the right panel of Figure 3(a), we can clearly delineate the boundaries where the dips changed in the original fabric, while in the non-stationary case these changes are not as obvious. In both cases the residual is approximately the same amplitude. As expected, the non-stationary filter did not do a significantly better job in capturing the fabric's covariance. Looking at the center panel of Figure 3(b) we see it attempted to explain the abrupt changes through smoothly changing the dips. Exactly what we hoped to see.

Feature identification

One of the advantages of our approach to estimating non-stationary covariance is that we have fewer adjustable parameters as compared to other approaches. Our approach requires us to define the shape of the PEF and the distance between our control points where the different PEFs sit. Claerbout (2014) gives a fuller description of PEF design, but for this paper two concepts are important. In the stationary case the length of the filter along the time axis controls the range of dips we can see and how accurately we can capture a wavelet (more points implies finer sampling as a function of frequency). In the stationary case the number of rows defines how many dips we wish to capture. In the non-stationary case these rules generally still hold. With our non-stationary approach we also set the distance between our control points, which generally determines the rate of dip spectrum change we want to capture. The "generally" used above is because these parameters do interact. Adding rows to a PEF will allow it to find additional dips (that might be a function of non-stationarity). As the control points get closer together different PEFs can start to detect different dips that are actually stationary in nature.



(a)

(b)

Figure 3: Basket weave: (a) Stationary, (b) Non stationary. A 45° trend dominates. **Realization:** On the stationary we see the 45° trend everywhere. **Residual:** Trends are less pronounced. [**ER**] kaixi1/. basket-s,fabric-ns

For our second test we wanted to see how well a non-stationary PEF could capture how the spectrum of an image changes in frequency content as a function of time and dip content as function of (mainly) space. We made the choice to *not* try to capture fast changes. To accomplish these goals, we designed our PEF so it was limited in how many dips it could capture, and we put a significant distance between our control points. Figure 4(b) shows the result. Again the left panel show the original data, the center panel the simulated data using equation 4, and the right panel the residual (equation 5). For comparison we did the same test using a stationary filter (Figure 4(a)). If you look at the residual of the stationary result you can still clearly see bedding structures throughout the residual. In the non-stationary residual almost all of the horizontal bedding structure is absent. The simulated data shows why. The non-stationary simulated data have the same changing dip pattern seen in the original image while stationary result only shows the dominant dip direction. As mentioned, we carefully chose our parameters to limit the number of dips we could capture. Note that in the residual of the non-stationary case the diffractions are the dominant feature.

Qdome 3-D

The Qdome 3-D quarter dome synthetic Claerbout (1993) shown in Figure 5(a) has a wide range of dips and azimuths, including aliased reflectors, and so provides an appropriate challenge for our nonstationary prediction error filtering. The prediction error of Figure 5(b) is a mixed result. On the positive side, it removed a large percentage of the smooth bedding amplitudes and clearly outlined the location of the fault surface. On the negative

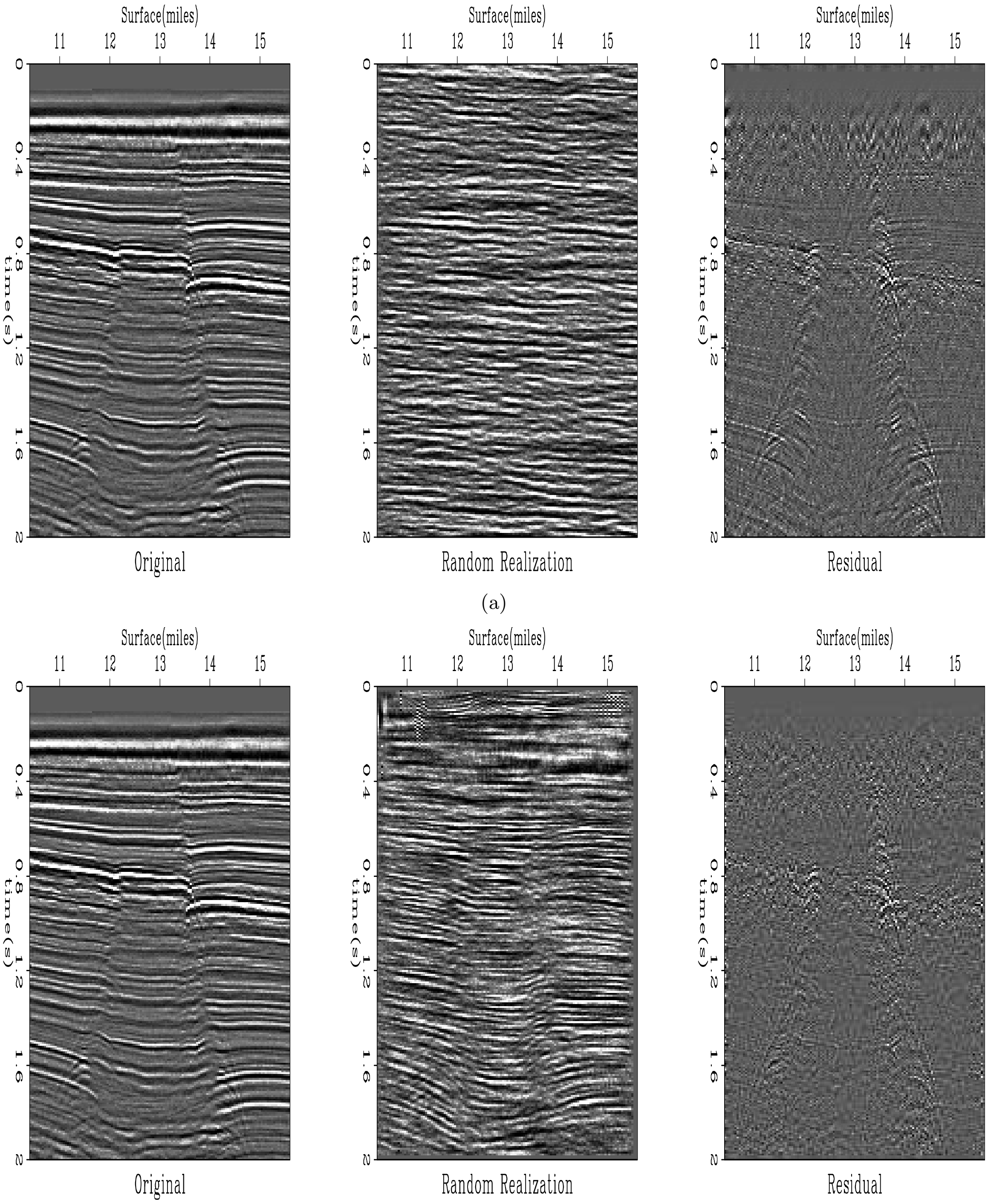


Figure 4: Gulf of Mexico seismic stacked section: (top) stationary, (bottom) nonstationary. **Random realization:** The nonstationary preserves the fault location because it is embedded in the local PEF. While stationary scatters low frequencies throughout, nonstationary keeps them at early time. **Residual:** The stationary does not drive down the horizontal layers very well because their slopes vary with location. [**ER**] kaixi1/. wg-s,wg-ns

side, we have left significantly more bedding texture than in our earlier seismic Gulf of Mexico data example and we have, unexpectedly, not handled the steep aliased dips along the left hand side. Deadline pressure did not allow exploring various filter shapes and sizes.

CONCLUSIONS AND DISCUSSION

We presented an addition to the SEP inversion library to capture non-stationary covariance. We tested the library on a series of problems including feature identification and missing data. In the future we will extend this work into even higher dimensional spaces and use it for non-stationary decon and signal and noise separtion.

REFERENCES

Claerbout, J., 2014, Geophysical Image Estimation by Example: lulu.com.

Claerbout, J. F., 1993, 3-D local monoplane annihilator: SEP-Report, **77**, 19–25.

Crawley, S., 2000, Seismic trace interpolation with nonstationary prediction-error filters: PhD thesis, Stanford University.

Flahive, M., 2008, Balancing R-ary Gray codes II: The Electronic Journal of Combinatorics, **15**, no. R128.

