

Rapid 4D FWI using a local wave solver

Alison Malcolm¹ and Bram Willemsen²

Abstract

Much of the computational cost involved in full-waveform inversion comes from the solution of the wave equation in a large domain. These computations must be done for the entire domain through which we expect waves to pass for a particular survey, despite the fact that our region of interest is often significantly smaller. In addition to the wasted time spent propagating waves through less important parts of the model, computing updates on the entire domain may result in slower convergence of the inversion algorithm due to the larger model space. This can be especially important in 4D seismic monitoring, where we often see the majority of changes within a small subregion of the total domain, such as the reservoir. We present a local wave solver that accurately computes the solution of the wave equation within only a subdomain of the region covered by the survey, representing a significant cost saving in the computation of full-waveform inversion. We also show how this solver can improve the resulting velocity estimates in full-waveform inversion for time-lapse applications and observe that the local solver requires fewer iterations to converge than does the full-domain solver.

Introduction

Full-waveform inversion (FWI) typically involves solving the forward problem of propagating waves from the source to the receiver multiple times per iteration (see, e.g., Virieux and Operto, 2009, and references therein). The calculation of these wavefields is the primary computational cost of an FWI algorithm. However, when waves from the same sources are propagated through a model that has changed little between iterations, much of this computation is repeated from one iteration to the next. For 4D seismic applications, the impact of this repetition on computational cost is greater because we are typically interested in only a small region of the model (e.g., the reservoir), and we may have several data sets to invert. If we are able to perform FWI on only a small subset of the subsurface model, while still taking all of the data into account, we can improve the speed of the inversion process while still preserving the accuracy of the recovered results. To do this, we require an algorithm that solves the wave equation on a subdomain, giving exactly the same wavefield (on that subdomain) as would be obtained when solving the full wave equation in the full domain. We presented such an algorithm in Willemsen et al. (2016), building upon the work of van Manen et al. (2007). Here, we show how this algorithm can be used to increase the efficiency of FWI for the particular case of 4D velocity change estimation. If there are significant changes outside the reservoir region, as can occur due to compaction for example, then our method may not result in significant cost savings as the number and size of the relevant subdomains grow.

Full-waveform inversion for 4D is challenging because of the nonuniqueness of the problem (there are multiple models that fit

the data). When performing FWI for multiple surveys, spurious time-lapse changes may be generated when the results for different data sets disagree in more than just the regions of true time-lapse perturbation (i.e., the models come from different local minima of the objective function). Several approaches have been proposed to mitigate this problem including double-difference FWI (DDFWI) (Watanabe et al., 2004; Denli and Huang, 2009), image domain waveform tomography (Yang et al., 2014a), and model-space regularization methods (Zhang and Huang, 2013a; Maharramov and Biondo, 2014b; Asnaashari et al., 2015). Another option is to redatum the data to subsurface locations (Mulder, 2005), allowing the inversion to focus on a smaller domain (Yang et al., 2012). There is also a growing body of work on 4D-specific survey planning and algorithm development designed to collect and exploit the right level of redundancy in the data (e.g., Denli and Huang, 2010; Shabelansky et al., 2013; Zhang and Huang, 2013b; Wason et al., 2014; Oghenekohwo et al., 2015).

Local solver

As mentioned above, we require an accurate solution of the wave equation in a subdomain of the full model. We begin by splitting the model into two parts: the interior (Ω) and exterior (Ω^c) of the truncated domain. Within the exterior, Ω^c , the model will not be updated and is assumed known. In this paper, we use the result of an FWI inversion of the baseline data set for this model. The model in Ω^c along with the initial guess within the interior of the truncated domain Ω make up the background model, m_0 . The wavefield, excited by the source $f(x, \omega)$, in m_0 is the background field, u_0 , and satisfies

$$\nabla^2 u_0 + k_0^2 u_0 = f, \quad (1)$$

where $k_0(x) = \omega / c_0(x)$ is the wavenumber at angular frequency ω

in the background model represented by the velocity $c_0(x)$.

Within Ω , we are going to update the model, m , where $m(x) = m_0(x) + \delta m(x)$ and the model perturbation, δm , is non-zero only

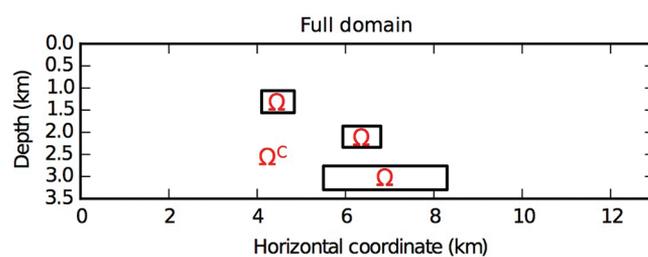


Figure 1. This illustrates the splitting of the domain into the local domain(s) Ω and the background, Ω^c . In this example, we show three local domains on which we compute time-lapse updates. Alternatively, we could make one larger domain, with an increased cost proportional to the ratio of the sizes of the three smaller domains to the single larger domain.

