Geophysical Journal International

Geophys. J. Int. (2016) **204,** 1703–1720 doi: 10.1093/gji/ggv547
GJI Seismology

A numerically exact local solver applied to salt boundary inversion in seismic full-waveform inversion

Bram Willemsen,[1] Alison Malcolm[1,2] and Winston Lewis[3]

[1] *Earth Resources Lab, Massachusetts Institute of Technology, Cambridge, MA* 02139, *USA. Email: bramwillemsen@gmail.com*
[2] *Department of Earth Sciences, Memorial University of Newfoundland, St John's NL* A1B 3X5, *Canada*
[3] *Schlumberger Petrotechnical Services,* 3750 *Briarpark Drive, Houston, TX* 77042, *USA*

Accepted 2015 December 21. Received 2015 December 15; in original form 2015 March 25

SUMMARY

In a set of problems ranging from 4-D seismic to salt boundary estimation, updates to the velocity model often have a highly localized nature. Numerical techniques for these applications such as full-waveform inversion (FWI) require an estimate of the wavefield to compute the model updates. When dealing with localized problems, it is wasteful to compute these updates in the global domain, when we only need them in our region of interest. This paper introduces a local solver that generates forward and adjoint wavefields which are, to machine precision, identical to those generated by a full-domain solver evaluated within the region of interest. This means that the local solver computes all interactions between model updates within the region of interest and the inhomogeneities in the background model outside. Because no approximations are made in the calculation of the forward and adjoint wavefields, the local solver can compute the identical gradient in the region of interest as would be computed by the more expensive full-domain solver. In this paper, the local solver is used to efficiently generate the FWI gradient at the boundary of a salt body. This gradient is then used in a level set method to automatically update the salt boundary.

Key words: Numerical solutions; Numerical approximations and analysis; Controlled source seismology; Computational seismology; Wave propagation; Acoustic properties.

1 INTRODUCTION

Historically, first arrivals and primary reflections have been used to construct increasingly complex velocity models. A stratified velocity model can be obtained by applying Dix's equation (Dix 1955) to stacking velocities obtained from common midpoint gathers. Reflection tomography has the potential to give higher resolution velocity models (Stork & Clayton 1991) from pre-stack reflection arrivals. Migration Velocity Analysis (MVA; Brandsberg-Dahl *et al.* 1999) and Wave Equation Migration Velocity Analysis (WEMVA; Sava & Biondo 2004) extract additional information from the reflections by extending the migrated image in the subsurface offset or angle domains.

Although the evolution of these reflection-based methods has resulted in velocity models with increasing resolution, they use only a subset of the recorded seismic data. Full-waveform inversion (FWI), on the other hand, views the entire trace as data. FWI has the potential to invert for material properties such as the pressure-wave (*P*-wave) speed, shear wave (*S*-wave) speed and density (Fichtner 2010). Distributions of these parameters are useful in resource exploration as well as in whole earth imaging. Monitoring changes of these parameters over time is useful in reservoir monitoring, where changing fluid distributions influence the bulk properties of the rock (Arts *et al.* 2009). The development of the theoretical framework of FWI is often attributed to the work of Lailly (1983) and Tarantola (1984), but the large amount of data contained in the seismic traces initially made application of the method infeasible. At the end of the 1990s, some small-scale 2-D tests were performed, most notably those done by Pratt (1999). With the continuous increase in computational power and algorithmic efficiency, FWI is now often used in the field. An extensive collection of case studies is available, with some recent 3-D examples given by Plessix *et al.* (2010), Prieux *et al.* (2013) and Bansal *et al.* (2013). A good overview of FWI, including references to more field applications, is given by Virieux & Operto (2009).

Additional increases in computational power and further improvements in algorithmic efficiency will continuously enable field data applications of FWI using higher frequencies and more realistic physics. The algorithmic improvements of FWI can be grouped into several categories. Some algorithmic studies focus on the inverse problem to improve the convergence speed (Métivier *et al.* 2013) or on the objective function to make the problem less sensitive to incorrect initial models (van Leeuwen & Mulder 2007; Jugnon & Demanet 2013). Other approaches focus on reducing the number of wavefield simulations required (Krebs *et al.* 2009; Herrmann *et al.* 2013). Efficient computation of the wavefields is also studied