Hale, D., 2006, Recursive Gaussian filters: Center for Wave Phenomena Report, **546**.

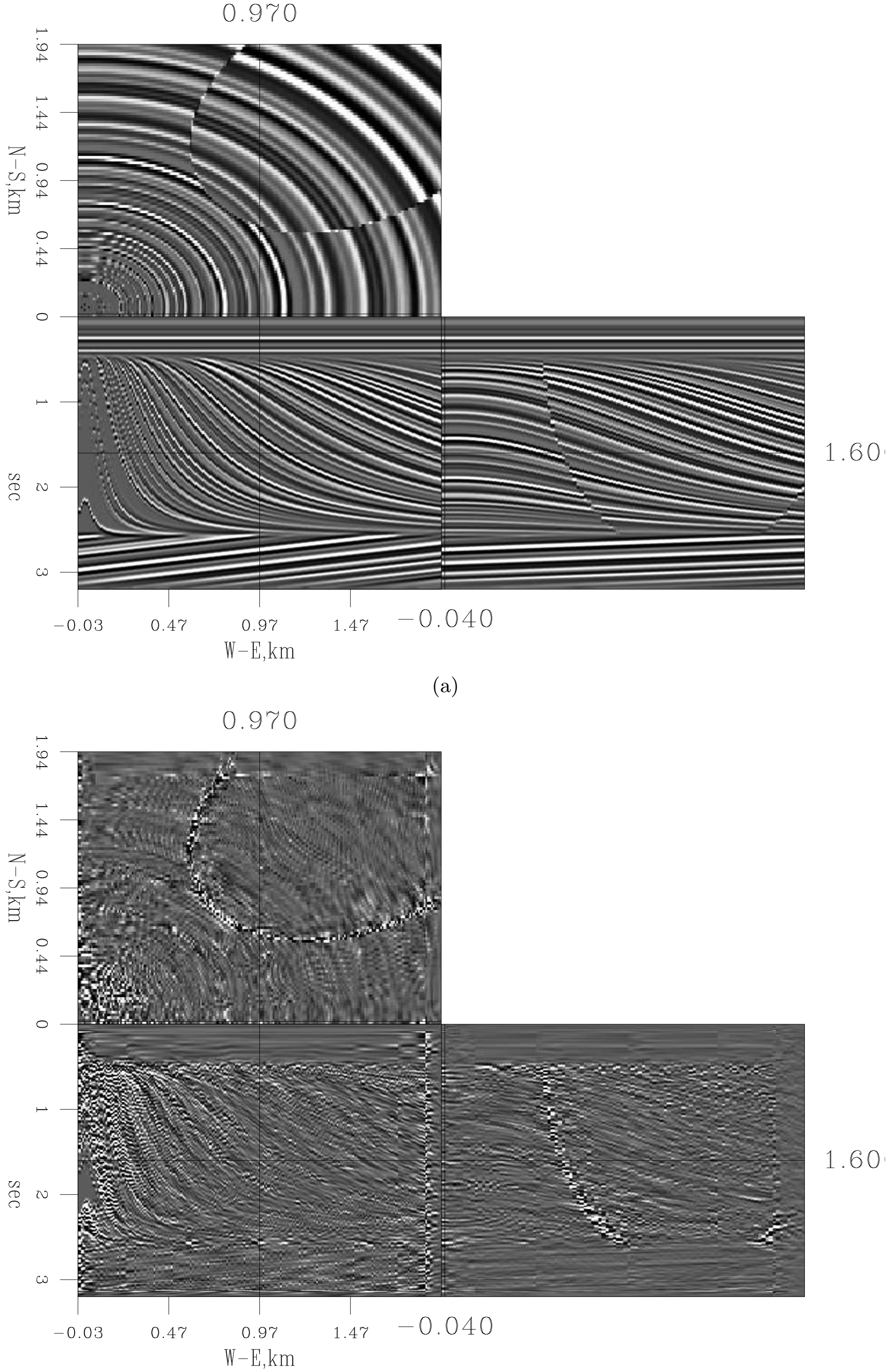
Knuth, D. E., 2005, The art of computer programming: Volume 4 Generating all tuples and permutations.

Newton, I., 1687, Philosophiae Naturalis Principia Mathematica: S. Pepys. (Divided differences are outlined in Book III, Lemma V).

Savage, C., 1997, A survey of combinatorial Gray codes: SIAM Review, **39**, no. 4, 605–629.

Schwab, M., 1998, Enhancement of discontinuities in seismic 3-D images using a Java estimation library: **99**.

Stirling, J., 1730, Methodus differentialis: sive tractatus de summatione et interpolatione serierum infinitarum. Auctore Jacobo Stirling, R.S.S.: Gul. Bowyer, London. (Newton's divided difference method described in *Pars Secunda de Interpolatio Serierum, p. 85ff*).



(a)

(b)

Figure 5: (a) Original Qdome data. (b) Prediction-error residual. The blank borders tell us the size of our $20 \times 10 \times 10$ filter stencil. Here the horizontal display exaggeration is 2:1 in x and 4:1 in y relative to the z axis. [**CR**] kaixi1/. qdome,qdome-rsd

Stable reorientation for the Forties dataset

Stewart A. Levin and Jason P. Chang

ABSTRACT

Traces from the nodal receivers used in the Forties platform undershoot require rotation to provide consistent multicomponent orientation. A quick quality control check on input data that were within just a few degrees of the desired North-East-Vertical orientation found vector magnitude changes approaching 25%. Inspecting the code in the existing module that applied nodal reorientation, the presence of an **IF** test against an arbitrary choice of epsilon, explicit unit vector renormalization, and angle transformations that their author questioned spurred us to derive and apply a stable and robust alternative that avoided those issues.

INTRODUCTION

The seafloor nodes used in the Forties platform undershoot deliver four output traces: one hydrophone and three geophone records. In addition to an X heading, three tilt angles, one each for X, Y and Z, are provided in the SEG Y trace headers. These angles provide sufficient information for reorienting the geophone outputs to Vertical, North and East. The challenge is to ensure that the transformation(s) derived from those angles preserves amplitude fidelity and component orthogonality. We emphasize that, while rotation matrices appear throughout the development, we are not "rotating" data but understanding out how to reexpress it in various useful choices of coordinate systems.

THEORY

The vertical components of the X, Y and Z unit direction vectors are given by the sines of their respective tilt angles $\theta_x, \theta_y, \theta_z$. With these nodes, positive is upward and X-Y-Z, like the original Galperin G_1-G_2-G_3 directions[1], are a right handed coordinate system. We need to determine how the tilted geophone axes relate to the global E-N-V coordinate system.

To help understand and translate it into a conventional matrix notation, we dredge up our basic linear algebra. Let $(\mathbf{p}_1,\ \mathbf{p}_2,\ \mathbf{p}_3)$ be the orthonormal vector basis in which a vector $\mathbf{v}$ of interest $(a,\ b,\ c)$ is expressed, i.e. $\mathbf{v} = a\mathbf{p}_1 + b\mathbf{p}_2 + c\mathbf{p}_3$, and let $(\mathbf{q}_1,\ \mathbf{q}_2,\ \mathbf{q}_3)$ be the orthonormal vector basis in which we want to reexpress $\mathbf{v}$. The coefficients $(\hat{a},\ \hat{b},\ \hat{c})$ are obtained via the dot product of the new basis with $\mathbf{v}$:

$$\begin{aligned}
\hat{a} &= a\,\mathbf{q}_1\cdot\mathbf{p}_1 + b\,\mathbf{q}_1\cdot\mathbf{p}_2 + c\,\mathbf{q}_1\cdot\mathbf{p}_3 \\
\hat{b} &= a\,\mathbf{q}_2\cdot\mathbf{p}_1 + b\,\mathbf{q}_2\cdot\mathbf{p}_2 + c\,\mathbf{q}_2\cdot\mathbf{p}_3 \\
\hat{c} &= a\,\mathbf{q}_3\cdot\mathbf{p}_1 + b\,\mathbf{q}_3\cdot\mathbf{p}_2 + c\,\mathbf{q}_3\cdot\mathbf{p}_3
\end{aligned}$$