¹Memorial University of Newfoundland.

²Massachusetts Institute of Technology.

<http://dx.doi.org/10.1190/tle35121053.1>

within Ω . Here, the total wavefield $u(x) = u_0(x) + \delta u(x)$ (where δu is the scattered field), satisfies

$$\nabla^2 u + k^2 u = f, \quad (2)$$

with $k(x) = \frac{\omega}{c(x)}$ the wavenumber in the updated model represented by the velocity $c(x)$.

Clearly, within Ω we can simply solve equation 2 and get an accurate wavefield within that truncated domain, solving a relatively small problem. The problem is to determine what boundary conditions are necessary so that the field within Ω is the same whether we locally solve the wave equation (only within region Ω) or globally solve the wave equation (throughout regions Ω and Ω^c). The basic idea for how to do this, as suggested by van Manen et al. (2007) for the time domain, goes back to the representation theorem (e.g., Fokkema and van den Berg, 2013), which states that the scattered field outside a domain can be uniquely and completely determined by the field on the boundary. For our problem, this amounts to relating the scattered field δu to the field and its normal derivative on the boundary. (Note that this is the entire scattered field, including all multiple scattering between the interior and exterior of the local domain.) Writing the representation theorem in the discrete form and cancelling terms gives us the following expression (see Willemsen et al., 2016 for a derivation):

$$\delta u(y, \omega) = - \sum_{x \in \partial\Omega} \frac{1}{h^2} \left(u^{\partial\Omega}(x, \omega) G_0^{\partial\Omega_+}(x, y, \omega) - u^{\partial\Omega_+}(x, \omega) G_0^{\partial\Omega}(x, y, \omega) \right). \quad (3)$$

In equation 3, $G_0(x, y)$ is the Green's function between x and y in the background model, m_0 , and h is the distance between adjacent mesh points. The superscript indicates where the point x is located, with $\partial\Omega_+$ being a gridpoint one layer to the inside of $\partial\Omega$ (which is the boundary between Ω and Ω^c). The point y is in Ω^c or $\partial\Omega$. These Green's functions represent the wavefield between the actual physical sources and receivers and points on or near the boundary of Ω , as well as Green's functions between different points on or near this boundary. Doing the derivation directly in the discrete domain avoids issues that can arise with the use of sharp corners in the domain or variations in the final equations as a result of different discretization schemes.

From equation 3, we obtain the scattered field on the boundary. We can then compute the total field on the boundary by simply forming $u = u_0 + \delta u$, where u_0 is known from the solution of equation 1. Once we have the total field on the boundary, it acts as a boundary condition for our second-order accurate solver. Using this boundary condition we can compute the total field inside Ω by setting up a system of equations including equation 2 inside Ω and equation 3 on the boundary. This system is significantly smaller than what we would have if we solved equation 2 throughout the entire domain, which is the reason the local solver is significantly faster. From the solution of this system on the boundary and using equation 3 we can then compute the total wavefield at any required points in Ω^c (e.g., receiver locations).

In looking at equation 3, we see that we do require Green's functions computed in the background model for the full domain, meaning that we require a number of wave solves in the full domain. However, once these Green's functions are available, we can compute the wavefield in Ω and at the receiver locations many times for very little additional cost, allowing each iteration of FWI to be done much more quickly. We can reuse these Green's functions because, after each iteration of FWI the model m changes locally in Ω , but equation 3 remains valid; no additional full-domain simulations are required for subsequent local wavefield computation. In the examples shown here, each iteration is roughly 10 to 15 times faster in the local domain than it is in the full domain. This allows for many more inversions to be run to update the model within this domain, testing different parameters, initial models, etc., without the overhead of solving the wave equation in the full domain (as the Green's functions necessary for equation 3 can be saved).

The key advantage in this approach as compared with that of van Manen et al. (2007) is that van Manen et al. require that the wavefield satisfies a time-dependent boundary condition. In their approach, the multiplication in equation 3 for the scattered field becomes a convolution in the time domain. At each time step, this updates a time-dependent Dirichlet boundary condition, which exactly matches the incoming wavefield and by doing so decouples the local domain. The total cost of this procedure is asymptotically quadratic in time, reducing the computational benefit of having a local solver. Other local solvers involve approximations to the wavefield and are discussed in Willemsen et al. (2016).

We demonstrate the accuracy of the solver using the Marmousi II model (Martin et al., 2002). Figure 2a shows the true baseline model, and Figure 2b shows a zoomed-in section with three time-lapse perturbations. The three black boxes represent the boundaries of the truncated domains/domains of interest.