extensively; for instance, for the frequency domain see the work of Wang *et al.* (2011) and Zepeda-Núñez & Demanet (2014). Van Manen *et al.* (2005, 2006) show how interferometric methods can be used to compute wavefields efficiently. When there is interest in enhancing the model in a subset of the entire domain, the associated reduction in model size can also be exploited to increase the efficiency. There are at least two different methodologies for this. One is redatuming, where the seismic survey is propagated to the region of interest. This reduces the model space and typically also the amount of data used, leading to improved speeds. The other approach attempts to generate accurate local wavefields for velocity updates (scatterers) in the region of interest. These wavefields can then be used to update the velocity model. The work introduced in this paper falls into the latter category. In addition, an interesting procedure for doing local inversion in many small domains is introduced by Haffinger (2012). He poses the FWI problem in a novel way that makes possible the subdivision of the global problem into many small local problems. All of the local domains can communicate with each other, but the method can only start from a homogenous initial model.

The idea of seismic redatuming is described by Berryhill (1984). It estimates the measured response between a virtual source and receiver at a datum by trying to remove the effects of the overburden in the data. Schuster & Zhou (2006) give an overview of model- and correlation-based redatuming methods. A model-based redatuming method requires an estimate of the Green's function. An example is given by Dong *et al.* (2009), who use a subsalt datum to improve a Kirchhoff-migrated image locally. In another example, Yang *et al.* (2012) use modelled Green's functions between the acquisition surface and a datum to localize a 4-D seismic survey. In a different approach, using a modelled direct wave, Broggini *et al.* (2014) create virtual sources inside the earth at locations where no source or receiver is present. They then use a multidimensional decovolution approach to approximate the redatumed reflection response. By contrast, correlation-based methods do not require an estimate of the Green's function (Schuster 2009). This is especially useful when the overburden above the region of interest is complex. An example of a correlation-based redatuming approach was given by Byun *et al.* (2010), in which they cross-correlate recorded data from surface shots at two horizontal wells. In this way, the receivers in one well are replaced by virtual sources with receivers in the other well. The virtual data for waves traveling between the horizontal wells are then used to invert for velocity changes resulting from CO_2 sequestration for instance. The analytical expression for redatuming the sources and receivers requires an integral over the real source and receiver positions (Wiggins 1984). Real surveys have finite spacing between sources and between receivers, so the integrals turn into summations (Bevc 1995). The finite sampling of the integrals and the limited source and receiver coverage introduce artefacts in the redatumed data, as was demonstrated by Haffinger (2012). Mulder (2005) expressed the redatuming as an inverse problem where he tried to correct these artefacts while successfully removing most of the effects of the stripped over-burden. Tang & Biondi (2011) generated a local data set for target-oriented MVA by using a generalized Born modelling approach, which shares similarities with redatuming. What all of these methods have in common is the fact that they modify the surface data recordings. In doing so, they introduce varying degrees of artefacts. An ideal approach for local problems would be to compute the wavefield exactly within the region of interest; this wavefield can then be used to update the velocity model locally. This is the problem addressed in this paper.

Several authors have investigated the computation of such local wavefields for many different model configurations. A special configuration is a model inhomogeneity embedded within an infinite homogeneous medium. By using an exact non-reflecting boundary condition, Teng (2003) generated the wavefield in a subdomain containing a collection of scatterers. His approach was limited to localized scatterers embedded in an infinite homogeneous space where waves exiting the scattering region do not re-enter it. Gillman *et al.* (2013) also investigated the case of a homogeneous medium with inhomogeneities restricted to a bounded subdomain. They computed the wavefield by partitioning the inhomogeneous subdomain into a collection of small volumes. Impedance-to-impedance maps connect the volumes to their neighbours. The volumes are merged recursively in a tree-like fashion to generate an impedance-to-impedance map for the entire subdomain. This impedance-to-impedance map then connects the subdomain to the homogeneous exterior. Grote & Sim (2011) introduce an efficient method for solving for the interaction between multiple inhomogeneous subdomains embedded in a homogeneous infinite space. The non-reflecting nature of the homogeneous medium gives exact knowledge of the arrival times for waves leaving one scattering domain and arriving at another. This locality in time makes an efficient time domain implementation possible when truncating the computational domain.