[1] See, for example, Grazier (2009) for a detailed discussion of the Galperin sensor configuration.

which, in matrix form, says

$$\begin{pmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_1 \cdot \mathbf{p}_1 & \mathbf{q}_1 \cdot \mathbf{p}_2 & \mathbf{q}_1 \cdot \mathbf{p}_3 \\ \mathbf{q}_2 \cdot \mathbf{p}_1 & \mathbf{q}_2 \cdot \mathbf{p}_2 & \mathbf{q}_2 \cdot \mathbf{p}_3 \\ \mathbf{q}_3 \cdot \mathbf{p}_1 & \mathbf{q}_3 \cdot \mathbf{p}_2 & \mathbf{q}_3 \cdot \mathbf{p}_3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} . \tag{1}$$

To go the other way, the matrix inverse is its transpose, and therefore,

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \mathbf{q}_1 \cdot \mathbf{p}_1 & \mathbf{q}_2 \cdot \mathbf{p}_1 & \mathbf{q}_3 \cdot \mathbf{p}_1 \\ \mathbf{q}_1 \cdot \mathbf{p}_2 & \mathbf{q}_2 \cdot \mathbf{p}_2 & \mathbf{q}_3 \cdot \mathbf{p}_2 \\ \mathbf{q}_1 \cdot \mathbf{p}_3 & \mathbf{q}_2 \cdot \mathbf{p}_3 & \mathbf{q}_3 \cdot \mathbf{p}_3 \end{pmatrix} \begin{pmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{pmatrix} . \tag{2}$$

Given a rotation $\mathbf{R}$ that transforms the $\mathbf{p}_i$ to $\mathbf{q}_i$, we initially express the vectors $\mathbf{p}_i$ and $\mathbf{q}_i$ in terms of the $\mathbf{p}_i$ basis itself, and place them as matrix columns in $\mathbf{P} = \mathbf{I}$ and $\mathbf{Q}$ respectively. With this representation, we have simply $\mathbf{Q} = \mathbf{R}$, and the transformation matrix in equation (1) becomes $\mathbf{R}^{\mathrm{T}}$ and, correspondingly, the transformation matrix in equation (2) is simply $\mathbf{R}$, which tells us that when we construct a rotation, we should apply its inverse to get the proper coordinate transformation.

The key step for nodal reorientation is to determine the orientation of a horizontal axis of rotation that is consistent with both the X and Y tilts. Working backward from the local E-N-V right handed orientation, let $(\cos\alpha, \sin\alpha, 0)$ be the to-be-determined unit vector perpendicular to this initial axis of rotation. Changing coordinates for this vector to become $(1, 0, 0)$ is accomplished by rotation

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{pmatrix} . \tag{3}$$

Applying this rotation to the original $(1, 0, 0)$ and $(0, 1, 0)$ unit vectors converts them to $(\cos\alpha, -\sin\alpha, 0)$ and $(\sin\alpha, \cos\alpha, 0)$ respectively.

Rotating downward by the angle $\phi_z = \pi/2 - \theta_z$ is accomplished by multiplication with the matrix

$$\begin{pmatrix} \cos\phi_z & 0 & \sin\phi_z \\ 0 & 1 & 0 \\ -\sin\phi_z & 0 & \cos\phi_z \end{pmatrix} = \begin{pmatrix} \sin\theta_z & 0 & \cos\theta_z \\ 0 & 1 & 0 \\ -\cos\theta_z & 0 & \sin\theta_z \end{pmatrix}$$

and produces the matrix

$$\begin{pmatrix} \cos\alpha\sin\theta_z & \sin\alpha\sin\theta_z & \cos\theta_z \\ -\sin\alpha & \cos\alpha & 0 \\ -\cos\alpha\cos\theta_z & -\sin\alpha\cos\theta_z & \sin\theta_z \end{pmatrix} .$$

From the last row of this matrix we see that the E and N vertical components of the are $-\cos\alpha\cos\theta_z$ and $-\sin\alpha\cos\theta_z$. As the nominal vertical axis is tilted, the actual vertical measurements in the E and N directions are $v\sin\theta_x$ and $v\sin\theta_y$ respectively, where v is the measurement on the nominal vertical axis. Therefore α is given by *atan2*$(-\sin\theta_y, -\sin\theta_x)$, where *atan2* is the C or Fortran library routine. (If both θ_x and θ_y are zero, any value returned is fine and should the library raise a domain error [EDOM], we can set $\alpha = 0$.)

We now rotate back to global coordinates using the transpose of the matrix in equation (3), which results in the transformation matrix

$$\begin{pmatrix} \sin^2\alpha + \cos^2\alpha \sin\theta_z & \sin\alpha\cos\alpha\sin\theta_z - \sin\alpha\cos\alpha & \cos\alpha\cos\theta_z \\ \sin\alpha\cos\alpha\sin\theta_z - \sin\alpha\cos\alpha & \sin^2\alpha\sin\theta_z + \cos^2\alpha & \sin\alpha\cos\theta_z \\ -\cos\alpha\cos\theta_z & -\sin\alpha\cos\theta_z & \sin\theta_z \end{pmatrix} . \tag{4}$$

Because we are concerned that rotating by α and then by $-\alpha$ may result in precision cancellation, we applied trigonometric identities

$$\begin{array}{rcl} \sin^2\dfrac{\gamma}{2} & = & \dfrac{1-\cos\gamma}{2} \ , \\ \cos^2\dfrac{\gamma}{2} & = & \dfrac{1+\cos\gamma}{2} \ , \\ \sin 2\gamma & = & 2\sin\gamma\cos\gamma \ , \\ & \text{and} & \\ \cos 2\gamma & = & 2\cos^2\gamma - 1 = 1 - 2\sin^2\gamma \end{array}$$

to recast this matrix into the equivalent form

$$\begin{pmatrix} \cos^2\frac{\phi_z}{2} - \cos 2\alpha \sin^2\frac{\phi_z}{2} & -\sin 2\alpha\sin^2\frac{\phi_z}{2} & \cos\alpha\sin\phi_z \\ -\sin 2\alpha \sin^2\frac{\phi_z}{2} & \cos^2\frac{\phi_z}{2} + \cos 2\alpha\sin^2\frac{\phi_z}{2} & \sin\alpha\sin\phi_z \\ -\cos\alpha\sin\phi_z & -\sin\alpha\sin\phi_z & \cos\phi_z \end{pmatrix} , \tag{5}$$

where $\phi_z = \pi/2 - \theta_z$ as before.

After this transformation, the original E direction unit vector is generally no longer at a 90° azimuth. Call the counterclockwise rotation of that direction from due east β. From the first column of matrix (5), we immediately find that

$$\beta = \mathit{atan2}(-\sin 2\alpha \sin^2\frac{\phi_z}{2}, \cos^2\frac{\phi_z}{2} - \cos 2\alpha\sin^2\frac{\phi_z}{2}) \, ,$$

which says that we have calculated the E-N-V components of what the original X-Y-Z unit vector configuration would be if its heading was $\pi/2 - \beta$ instead of the actual heading h of the X component. The two differ by an angle of $(\pi/2 - h) - \beta$ radians measured counterclockwise from E.

This final correction is handled just the same as in (3) with α replaced by $\pi/2 - (h + \beta)$ and leads to the transformation matrix

$$\begin{pmatrix} \sin(h+\beta) & \cos(h+\beta) & 0 \\ -\cos(h+\beta) & \sin(h+\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix} . \tag{6}$$

By transposing the sequence of transformations, we have the following stable conversion of the X-Y-Z sensor orientation to E-N-V global coordinates:

$$\begin{pmatrix} \mathrm{E} \\ \mathrm{N} \\ \mathrm{V} \end{pmatrix} = \begin{pmatrix} \cos^2\frac{\phi_z}{2} - \cos 2\alpha \sin^2\frac{\phi_z}{2} & -\sin 2\alpha\sin^2\frac{\phi_z}{2} & -\cos\alpha\sin\phi_z \\ -\sin 2\alpha \sin^2\frac{\phi_z}{2} & \cos^2\frac{\phi_z}{2} + \cos 2\alpha\sin^2\frac{\phi_z}{2} & -\sin\alpha\sin\phi_z \\ \cos\alpha\sin\phi_z & \sin\alpha\sin\phi_z & \cos\phi_z \end{pmatrix} \times$$

$$\begin{pmatrix} \sin(h+\beta) & -\cos(h+\beta) & 0 \\ \cos(h+\beta) & \sin(h+\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix} .$$

To apply the calculated rotation matrix to our data, we must remember that each data sample constitutes the time derivative of a 3-vector displacement from an origin. In terms of the displacements, the progressive columns of the rotation matrix are the original X, Y and Z unit vectors expressed in E-N-V coordinates. Writing the displacement as the linear combination $pX + qY + rZ$ of coordinate displacements says we should multiply (p, q, r) by the rotation matrix. Because the time derivative commutes with multiplication by a time-independent matrix, multiplying velocities (or accelerations) by the rotation matrix is also the correct operation.