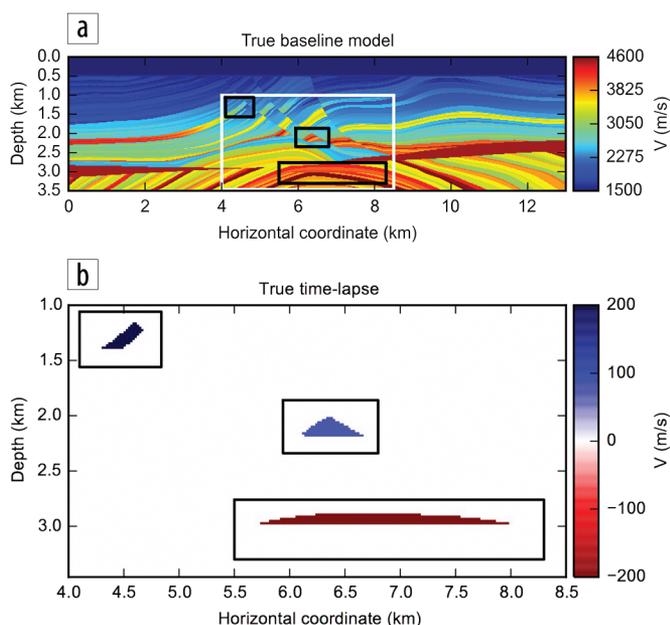


Figure 2. To demonstrate our method, we use the Marmousi II model, shown in (a), and restrict it to three subdomains, shown in (b), in which we introduce the illustrated time-lapse changes.

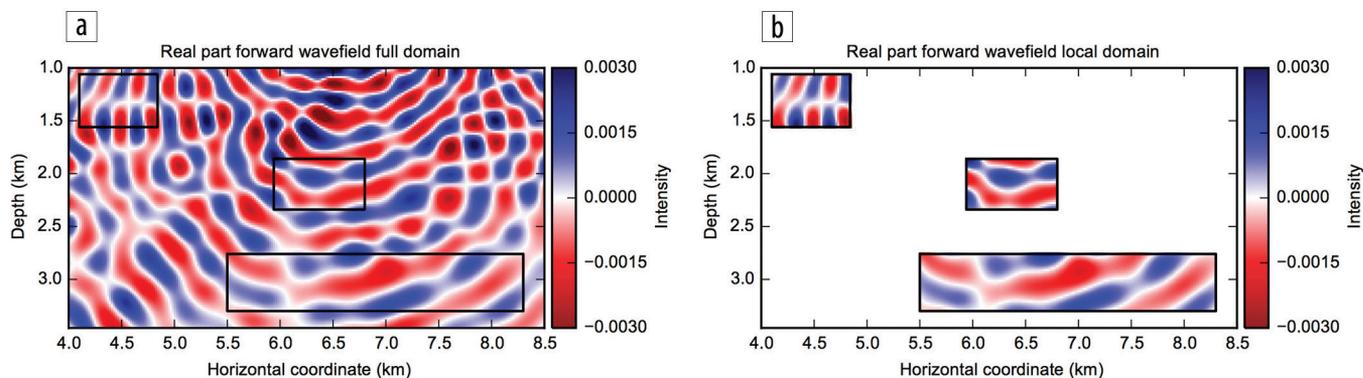


Figure 3. To test the local solver, we compare the wavefields in the Marmousi II model, (a) using a standard full-domain finite-difference solver and (b) using the local solver. The two fields agree to the accuracy of double-precision calculations.

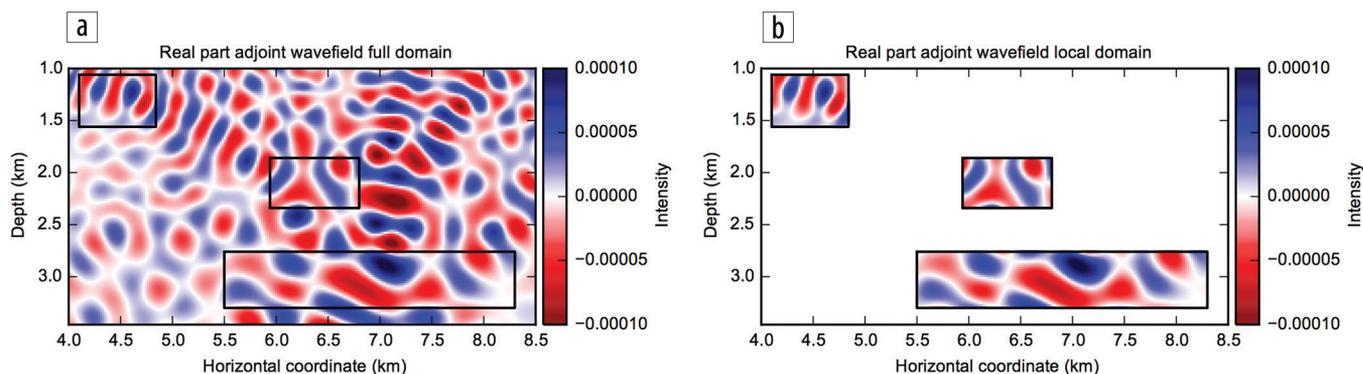


Figure 4. Comparison of the adjoint field using (a) the full-domain solver and (b) local domain solver. The two fields agree to machine precision.