The problem with these aforementioned local wavefield generators is that realistic geological media are neither homogeneous nor infinite and the background medium will also scatter. Robertsson & Chapman (2000) approximated the wavefield with high accuracy within a locally perturbed subdomain of the full domain and then propagated this to the receivers in the inhomogeneous background model. But their formulation discarded the perturbed wavefield scattering outside of the region of interest and subsequently re-entering the subdomain. A time reversal mirror introduced by Masson *et al.* (2013) made the same approximation and therefore did not generate the exact local wavefield. Van Manen *et al.* (2007) introduced an exact domain truncation method in the time domain that will work for inhomogeneous background models. The key property of this method is that it includes waves that leave the subdomain where the background model can be perturbed, and then scatter an arbitrary number of times in the background media before re-entering the subdomain. Including all of these interactions between the subdomain, with its model perturbations, and the inhomogeneous background model, unfortunately results in a loss of locality in time. The computational domain is restricted to the subdomain by enclosing it with a time-dependent Dirichlet boundary condition that exactly matches incoming and outgoing waves. Each node on this time-dependent Dirichlet boundary depends on the wavefield at all the subdomain boundary nodes at all past times. This adds a significant cost to the implementation of the exact non-reflecting boundary condition in the time domain. Including these multiple interactions between the subdomain and the full domain is essential for physical applications (Vasmel *et al.* 2013) where waves are expected to exit and re-enter the embedded domain multiple times.

The method that is introduced in this paper also generates the wavefield exactly in a locally perturbed subdomain. The approach shares similarities with the work of van Manen *et al.* (2007), but implements a boundary condition around the subdomain in the frequency domain. By using a frequency domain formulation, application of the boundary conditions is much less computationally intensive than the approach of van Manen *et al.* (2007). The boundary condition matches incoming and outgoing waves exactly and conserves all orders of scattering. Once the local solver is set up, local wavefield generation and inversion is much faster than doing

full domain solves. The wavefield in the subdomain is, up to machine precision, the same as the wavefield that is generated by the full domain solver. This makes post-processing of a full domain FWI procedure intuitive. Identical wavefields and FWI gradients are generated within the subdomain as those that would have been generated by the full domain solver.

In this paper, the local solver is applied to the problem of automatically updating the shape of a salt body in the velocity model. The salt body is parametrized in an implicit way using a level set method (Lewis *et al.* 2012). The work builds on that of Dorn *et al.* (2000) and references therein who previously used level set methods for shape estimation in geophysical inverse problems. In this paper, the FWI gradient at the boundary of the salt body is used to determine the direction and amplitude of shape perturbation. Instead of calculating the gradient in the entire computational domain, the local solver is used to generate the gradient within a subdomain containing a part of the salt boundary. The salt body is then iteratively updated within this subdomain at high speed. We demonstrate that improving the salt body estimate locally results in a more accurate subsalt image.

2 THEORY

In order to iteratively update the shape of the salt body in the velocity model, we need the FWI gradient at the current estimate of the salt boundary. The reason for this is that the gradient contains information about whether the velocity estimate needs to be increased or decreased, which corresponds to expanding or contracting the salt respectively. Instead of computing the full domain gradient and updating the entire salt body, we choose an inversion subdomain around a section of the salt body. From now on we refer to this inversion subdomain as the truncated domain. Within the truncated domain we require the gradient to update the salt body locally. In this study, we assume that the sediment velocity is known accurately enough to focus the gradient with the correct sign at the frequencies we invert for. Even though we invert only for the salt boundary, the gradient could simultaneously update the sediment velocity within the truncated domain as well. In this section, we introduce a local Helmholtz solver that is capable of computing the local forward and adjoint wavefields required for the gradient. This local Helmholtz solver requires a number of full domain Green's functions in the initial model. The local solver will then generate the exact wavefields when the model changes are restricted to the interior of the truncated domain. This section will explain how this is achieved. The local solver may also have potential applications in other localized inverse problems, such as 4-D seismics.

Fig. 1 illustrates the problem, with the truncated domain enclosed by a red dashed line. The full computational domain Ω is the union of all the domains:

$$\Omega = A \cup B \cup \partial C \cup C. \tag{1}$$

We split the computational grid outside of the truncated domain C into regions A and B. This choice will aid the derivations in Appendix A. As long as the boundary conditions are in region A our derivations will not be influenced by their characteristics. The model perturbations $\delta m(x)$ are restricted to C (i.e. the yellow nodes), which is the interior of the truncated domain, that is,