Should we want, as is typical, the first component of our output to be northward motion and the second eastward motion, we need only flip the order of the E and N components to complete the transformation.

EXAMPLE

To confirm the theory, we applied it to nodes in the Forties platform undershoot survey. The first check was to see if, indeed, the transformations did preserve vector magnitudes, so we grabbed an arbitrary G$_1$-G$_2$-G$_3$ sample with tilts of $\theta_x = 0.27°$, $\theta_y = 1.79°$, $\theta_z = 89.99°$, a heading of 353° and converted it to X-Y-Z and N-E-V. The results in Table 1 confirm the norm preservation. Applying the transformation to a node that had 10 to 15 degrees of vertical tilt and a heading 12 degrees away from the North, yielded the correction from Figure 1 to Figure 2 where we have displayed the relative RMS amplitudes for each shot location around that node.

Galperin symmetric		Local axis rotated		Global axis rotated	
G_1	-13805	X	11672	N	10674
G_2	-4078	Y	7478	E	-8846
G_3	6498	Z	-6159	V	-6156
Norm	15168	Norm	15168	Norm	15168

Table 1: Vector norm QC check for node geophone reorientation.

ACKNOWLEDGMENTS

We thank Apache North Sea Limited for providing the data used in this report.

REFERENCES

Grazier, V., 2009, The response to complex ground motions of seismometers with Galperin sensor configuration: Bulletin of the Seismological Society of America, **99**, no. 2B, 1366–1377.

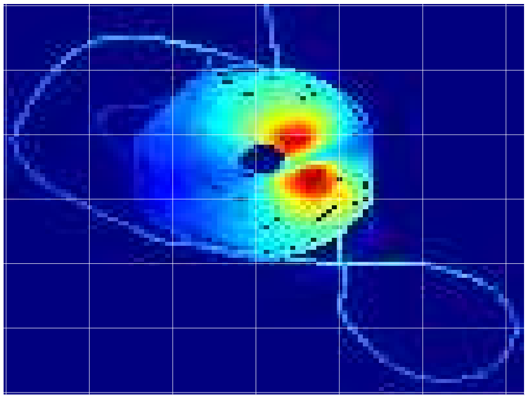


Figure 1: RMS of X component for node with heading 348°. [**ER**] stew1/. x

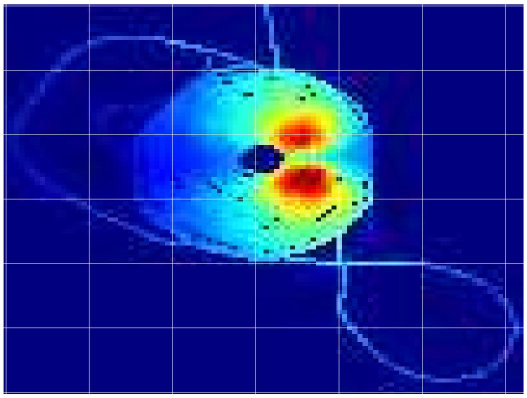


Figure 2: RMS of N component after reorientation. [**ER**] stew1/. n

Fortran calling C: Clock drift correction

Stewart A. Levin

ABSTRACT

Seafloor nodal clock drift is a skew that is taken as distributed linearly across the time between GPS synchronizations. When it came time to correct such skew in a nodal field dataset, I wanted to use the well-tested, accurate 8 point sinc resampler in the Colorado Schools of Mines Seismic Unix (SU) package while staying within the existing Fortran 2003 code I was modifying. In this short note I illustrate how I used the BIND features of modern Fortran to accomplish this.

INTRODUCTION

It is currently infeasible to synchronize actively deployed seafloor nodal recorders with GPS signals. As a result, the internal clocks of the nodes can, and almost surely will, drift away from the GPS standard between the time they are deployed and the time they are retrieved. Lacking any other detail on the drift progress, it is assumed to be linear with time.

Correcting for this drift requires some interpolation between samples, which, as with statics corrections, is best deferred to as late as possible in any processing sequence in order to do the least damage to the signal. In most cases, the drift is treated directly as a static. However, during extended recording such as multi-day passive recording, the change in drift between the start and end of such traces should, and can, be taken into account by adjusting the sample interval from, e.g., 2 msec to 2.000004 msec, and then, for convenience, resampling back to the nominal sample interval.

CWP TO THE RESCUE

While a quick and dirty linear interpolation was enough to generate some preliminary displays of moved-out nodal common receiver records this summer, linear interpolation is generally too damaging to higher frequency content than is normally acceptable. (See, however, Levin (2012) for exceptions.) Fortunately, the widely used SU software (`www.cwp.mines.edu/cwpcodes`) contains a high quality[1] 8-coefficient approximate sinc interpolator `shfs8r` written in C by Dave Hale that has a *maximum* error less than one percent for frequencies up to 60% Nyquist.

Much of the software at SEP is written in Fortran. As the reader is probably aware, invoking a C routine from Fortran has traditionally been a system and compiler dependent process, involving special wrappers to translate between naming conventions and argument handling. Recognizing this, modern Fortran standards, starting with the Fortran 2003 standard, provide a BIND language mechanism for calling C routines. This, in effect, allows

[1] Admittedly, we caused a segfault the first time we used it! Bug fix sent to CWP.

us to write a "universal wrapper" for the C routine by means of an interface definition. For `shfs8r` this definition is:

```
interface
   subroutine CWPshfs8r(dx,nxin,fxin,yin,yinl,yinr,nxout,fxout,yout) &
      BIND(C,name="shfs8r")
      use iso_c_binding, only: c_int, c_float
      REAL(C_FLOAT), VALUE :: dx
      INTEGER(C_INT), VALUE :: nxin
      REAL(C_FLOAT), VALUE :: fxin
      REAL(C_FLOAT), DIMENSION(nxin) :: yin(:)
      REAL(C_FLOAT), VALUE :: yinl
      REAL(C_FLOAT), VALUE :: yinr
      INTEGER(C_INT), VALUE :: nxout
      REAL(C_FLOAT), VALUE :: fxout
      REAL(C_FLOAT), DIMENSION(nxout) :: yout(:)
   end subroutine CWPshfs8r
end interface
```

where I opted to use a different Fortran name for `shfs8r` just to provide a clue as to the C routine's origin.

REFERENCES

Levin, S. A., 2012, Integral operator quality from low order interpolation: SEP-Report, **147**, 323–332.

Anatomy of a header sort

Stewart A. Levin & Yinbin Ma

ABSTRACT

Routine trace sorting at SEP is basically a grid sort. When faced with spiral shooting around sea-bottom nodes, defining an offset grid was nonintuitive. Leveraging the venerable Unix disk-based `sort` program, we were able to complement our `Sort3d` with a gridless header `SortByHdrs` program capable of handling a couple billion traces if needed. Here we highlight some important subtleties in this approach.

INTRODUCTION

For some nodal data QC we applied hyperbolic moveout (LMO-like, not NMO-like) in order to assess orientation, drift and water velocity corrections as a function of offset. In this case a spiral shooting geometry made applying our usual grid-based `Sort3d` problematic. As years ago Dr. Levin had written a sort program for the massive first-generation SEAM dataset (`www.seg.org/resources/research/seam`), he confidently proposed, and we developed, a derivative module that simply sorted on trace header values so that we didn't need an offset-defined grid. Like the SEAM sort, the engine for this new sort is also the venerable Unix `sort` program.