To verify the exactness of the local solver, we first compute the wavefield in the usual way with a finite difference solution to the Helmholtz equation 2 in the entire domain (Ω and Ω^c together). The real component of the resulting wavefield is shown in Figure 3a for a shot at $x = 6.6$ km and a frequency of 8.0 Hz. Next, we compute the field using the local solver in the truncated domain Ω . This result is shown in Figure 3b. Differencing the two wavefields within the truncated domain shows that the solver is indeed exact to the expected accuracy of double-precision calculations (approximately 10^{-16}). This demonstrates that the local solver preserves all orders of scattering between the model updates δm and the background model m_0 .

Computing the adjoint field

We have established that we are able to compute the wavefield in a truncated subdomain and that it is exactly the same as that computed in the full domain. We now turn our attention to the second wavefield that is needed in FWI: the adjoint field. As explained by Virieux and Operto (2009) and others, velocity-model updates are computed from the cross correlation of the forward and adjoint fields.

For this problem, the adjoint field satisfies the same set of wave equations given in equations 1 and 2, with the exception that the source f is replaced with the data residuals δd at the receiver locations.

$$\delta d = d - u, \quad (4)$$

where d are the data, and u continues to denote the total wavefield. The total wavefield u is computed in the truncated domains, but

not yet at the receiver locations. To obtain the field at the receiver locations, we use equation 3 to obtain the scattered field δu , at the receiver locations, which can be added to the background field u_0 to obtain the total field u , from which we can then compute the data residual.

Now that we have δd , we can compute the adjoint field by simply solving the same system of equations used for the forward problem but with the adjoint source, δd , replacing f . To obtain the incident field on the boundary, we simply multiply δd by the precomputed Green's function. Note that we do not need to recompute the full-domain Green's functions, as they remain the same for both problems. Figure 4a shows the real component of the full domain adjoint field, again for the physical source at $x = 6.6$ km. Figure 4b shows the local-domain equivalent. Again their difference is negligible.

At each FWI iteration, we can repeat the procedure described above. We first compute the local wavefield, which requires refactoring a small matrix as the model has changed only in Ω . Next, we propagate this field to the receivers using equation 3 and evaluate the residuals to obtain the adjoint sources. We then compute the background adjoint wavefields on the boundary using the precomputed background Green's functions and solve for the total adjoint wavefield which, when combined with the forward field, gives us the gradient within Ω .

Again, the key advantage of this method is that the computation of the full-domain Green's functions is done only once. As a result, all subsequent wave solves happen in only the truncated domain. Thus, we are able to compute a gradient update throughout the truncated domain at significantly reduced cost. The more iterations

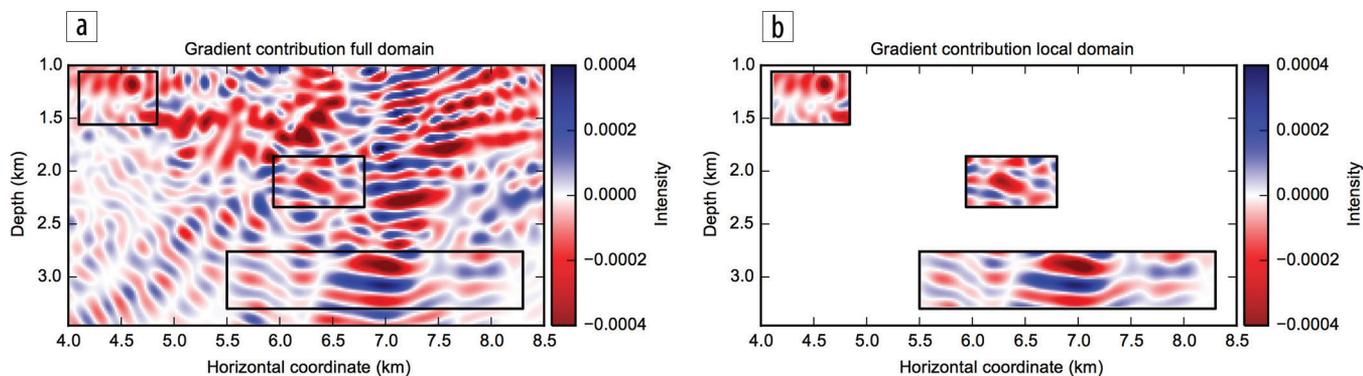


Figure 5. Gradients computed with (a) the full-domain and (b) the local domain solver. As the gradient is the product of the forward and adjoint fields, it is not surprising that it retains the machine-precision accuracy of those two fields.

(or new inversions) that are performed without changing the background model, m_0 , the greater is the reduction in computation time.