SUBTLETIES

Unix `sort` operates on text files, not binary files. Therefore, to use it to sort SEP binary files, we create a textual index file with printed values for header keys and a sequential trace number. After sorting on the key values, we read back the reordered trace numbers to select and process the actual headers and traces.

Unix `sort` is capable of handling more than $2^{31}-1$ records, but since current seplib query and disk block access routines use four byte integers, that is the maximum number of traces we can sort.

The only (well) supported SEP trace header key formats are 32 bit integers and floats, the printed values of which are understood by GNU `sort`'s numeric comparison routines. To avoid tangling with floating point format precision selection, we use the observation that nonnegative IEEE floating point bit patterns increase monotonically when viewed as integers and hence sort order is preserved when that bit pattern is printed as an integer. For negative floating point numbers, we use the negative of the bit pattern of its absolute value. One issue with the numeric sort in general is that it is locale dependent. We handle this by overriding the default locale with LC_ALL=C for the sort process.

With network mounted filesystems it is notoriously difficult to ensure that a file written by one process can be immediately and fully seen by another process. This is not specific to the processes being on different computers, though that is where it is most often seen (or not seen). To bypass this potential problem, we communicate all data between `SortByHdrs` and `sort` via pipes instead of shared disk files. The SEP *datapath* query is used to set a suitably large directory for `sort`'s scratch files.

There are times when someone might want to interrupt a sort, but does not want to corrupt what has already been sorted due to partially completed I/O. We use a signal handler to set a graceful termination flag in the event of an interrupt.

The more recent versions starting with 8.6 (2010-10-15) of GNU `sort` automatically use multiprocessor parallelism for speed. Checking around at SEP, revealed that all the machines we've used have versions prior to that. A good reason to do some long delayed upgrading.

Finally, we explicitly wait on the sort process to terminate. This avoids possible "zombie" processes and undeleted temporary disk files. In the `SortByHdrs` code, this wait is directly for the `sort` process. When a more traditional *system*() or *popen*() mechanism is used to start a process, it is the shell that spawns the command and not the command itself that is the target of the wait.

SEP PHONE DIRECTORY

Name	**Phone**	**Login Name**
Almomin, Ali	723-0463	ali
Alves, Gustavo	723-9282	gcalves
Barak, Ohad	723-9282	ohad
Barnier, Guillaume	723-6006	gbarnier
Baroni, Claudia	723-5002	
Biondi, Biondo	723-1319	biondo
Biondi, Ettore	723-9282	ettore
Cabrales, Alejandro	723-6006	cabrales
Chang, Jason	724-4322	jasonpc
Claerbout, Jon	723-3717	jon
Clapp, Bob	725-1334	bob
Dahlke, Taylor	724-4322	taylor
Huot, Fantine	723-3187	fantine
Jennings, Joe	723-3187	joseph29
Le, Huy	723-1250	huyle
Leader, Chris	723-0463	chrisl
Levin, Stewart	726-1959	stew
Ma, Yinbin	723-3187	yinbin
Maharramov, Musa	723-1250	musa
Martin, Eileen	723-0463	ermartin
Ruan, Kaixi	723-3187	kaixi
Sarkar, Rahul	725-1625	
Shen, Yi	723-6006	yishen

Dept fax number: (650) 725-7344

E-MAIL

Our Internet address is “*sep.stanford.edu*”; i.e., send Jon electronic mail with the address “*jon@sep.stanford.edu*”.

WORLD-WIDE WEB SERVER INFORMATION

Sponsors who have provided us with their domain names are not prompted for a password when they access from work. If you are a sponsor, and would like to access our restricted area away from work, visit our website and attempt to download the material. You will then fill out a form, and we will send the username/password to your e-mail address at a sponsor company.

STEERING COMMITTEE MEMBERS, 2015-2016

Name	Company	Tel #	E-Mail
Raymond Abma	BP	(281)366-4604	abmar1@bp.com
Francois Audebert	TOTAL	(33)-6-1278-4195	francois.audebert@total.com
Biondo Biondi	SEP	(650)723-1319	biondo@sep.stanford.edu
Jon Claerbout	SEP	(650)723-3717	jon@sep.stanford.edu
Thomas Dickens	ExxonMobil	(713)431-6011	tom.a.dickens@exxonmobil.com
Faqi Liu (Co-chair, 2nd year)	Hess	(823)231-9218	fliu@hess.com
Yi Luo	Saudi Aramco	–	yi.luo@aramco.com
Alejandro Valenciano	PGS	–	alejandro.valenciano@pgs.com
Kathy Zou (Co-chair, 1st year)	PGS	–	kathy.zou@pgs.com

Research Personnel

Ali Almomin graduated from Texas A&M University in 2007 with a BS in Geophysics. Then, he joined Saudi Aramco and worked in several exploration and research departments with a focus on 3D seismic processing and near surface imaging. He joined Stanford Exploration Project in 2009 to pursue a PhD in Geophysics and is currently working on seismic tomography. He is a member of SEG, EAGE, and SPE.

Gustavo Catao Alves graduated from the Federal University of Minas Gerais, Brazil, in 2004 with a BS in Physics. In 2006, he got a MS in Solid State Physics from the same Institution and was awarded Best Student Presenter on the 3rd Conference of the Brazilian Carbon Association. That same year he joined Petrobras, where he worked at the Company's research center for the next 7 years, on projects focused on seismic illumination studies, Reverse Time Migration and Full Waveform Inversion. In 2008, he received an Honorable Mention for Best Poster Paper at SEG's Annual Meeting, with the work entitled "Target oriented illumination analysis using wave equation". He joined Stanford Exploration Project in 2013 to pursue a PhD in Geophysics and is currently working on amplitude studies in long offset data. He is a member of SBGf, SEG, EAGE and AAPG.

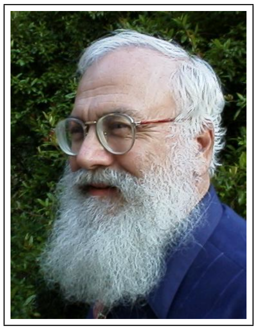
Biondo L. Biondi is professor of Geophysics at Stanford University. Biondo graduated from Politecnico di Milano in 1984 and received an M.S. (1988) and a Ph.D. (1990) in geophysics from Stanford. He is co-director of the Stanford Exploration Project and of the Stanford Center for Computational Earth and Environmental Science. In 2004 the Society of Exploration Geophysicists (SEG) has honored Biondo with the Reginald Fessenden Award. Biondo recently published a book, 3-D Seismic Imaging, that is the first text book to introduce the theory of seismic imaging from the 3-D perspective. The book is published by SEG in the Investigations in Geophysics series. During 2007 gave a one-day short course in 28 cities around the world as the SEG/EAGE Distinguished Short Course Instructor (DISC) . He is a member of AGU, EAGE, SEG and SIAM.

Ettore Biondi received a B.S. (2010) in geology from the University of Genoa, an M.S. (2012) in geophysics from the University of Pisa, and a diploma (2013) in computational chemistry from the Scuola Normale Superiore of Pisa. He spent almost two years as a research fellow at the University of Milan within the geophysics section of the Earth Science Department. In the summer of 2014, he joined the SEP and started his Ph.D. at the Stanford University. He is interested in multi-parameter inversion and wave-equation velocity estimation methods.

Alejandro Cabrales-Vargas obtained his Bachelors degree in Geophysics in the University of Mexico in 2002. He has being working for Petroleos Mexicanos since 2002, initially in seismic interpretation for oil and gas exploration, and more recently in the supervision of depth imaging processes. He obtained his Masters Degree in Geophysics in the University of Oklahoma in 2011. He joined SEP in the fall of 2014, and is currently working towards his PhD.

Jason Chang received his B.A. in geophysics from the University of California, Berkeley, in 2010. He joined SEP in autumn of 2011 and is currently working toward a Ph.D. in geophysics. He is a student member of SEG and AGU.

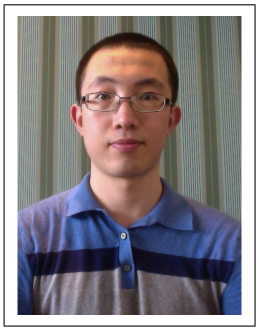
Jon F. Claerbout (M.I.T., B.S. physics, 1960; M.S. 1963; Ph.D. geophysics, 1967), professor at Stanford University, 1967. Emeritus 2008. Best Presentation Award from the Society of Exploration Geophysicists (SEG) for his paper, *Extrapolation of Wave Fields.* Honorary member and SEG Fessenden Award "in recognition of his outstanding and original pioneering work in seismic wave analysis." Founded the Stanford Exploration Project (SEP) in 1973. Elected Fellow of the American Geophysical Union. Authored three published books and five internet books. Elected to the National Academy of Engineering. Maurice Ewing Medal, SEG's highest award. Honorary Member of the European Assn. of Geoscientists & Engineers (EAGE). EAGE's highest recognition, the Erasmus Award.



Robert Clapp received his B.Sc. (Hons.) in Geophysical Engineering from Colorado School of Mines in May 1993. He joined SEP in September 1993, received his Masters in June 1995, and his Ph.D. in December 2000. He is a member of the SEG and AGU.

Joseph Jennings Joseph received his BS in Geophysical Engineering from Colorado School of Mines in 2014. Following graduation, he worked in land seismic processing R&D at ION Geophysical. In the summer of 2015, he joined the SEP as a PhD student where he hopes to pursue research in simultaneous source imaging and imaging of unconventional reservoirs.

Stewart A. Levin was acting director of the Stanford Exploration Project during Jon Claerbout's 1993-4 sabbatical year. After a distinguished career in industry at Mobil and Halliburton, he has returned to Stanford as a consulting professor in the Department of Geophysics.



Yinbin Ma graduated from University of Science and Technology of China in 2009 with a B.S. in physics. In 2013, he obtained a M.S. in Condensed Matter Theory at Rice University. Yinbin moved to Stanford in 2013 and is currently working towards a Ph.D. in the Institute for Computational and Mathematical Engineering. He started doing research at SEP from June 2014. He is a student member of the SEG.

Musa Maharramov received his diploma c.l. in Mathematics from Baku State University in 1993. In 1995, he received a Ph.D. in Mathematics (Differential Equations) from the Azerbaijan Institute of Mathematics for his thesis "Asymptotic Solutions of Quasilinear Parabolic Equations". In 1995-2001, he worked as an IT& Comms consultant, later IT manager, at Fluor Corporation in Baku. In 2001, he joined BP Azerbaijan, and in 2006 he started his work at BP High Performance Computing as a Computational Scientist. In that role he worked with the BP Imaging Team on mathematical, algorithmic and computational aspects of seismic migration and inversion. Musa joined SEP in the fall of 2011 and is pursuing his second Ph.D., in Geophysics. He is currently working on regularization of seismic inversion problems through the application of Geological/Geomechanical constraints and advanced optimisation techniques. Musa is a member of the SEG, EAGE and SIAM.

Ossian O'Reilly joint PhD Candidate in Geophysics at Stanford and in Scientific Computing at Linkoping University, Sweden. Specializes in the development of numerical methods for wave propagation within fluid-filled fractures related to oil-and-gas and volcano applications.

Kaixi Ruan Kaixi was born and raised in Shanghai, China. After passing the concours (a competitive exam organized by the Embassy of France in China) for the Grandes Ecole, she attended the Ecole Polytechnique in Palaiseau, France majoring in Applied Mathematics. She is currently working towards her Masters in the Institute for Computational and Applied Mathematics.

Yi Shen got her Bachelors degree in Acoustics, from Dept. of Electronic Science and Engineering, Nanjing University, China in 2010. She joined SEP in the fall of 2010, and is currently working toward a Ph.D. in Geophysics at Stanford University. She is a student member of the SEG.

SEP ARTICLES PUBLISHED OR IN PRESS

Almomin, A., and Biondi, B., 2012, Tomographic full waveform inversion: Practical and computationally feasible approach. SEG Technical Program Expanded Abstracts, 1–5.

Almomin, A., and Biondi, B., 2013, Tomographic full waveform inversion (TFWI) by successive linearizations and scale separations. SEG Technical Program Expanded Abstracts, 1048–1052.

Almomin, A., and Biondi, B., 2014, Preconditioned tomographic full waveform inversion by wavelength continuation. SEG Technical Program Expanded Abstracts, 944–948.

Almomin, A., and Biondi, B., 2014, Generalization and amplitude normalization of tomographic full waveform inversion. SEG Technical Program Expanded Abstracts.

Barak, O., Jaiswal, P., de Ridder, S., Giles, J., Brune, R. and Ronen, S., 2014, Six-component seismic land data acquired with geophones and rotation sensors: wave-mode separation using 6C SVD: SEG Technical Program Expanded Abstracts (Submitted).

Chang, J., de Ridder, S., and Biondi, B., 2013, Power spectral densities and ambient noise cross-correlations at Long Beach: SEG Technical Program Expanded Abstracts **32**, 2196–2200.

Chang, J., Nakata, N., Clapp, R.G., de Ridder, S., and Biondi, B., 2014, High-frequency surface and body waves from ambient noise cross-correlations at Long Beach, CA: SEG Technical Program Expanded Abstracts **33**.

Chang, J., and B. Biondi, 2015, Rayleigh-wave tomography using traffic noise at Long Beach, CA: SEG Technical Program Expanded Abstracts **34**.

Dahlke, T., Beroza, G., Chang, J., and de Ridder, S., 2014, Stochastic variability of velocity estimates using eikonal tomography on the Long Beach data set: SEG Technical Program Expanded Abstracts **33** .

Dahlke, T., Biondi, B., Clapp, R., 2015, Domain decomposition of level set updates for salt segmentation: SEG Technical Program Expanded Abstracts **33** (Submitted).

Grobbe, N., F.C. Schoemaker, M.D. Schakel, S.A.L. de Ridder, E.C. Slob, and D.M.J. Smeulders, 2012, Electrokinetic fields and waves: Theory, experiments, and numerical modeling. Geophysical Research Abstracts, Vol. **14**, EGU2012–10636.

de Ridder, S., and B. Biondi, 2012, Continuous passive seismic monitoring of CCS projects by correlating seismic noise: A feasibility study. 74th Conference & Technical Exbition, EAGE, Extended Abstracts (Accepted).

de Ridder, S., and B. Biondi, 2012, Reservoir monitoring by passive seismic interferometry. SEG/SPE/AAPG Joint Workshop, La Jolla, CA, 24-29 June 2012 (Accepted).

de Ridder, S., 2012, Ambient seismic noise correlations for reservoir monitoring. SEG Technical Program Expanded Abstracts **31** (Submitted).

de Ridder, S., and B. Biondi, 2012, Continuous reservoir monitoring by ambient seismic noise tomography. SEG Technical Program Expanded Abstracts **31** (Submitted).

Schoemaker, F.C., N. Grobbe, M.D. Schakel, S.A.L. de Ridder, E.C. Slob, and D.M.J. Smeulders, 2012, Experimental validation of the electrokinetic theory and development of seismoelectric interferometry by cross-correlation: International Journal of Geophysics (Accepted).

Farghal, M.S., and Levin, S.A., 2012, Hunting for microseismic reflections using multiplets: SEG Technical Program Expanded Abstracts **31** (Submitted).

Guitton, A., Ayeni, G., and Esteban, D.A., 2012, Constrained full-waveform inversion by model reparameterization: Geophysics, 77, No. 2, R117–R127.

Guitton, A., and Esteban, D.A., 2012, Attenuating crosstalk noise with simultaneous source full waveform inversion: Geophysical Prospecting (In Press).

Guitton, A., 2012, Blocky regularization schemes for full waveform inversion: Geophysical Prospecting (In Press).

Halpert, A., 2012, Edge-preserving smoothing for segmentation of seismic images: SEG Technical Program Expanded Abstracts **31**, 1–5.