Figure 5 shows the gradient contribution of the source at $x = 6.6$ km using the full and local domain solvers. This gradient contribution is the multiplication of the forward field and the complex conjugate of the adjoint field for the same shot. Since both of these fields were numerically exact, the gradient contribution for the source is also numerically exact. This gradient forms a part of the model update; the full gradient is the summation of these contributions from all the sources and is added to the initial model to produce an updated model (here only within Ω).

4D inversion strategy

Having illustrated how the inversion process can be performed in the truncated domain, we move on to a discussion of how this can be used to enhance 4D imaging. In this paper, we focus on double-difference full-waveform inversion (DDFWI) (Watanabe et al., 2004; Denli and Huang, 2009), because in that method we explicitly solve for an update to the background model, rather than solving for the updated model itself. This means that we are already focusing our attention on a small subregion, which can be our truncated domain, Ω . Once we have defined our region(s) of interest, we can perform DDFWI and obtain an update to our velocity model within that region. This can be thought of as an extension of Zhang and Huang (2013a), Maharramov and Biondi (2014a), or Asnaashari et al. (2015) where regularization is used to restrict changes to a subdomain of the full model.

In this work, we have adapted codes from the PySIT library (Hewett et al., 2013) to do the computations. The initial background model is formed via a standard frequency-domain acoustic FWI using 64 shots and 651 receivers spread over the entire surface of the Marmousi model. We invert six frequencies sequentially to obtain our baseline model shown in Figure 6.

Having obtained a good baseline velocity model, we can now begin to compute the updates from 4D changes. Because we are not updating the model outside the truncated domain, we compute the full-domain Green's functions required by the local solver only once for the entire inversion process in the model shown in Figure 6. For the adjoint field, we need to compute a new data residual for each monitor data set, as is described above. In DDFWI, we replace the standard data residual shown in equation 4, with

$$\delta d = (d_{\text{monitor}} - d_{\text{baseline}}) - (u_{\text{monitor}} - u_{\text{baseline}}), \quad (5)$$

where d_{baseline} and d_{monitor} are the data for the baseline and monitor field data sets, and u_{baseline} and u_{monitor} are the modeled baseline and monitor data in the current iteration of FWI. DDFWI gets its name from the two differences in equation 5: one between modeled and field data and one between baseline and monitor data residuals. We minimize the δd in equation 5 to determine a model update, δm , between the baseline and monitor data sets.

In our target-oriented approach, we restrict the model update to be within the truncated domain, Ω , which in our case is the union of the three domains shown in Figure 1. This significantly reduces the computational cost of the algorithm but also restricts the recovered change to a particular subset of the model space. In Yang et al. (2014b), it was observed that DDFWI predicted changes throughout the model, leading to the development of a tomography method in the image space (Yang et al., 2014a). That method allowed for the use of narrower-offset data by minimizing the difference between baseline and monitor images rather than data sets. Here, we take a complementary approach, explicitly restricting the changes to a subset of the model space but still working directly with the data rather than images. As previously mentioned, this approach is similar to the methods proposed in Zhang and Huang (2013a), Maharramov and Biondi (2014a, 2014b), or Asnaashari et al. (2015). The key difference here is that rather than regularizing the inversion, we compute the updates themselves only in the subdomain of interest, significantly reducing the computational cost.

The results of our approach are shown in Figure 7. Here, we compare the results of DDFWI applied to the entire domain to the results where the changes have been restricted to the

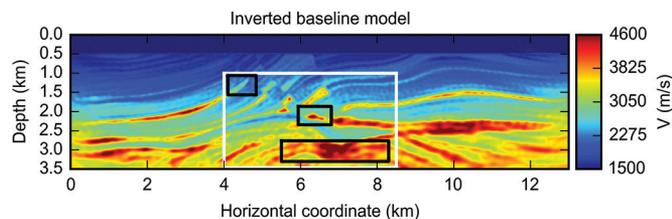


Figure 6. The recovered baseline model, inverted using standard methods from the baseline data set. The true model is shown in Figure 2. We clearly have better resolution in the shallower regions of the model than in the deeper regions.

truncated domain. We see that both the inversions in the full domain (Figure 7a) and in the truncated domain (Figure 7c) are able to recover the time-lapse perturbations fairly well. There is error in the shape of the deepest perturbation, which we suspect is caused by imperfections in the background model. We conclude this because other parameters (shape, size, velocity contrast) are similar for the other perturbations, but comparing Figures 6 and 2a illustrates that there are clearly some errors in the deeper parts of our estimated baseline model.

Also in Figure 7, we compare the number of iterations required for the two methods to converge. In the full domain, many more iterations are required to recover the magnitude of the time-lapse perturbations. Figures 7b and 7c show the results for three iterations. We observe that the full-domain solution has not correctly estimated the magnitude of the perturbations. In the full domain, we need 16 iterations (Figure 7a) to achieve results similar to those obtained in the truncated domain with only three iterations. Although we are not certain why this is the case, we speculate that the reduced model space in the local inversion allows the FWI algorithm to converge more rapidly, thus further reducing the cost of FWI (though a reduction in iterations may also be present when using a full-domain solver if model updates are restricted to a local domain). Care must be taken in using the local solver that the changes across the boundaries are not too abrupt. In this case, if we continue beyond three iterations we will see artifacts caused by jumps in the model perturbation, δm . This is easily mitigated using tapers and slightly larger domains.