Halpert, A., Clapp, R.G., and B. Biondo, 2014, Salt delineation via interpreter-guided seismic image segmentation, Interpretation: **2**, T79–T88.

Arevalo, Humberto and Stewart A. Levin, 2014, Well and Seismic matching with ArcGIS and ProMAX via KML: Esri International User Conference Paper 901, San Diego, 15 July.

Leader, C., and R. Clapp, 2012, Least squares reverse time migration on GPUs - balacing IO and computation: 74th Conference & Technical Exbition, EAGE, Extended Abstracts (Accepted).

Leader, C., and A. Almomin, 2012, How incoherent can we be? Phase encoded linearised inversion with random boundaries: SEG Technical Program Expanded Abstract **31** (Submitted).

Levin, Stewart A. and Fritz Foss, 2014, Downward continuation of Mars SHARAD data: SEG Technical Program Expanded Abstract **33** (Submitted).

Li, Y., Y. Zhang, and J. Claerbout, 2012, Hyperbolic estimation of sparse models from erratic data: Geophysics **77**, 1–9.

Li, Y., P. Shen, and C. Perkins, 2012, VTI migration velocity analysis using RTM: SEG Technical Program Expanded Abstract **31**.

Li, Y., Image-guided WEMVA for azimuthal anisotropy: SEG Technical Program Expanded Abstract **32**.

R.E. Plessix and Y. Li, 2013, Waveform acoustic impedance inversion with spectral shaping: Geophysical Journal International **195**(1), 301–314, 2013.

Li, Y., B. Biondi, D. Nichols, and R. Clapp, 2014, Wave equation migration velocity analysis for VTI models: Accepted for publication in Geophysics, 2014.

Li, Y., M. Wong, and R. Clapp, 2014, Equivalent accuracy by fractional cost: overcoming temporal dispersion: Submitted to SEG 2014.

Li, Y., D. Nichols, and G. Mavko, 2014, Stochastic rock physics modeling for seismic anisotropy: Submitted to SEG 2014.

Li, Y., B. Biondi, R. Clapp, and D. Nichols, 2014, Rock physics constrained anisotropic wave-equation migration velocity analysis: Submitted to SEG 2014.

Maharramov M., and Biondi, B., 2014, Joint full-waveform inversion of time-lapse seismic data sets. SEG Technical Program Expanded Abstracts, 954–959.

Maharramov M., and Biondi, B., 2014, Joint 4DFWI with model-difference regularization. SEG-AGU Summer Research Workshop. Advances in Active+Passive "Full Wavefield" Seismic Imaging: From Reservoirs to Plate Tectonics.

Maharramov M., and Biondi, B., 2015, Robust Simultaneous Time-lapse Full-waveform Inversion with Total-variation Regularization of Model Difference. EAGE Technical Program Expanded Abstracts (accepted).

Maharramov M., 2015, Efficient Finite-difference Modelling of Acoustic Wave Propagation in Anisotropic Media with Pseudo-sources. EAGE Technical Program Expanded Abstracts (accepted).

Maharramov M., and Biondi, B., 2015, Simultaneous TV-regularized time-lapse FWI with application to field data. SEG Technical Program Expanded Abstracts (submitted).

Martin E.R., J. Ajo-Franklin, S. Dou, N. Lindsey, T.M. Daley, B. Freifeld, M. Robertson, A.

Wagner, and C. Ulrich, 2015, Interferometry of ambient noise from a trenched distributed acoustic sensing array. SEG Technical Program Expanded Abstracts (submitted).

Ajo-Franklin, J., N. Lindsey, S. Dou, T.M. Daley, B. Freifeld, E.R. Martin, M. Robertson, C. Ulrich, and A. Wagner, 2015, A field test of distributed acoustic sensing for ambient noise recording. SEG Technical Program Expanded Abstracts (submitted).

Li, Y., H. Yang, E.R. Martin, K. Ho, and L. Ying, 2015, Butterfly factorization. Multiscale Modeling and Simulation (submitted). http://arxiv.org/abs/1502.01379

Shen, Y., B. Biondi, R. Clapp, and D. Nichols, 2013, Wave-equation migration Q analysis (WEMQA): EAGE Workshop on Seismic Attenuation Extended Abstract 2013.

Shen, Y., B. Biondi, R. Clapp, and D. Nichols, 2014, Wave-equation migration Q analysis (WEMQA): SEG Technical Program Expanded Abstracts.

Li, Y., Y. Shen and P. Kang, 2015, Integration of seismic and fluid-flow data: a two-way road linked by rock physics :77th EAGE Conference and Exhibition.

Wong, M., and Ronen, S., and Biondi, B.L., 2012, Joint imaging with streamer and ocean bottom data. SEG Technical Program Expanded Abstracts, 1–5.

Wong, M., and Ronen, S., and Biondi, B.L., 2012, Imaging with multiples using linearized full-wave inversion. SEG Technical Program Expanded Abstracts, 1–5.

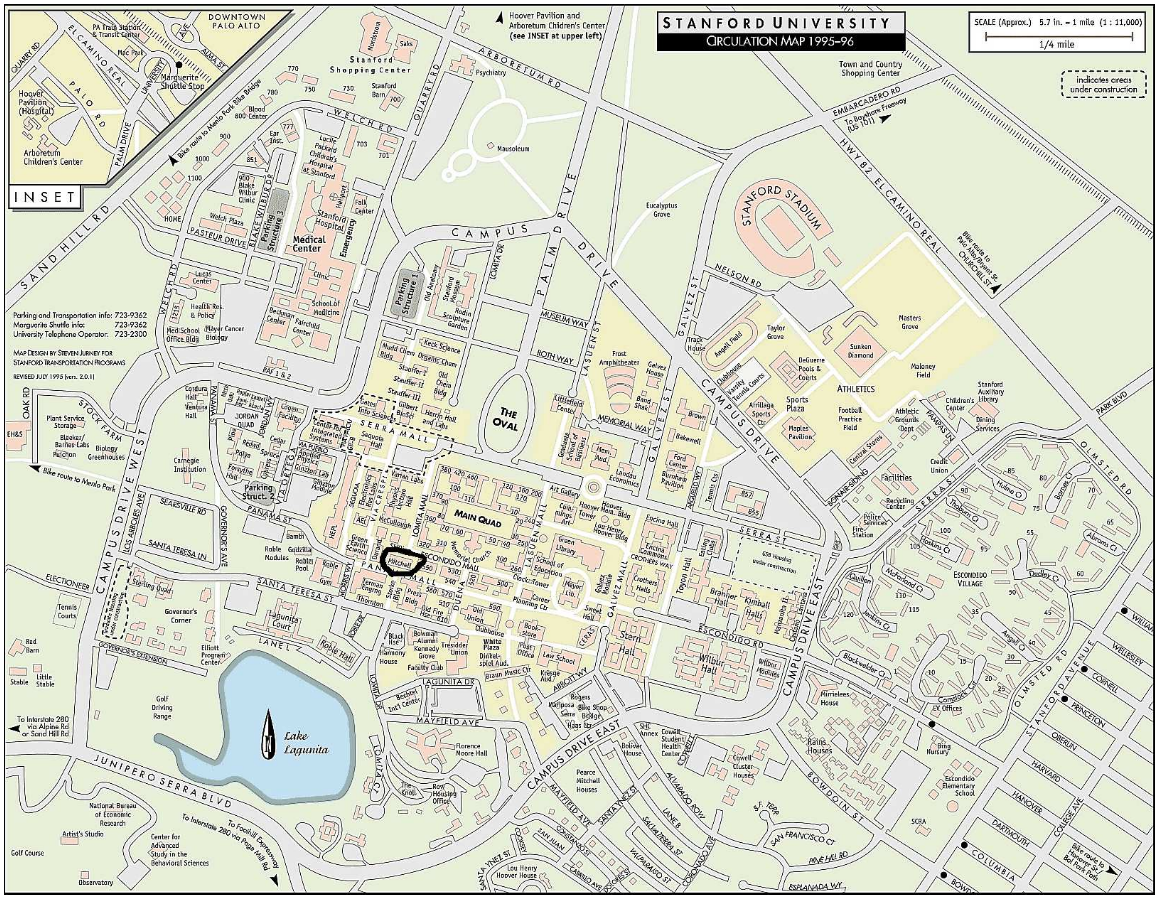
Wong, M., 2013, Handling salt reflection in Least-squares RTM SEG Technical Program Expanded Abstracts, 3921–3925.

Wong, M., B. Biondi, S. Ronen, C. Perkins, M. Merritt, V. Goh, and R. Cook, 2014, Robbust least squares RTM on the 3D Deimos ocean bottom node dataset SEG Technical Program Expanded Abstracts, (Submitted)

Wong, M., B. Biondi, and S. Ronen, 2014, Imaging with multiples using least-squares reverse-time migration. The Leading Edge, (submitted)

Huy Le and Stewart A. Levin, 2014, Removing shear artifacts in acoustic wave propagation in orthorhombic media: SEG Technical Program Expanded Abstracts, 486–490.

Huy Le, Biondo Biondi, Robert G. Clapp, and Stewart A. Levin, 2015, Using a nonlinear acoustic wave equation for anisotropic inversion: SEG Technical Program Expanded Abstracts (Submitted).


STANFORD UNIVERSITY
CIRCULATION MAP 1995–96
SCALE (Approx.) 5.7 in. = 1 mile (1 : 11,000)
1/4 mile
indicates areas under construction
Hoover Pavilion and Arboretum Children's Center (see INSET at upper left)
INSET
DOWNTOWN PALO ALTO
PA Train Station & Transit Center
Marguerite Shuttle Stop
Hoover Pavilion (Hospital)
Arboretum Children's Center
EL CAMINO REAL
PALO RD
QUARRY RD
UNIVERSITY AVE
ALMA ST
PALM DRIVE
Bike route to Menlo Park Bike Bridge
Stanford Shopping Center
Nordstrom
Saks
Stanford Barn
Psychiatry
ARBORETUM RD
Mausoleum
Town and Country Shopping Center
EMBARCADERO RD
To Bayshore Freeway (US 101)
HWY 82 EL CAMINO REAL
Bike route to Palo Alto/Bryant St.
CHURCHILL ST
SAND HILL RD
WELCH RD
Lucile Packard Children's Hospital at Stanford
Stanford Hospital
Medical Center
Emergency
Falk Center
Blake Wilbur Clinic
Parking Structure 3
PASTEUR DRIVE
Welch Plaza
Clinic
School of Medicine
Beckman Center
Fairchild Center
Lucas Center
Health Res. & Policy
Med School Office Bldg
Mayer Cancer Biology
Parking and Transportation info: 723-9362
Marguerite Shuttle info: 723-9362
University Telephone Operator: 723-2300
MAP DESIGN BY STEVEN JURNEY FOR STANFORD TRANSPORTATION PROGRAMS
REVISED JULY 1995 (vers. 2.0.1)
CAMPUS DRIVE
PALM DRIVE
Eucalyptus Grove
STANFORD STADIUM
NELSON RD
Parking Structure 1
Old Anatomy
Stanford Museum
Rodin Sculpture Garden
LOMITA DR
MUSEUM WAY
ROTH WAY
Keck Science
Mudd Chem Bldg
Organic Chem
Stauffer I
Stauffer II
Stauffer III
Old Chem Bldg
Gilbert Biology
Herrin Hall and Labs
Cogen Facility
Center for Integrated Systems
JORDAN QUAD
SERRA MALL
Sequoia Hall
THE OVAL
Frost Amphitheater
Galvez House
Band Shak
Littlefield Center
Graduate School of Business
Mem. Aud.
MEMORIAL WAY
Landau Economics
Ford Center
Burnham Pavilion
Brown
Bakewell
Track House
Angell Field
Taylor Grove
DeGuerre Pools & Courts
Clubhouse
Varsity Tennis Courts
Sunken Diamond
Masters Grove
Maloney Field
ATHLETICS
Arrillaga Sports Ctr
Sports Plaza
Maples Pavilion
Football Practice Field
Athletic Grounds Dept
Children's Center
Stanford Auxiliary Library
Dining Services
Central Stores
Facilities
Credit Union
PARK BLVD
PAMPAS LN
OLMSTED RD
CAMPUS DRIVE
GALVEZ ST
LASUEN ST
OAK RD
STOCK FARM
EH&S
Plant Service Storage
Bleeker/Barnes Labs
Pulchon
Biology Greenhouses
Carnegie Institution
Bike route to Menlo Park
Cordura Hall
Ventura Hall
Cedar
Pine
Redwood
Spruce
Cypress
Polya
Forsythe Hall
Parking Struct. 2
PANAMA ST
VIA ORTEGA
Varian Labs
Electronics Res. Labs
Physics Lecture Hall
McCullough
AEL
HEPL
Ginzton Lab
MAIN QUAD
Art Gallery
Hoover Tower
Hoover Mem. Bldg
Lou Henry Hoover Bldg
Cummings Art
Green Library
Encina Hall
Encina Commons
CROTHERS WAY
Crothers Halls
Toyon Hall
SERRA ST
Eating Clubs
GSB Housing under construction
Branner Hall
Kimball Halls
Recycling Center
Police Services
Fire Station
ESCONDIDO VILLAGE
Quillen
McFarland Ct.
Hoskins Ct.
Thoburn Ct.
Hulme Ct.
Barnes Ct.
Abrams Ct.
Dudley Ct.
Jenkins Ct.
Angell Ct.
Blackwelder Ct.
Comstock Ct.
EV Offices
CAMPUS DRIVE EAST
CAMPUS DRIVE WEST
SEARSVILLE RD
SANTA TERESA LN
LOS ARBOLES AVE
GOVERNOR'S AVE
Bambi
Roble Modules
Roble Pool
Godzilla
Roble Gym
Green Earth Sciences
Mitchell
Geology
Terman Engrng
Thornton
Skilling
Press Bldg
Old Fire Hse
Memorial Church
ESCONDIDO MALL
LASUEN MALL
GALVEZ MALL
LOMITA MALL
School of Education
Clock Tower
Meyer Lib
Career Planning Ctr
Old Union
Clubhouse
Sweet Hall
Tresidder Union
White Plaza
Dinkelspiel Aud.
Bookstore
Post Office
Law School
Kresge Aud.
Braun Music Ctr
ABBOTT WY
Stern Hall
Wilbur Hall
Wilbur Modules
ESCONDIDO RD
Mirrielees House
Rains Houses
Bing Nursery
Escondido Elementary School
SCRA
ELECTIONEER
Tennis Courts
Sterling Quad
Governor's Corner
Elliott Program Center
SANTA TERESA ST
Lagunita Court
Roble Hall
LANE L
GOVERNOR'S EXTENSION
Black Hse
Harmony House
Bowman Alumni House
Kennedy Grove
Faculty Club
Bechtel Int'l Center
LAGUNITA DR
MAYFIELD AVE
Red Barn
Stable
Little Stable
Golf Driving Range
To Interstate 280 via Alpine Rd or Sand Hill Rd
Lake Lagunita
JUNIPERO SERRA BLVD
Florence Moore Hall
The Knoll
Row Housing Office
Rogers
Mariposa
Sarra
Haas Ctr
Bike Shop
Bridge
Bolivar House
SHC Annex
Cowell Student Health Center
Cowell Cluster Houses
Pearce Mitchell Houses
CAMPUS DRIVE EAST
ALVARADO ROW
S.F. TERR
SAN FRANCISCO CT
BOWDOIN ST
PINE HILL RD
ESPLANADA WY
National Bureau of Economic Research
Artist's Studio
Golf Course
Center for Advanced Study in the Behavioral Sciences
To Foothill Expressway
To Interstate 280 via Page Mill Rd
Observatory
Lou Henry Hoover House
SANTA YNEZ ST
SAN JUAN
CAMPO
VALPARAISO ST
SALVATIERRA ST
COWPER
STANFORD AVENUE
WILLIAMS
WELLESLEY
CORNELL
PRINCETON
OBERLIN
HARVARD
HANOVER
DARTMOUTH
COLUMBIA
BOWDOIN
COLLEGE AVE
Bike route to Hanover St. & Bol Park Path