Discussion

Our goal in using a local solver is both to improve the resulting time-lapse estimates and to reduce computational cost. We compare the cost of a full-domain solver with the truncated solver, bearing in mind that each method produces different products. The full-domain solver gives a model update throughout the domain while the local solver gives an update only within the truncated domain. To quantify the reduction in computational cost compared to a global solution over the full domain, we must consider two aspects of the process. First, there is the cost of computing the background Green's functions. This is done once before the processing begins. Second, there is the cost of computing the field and its adjoint within the truncated domain, Ω . Together, these two steps make up the majority of the computational cost of the algorithm. Although it appears that fewer iterations are required when using the local solver, we choose to compare the costs of the two methods with the same number of iterations to obtain a conservative estimate of the speedup. We show this speedup in Table 1, looking at different numbers of inversions and iterations. We consider multiple inversions because this is done when refining the inversion parameters and processing multiple monitor data sets. Once the setup step is done, each iteration is about a factor of 10 to 15 times faster for the examples shown in this paper (the numbers in Table 1 use a factor of 13).

For one inversion with one iteration (first row), the local solver takes about six times as long as the full-domain solver. By contrast, for 20 inversions with 20 iterations each, the local solver takes

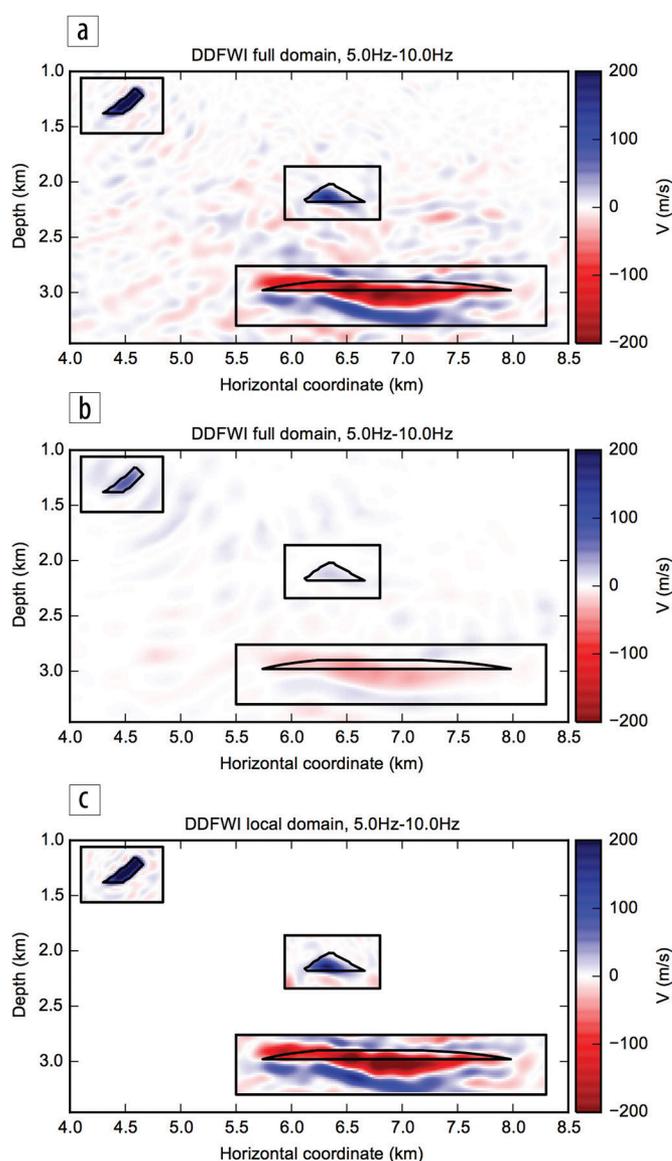


Figure 7. Results of DDFWI (a) using 16 iterations in the entire domain, (b) using three iterations in the entire domain, and (c) using three iterations in the local domain. Although all three inversions do a reasonable job of recovering the location and basic shape of the perturbations, using only three iterations on the full domain is not sufficient to recover the magnitude of the perturbations, whereas this is sufficient for the local domain. The relative cost of producing these images is explored in the discussion section.

about 9% of the time required by the full-domain solver. This difference accumulates because once the offline work generating the background Green's functions is done, each single wave solve executes much faster (here 13 times faster). The background work is done only once for the entire process, before the first inversion is initiated. This speedup could lead to more interactive workflows and allow the exploration of more inversion scenarios.

The local solver discussed in this paper is relevant to many other geophysical problems in addition to 4D imaging. It can be used in any situation in which a local update is required. An example of such a situation is subsalt imaging, in which the accurate determination of the base of salt is an important problem in velocity-model building. In this case, the local solver can be used to update the location of the bottom of the salt, isolating the

Table 1. This table compares the computational cost of using the local solver and the full-domain solver. We use a speedup factor of 13, which is what we observe in the example shown here — i.e., local solves are assumed to cost 1/13 of a full-domain solve. The computational cost of the local FWI includes both the precomputation of (~1225) full-domain Green's functions (done once) and subsequent local solves (assuming 3.5 wave solves per source including the line search, for 60 sources this gives 210 wave solves per iteration). We assume that the same number of iterations are used for each algorithm. The third column shows the ratio of the time for the local version of FWI to perform the inversion to that of the full-domain solve.

Number of Iterations	Number of Inversions	$t_{\text{local}}/t_{\text{full}}$
1	1	5.9
8	1	0.81
20	1	0.37
8	8	0.16
20	8	0.11
20	20	0.091

computations to a very small domain in the immediate vicinity of the salt, as shown by Willemsen et al. (2016).

Work is also ongoing on how to use the ideas behind the local solver discussed here to couple elastic and acoustic wave propagation. As discussed by, e.g., Chapman et al. (2014) and others, solving the elastic wave equation is much more costly than solving the acoustic equation. In situations in which there is a localized region within which we are interested in determining the elastic parameters, it would be advantageous to solve the elastic wave equation only within a localized domain. For this concept to work, the acoustic approximation must be sufficiently accurate for the regions outside of the elastic subdomain. Zhu and McMechan (2012) show that this is the case for phase inversion. Willemsen and Malcolm (2016) explore this idea incorporating the elastic local solver discussed by Robertsson and Chapman (2000). This sort of technique can also be used in concert with mechanisms such as those described by Chapman et al. (2014) that attempt to approximate the elastic field using an augmented solution of the acoustic wave equation.

Localization in the model space allows us to consider more expensive time-lapse processing ideas for 4D, such as the alternating FWI suggested in Yang et al. (2014c), or cross updating as proposed by Maharramov and Biondi (2014a). This opens up the possibility of truly estimating the uncertainty in 4D changes, which, as noted in Asnaashari et al. (2015), is not currently computationally tractable.

Conclusions

There are many applications in full-waveform inversion in which we attempt to determine Earth properties within a subset of the total region of the subsurface sampled by the data. In these cases, calculating wavefields outside the region of interest at each iteration of inversion is wasted effort. In this paper, we show that the work required for FWI is reduced by locally computing the wavefield only within the domain of interest. We also observe that we require fewer iterations when using the local solver. As a result, velocity estimates can be improved while controlling the computational cost. This is of particular interest and importance

for 4D seismic applications in which we are interested in an isolated region of the subsurface (the reservoir) and have good knowledge of the properties outside the reservoir, and do not expect these areas to change between surveys. The method also allows us to consider elastic inversions and to potentially extend the calculations to 3D without extraordinary costs. **■**

Acknowledgments

Funding for this work comes from MIT as well as from the Natural Science and Engineering Research Council of Canada, the Research and Development Corporation of Newfoundland and Labrador, and from Chevron Canada Ltd. We would like to thank our colleagues at MIT and MUN as well as Johan Robertsson and Filippo Broggin of ETH for useful discussion. We are also grateful for the work done by David Johnston and two anonymous reviewers to improve the presentation of the material.

Corresponding author: amalcolm@mun.ca

References

- Asnaashari, A., R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux, 2015, Time-lapse seismic imaging using regularized full-waveform inversion with a prior model: which strategy?: *Geophysical Prospecting*, **63**, no. 1, 78–98, <http://dx.doi.org/10.1111/1365-2478.12176>.
- Chapman, C., J. Hobro, and J. Robertsson, 2014, Correcting an acoustic wavefield for elastic effects: *Geophysical Journal International*, **197**, no. 2, 1196–1214, <http://dx.doi.org/10.1093/gji/ggu057>.
- Denli, H., and L. Huang, 2009, Double-difference elastic waveform tomography in the time domain: 79th Annual International Meeting, SEG, Expanded Abstracts, 2302–2306, <http://dx.doi.org/10.1190/1.3255320>.
- Denli, H., and L. Huang, 2010, Elastic-wave sensitivity propagation: *Geophysics*, **75**, no. 3, T83–T97, <http://dx.doi.org/10.1190/1.3428403>.
- Fokkema, J., and P. van den Berg, 2013, *Seismic applications of acoustic reciprocity*: Elsevier Science.
- Hewett, R., L. Demanet, and the PySIT Team, 2013, PySIT: Python seismic imaging toolbox v0.5. (Release 0.6).
- Maharramov, M., and B. Biondi, 2014a, Joint full-waveform inversion of time-lapse seismic data sets: 84th Annual International Meeting, SEG, Expanded Abstracts, 954–959, <http://dx.doi.org/10.1190/segam2014-0962.1>.
- Maharramov, M., and B. Biondi, 2014b, Robust joint full-waveform inversion of time-lapse seismic data sets with total-variation regularization: SEP report 155, arXiv preprint arXiv:1408.0645.
- Martin, G. S., K. J. Marfurt, and S. Larsen, 2002, Marmousi-2: An updated model for the investigation of AVO in structurally complex areas: 72nd Annual International Meeting, SEG, Expanded Abstracts, 1979–1982, <http://dx.doi.org/10.1190/1.1817083>.
- Mulder, W., 2005, Rigorous redatuming: *Geophysical Journal International*, **161**, no. 2, 401–415, <http://dx.doi.org/10.1111/j.1365-246X.2005.02615.x>.
- Oghenekohwo, F., R. Kumar, E. Esser, and F. J. Herrmann, 2015, Using common information in compressive time-lapse full-waveform inversion: 77th Conference and Exhibition, EAGE, Extended Abstracts, <http://dx.doi.org/10.3997/2214-4609.201413086>.

- Robertsson, J., and C. Chapman, 2000, An efficient method for calculating finite-difference seismograms after model alterations: *Geophysics*, **65**, no. 3, 907–918, <http://dx.doi.org/10.1190/1.1444787>.
- Shabelansky, A., A. Malcolm, and M. Fehler, 2013, Data-driven estimation of the sensitivity of target-oriented time-lapse seismic imaging to source geometry: *Geophysics*, **78**, no. 2, R47–R58, <http://dx.doi.org/10.1190/geo2012-0175.1>.
- van Manen, D.-J., J. O. A. Robertsson, and A. Curtis, 2007, Exact wave field simulation for finite-volume scattering problems: *The Journal of the Acoustical Society of America*, **122**, no. 4, EL115–EL121, <http://dx.doi.org/10.1121/1.2771371>.
- Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: *Geophysics*, **74**, no. 6, WCC1–WCC26, <http://dx.doi.org/10.1190/1.3238367>.
- Wason, H., F. Oghenekohwo, and F. J. Herrmann, 2014, Randomization and repeatability in time-lapse marine acquisition: 84th Annual International Meeting, SEG, Expanded Abstracts, 46–51, <http://dx.doi.org/10.1190/segam2014-1677.1>.
- 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, <http://dx.doi.org/10.1190/1.1845221>.
- Willemsen, B., A. Malcolm, and W. Lewis, 2016, A numerically exact local solver applied to salt boundary inversion in seismic full-waveform inversion: *Geophysical Journal International*, **204**, no. 3, 1703–1720, <http://dx.doi.org/10.1093/gji/ggv547>.
- Willemsen, B., and A. Malcolm, 2016, Coupling a local elastic solver to a background acoustic model to estimate phase variation: 85th Annual International Meeting, SEG, Expanded Abstracts, 3945–3949, <http://dx.doi.org/10.1190/segam2016-13867595.1>.
- Yang, D., Y. Zheng, M. Fehler, and A. Malcolm, 2012, Target-oriented time-lapse waveform inversion using virtual survey: 82nd Annual International Meeting, SEG Expanded Abstracts, <http://dx.doi.org/10.1190/segam2012-1308.1>.
- Yang, D., A. Malcolm, and M. Fehler, 2014a, Using image warping for time-lapse image domain wavefield tomography: *Geophysics*, **79**, no. 3, WA141–WA151, <http://dx.doi.org/10.1190/geo2013-0424.1>.
- Yang, D., A. Malcolm, M. Fehler, and L. Huang, 2014b, Time-lapse walkaway vertical seismic profile monitoring for CO₂ injection at the SACROC enhanced oil recovery field: A case study: *Geophysics*, **79**, no. 2, B51–B61, <http://dx.doi.org/10.1190/geo2013-0274.1>.
- Yang, D., A. Malcolm, and M. Fehler, 2014c, Time-lapse full waveform inversion and uncertainty analysis with different survey geometries: 76th Conference and Exhibition, EAGE, Extended Abstracts, <http://dx.doi.org/10.3997/2214-4609.20141120>.
- Zhang, Z., and L. Huang, 2013a, Double-difference elastic-waveform inversion with prior information for time-lapse monitoring: *Geophysics*, **78**, no. 6, R259–R273, <http://dx.doi.org/10.1190/geo2012-0527.1>.
- Zhang, Z., and L. Huang, 2013b, Reservoir monitoring using double-difference elastic-waveform inversion with sparse-array data: 83rd Annual International Meeting, SEG Expanded Abstracts, 998–1002, <http://dx.doi.org/10.1190/segam2013-1266.1>.
- Zhu, X., and G. A. McMechan, 2012, Elastic inversion of near- and postcritical reflections using phase variation with angle: *Geophysics*, **77**, no. 4, R149–R159, <http://dx.doi.org/10.1190/geo2011-0230.1>.