$$\text{supp}\,(\delta m) \subset C, \tag{2}$$

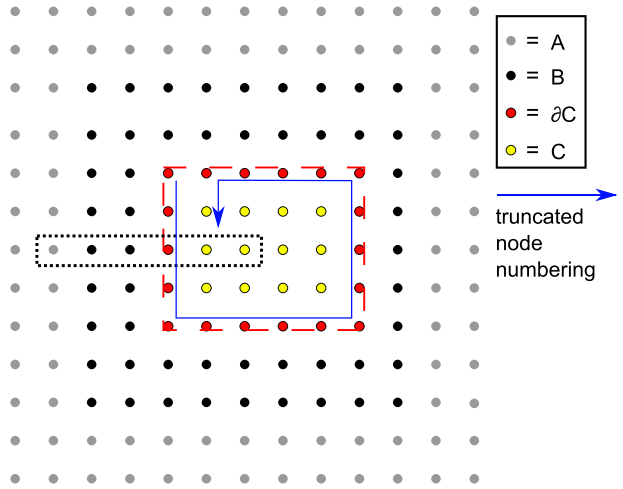

Figure 1. The geometrical setup of the problem. Eq. (1) defines the full computational domain Ω as the union of A, B, ∂C and C. The truncated domain is enclosed by the dashed red line. The nodes on the boundary of the truncated domain are in the set ∂C. The interior nodes of the truncated domain are in the set C. All the source and receiver nodes are in B. The size of zone B is not limited in any direction. It is normally much larger than the size of truncated domain C. The remainder of the computational domain, including the boundaries, is in set A. The nodes enclosed by the dotted black box are used later in this paper to guide the derivation of the scattered field.

where supp() refers to the support of its argument. The perturbed model $m(x)$ is defined as:

$$m(x) = m_0(x) + \delta m(x), \tag{3}$$

where $m_0(x)$ is the background model, which is the initial guess or model in an inversion context. The model is the squared slowness. We define the scattered field as:

$$u_s(x) = u(x) - u_0(x), \tag{4}$$

where $u(x)$ is the wavefield in the perturbed model satisfying:

$$-m(x)\omega^2 u(x,\omega) - \Delta_h u(x,\omega) = f(x,\omega), \tag{5}$$

with Δ_h the discrete Laplacian with node spacing h. In this paper, a second-order accurate stencil is used. The wavefield in the background model u_0 satisfies:

$$-m_0(x)\omega^2 u_0(x,\omega) - \Delta_h u_0(x,\omega) = f(x,\omega). \tag{6}$$

2.1 Numerically exact boundary equation for the scattered field

An essential component of the method introduced in this paper is a numerically exact expression for the scattered field $u_s(x)$ as defined in (4) when model perturbations are restricted to C (i.e. the interior of the truncated domain). With this expression we will later truncate the computational domain and efficiently propagate the scattered field $u_s(x)$ to the receiver pixels. For a continuous problem, the expression for $u_s(x)$ is derived from a seismic reciprocity theorem in the form of a boundary integral around the truncated domain (Fokkema & van den Berg 2013). The boundary integral expresses the scattered field outside the truncated domain in terms of the perturbed field on its boundary and background Green's functions from the boundary to the point where the scattered field is evaluated. Directly discretizing this boundary integral would leave some open questions. Different choices for what to do at the boundary nodes and different normal derivative stencils would result in

different $u_s(x)$. Using a discrete derivation and applying summation by parts (Guenther & Lee 1988) we answer these questions naturally. By doing this the resulting boundary expression returns exactly the same $u_s(x)$ as the simple subtraction of wavefields $u(x)$ and $u_0(x)$ would in the discrete case. This numerical consistency is important because the ultimate goal is to obtain the same wavefield in the truncated domain and at the receivers as would have been obtained had we solved on the full domain. If in an inversion strategy a transition is made from a full domain solver to a local domain solver, without this accuracy numerical artefacts in the wavefield would cause inversion artefacts. The numerically exact boundary expression for the scattered field prevents such possible complications. We derive an expression for $u_s(x)$, involving nodes on the boundary of the truncated domain, for model perturbations restricted to C. Appendix A gives the derivation of the boundary expression for the scattered field. The resulting expression is:

$$u_s(y,\omega) = -\sum_{x\in\partial C}\frac{1}{h^2}\Big(u^{\partial C}(x,\omega)G_0^{\partial C+1}(x,y,\omega) - G_0^{\partial C}(x,y,\omega)u^{\partial C+1}(x,\omega)\Big), \quad y\in B\cup\partial C. \tag{7}$$

where $u^{\partial C}(x,\omega)$ and $u^{\partial C+1}(x,\omega)$ are the perturbed wavefield on boundary node x and one node to the interior, respectively. The background Green's functions between the node y where the scattered field is evaluated and the node x on the boundary, and one node to the interior are $G_0^{\partial C}(x,y,\omega)$ and $G_0^{\partial C+1}(x,y,\omega)$, respectively. The scattered field at any point $y\in B\cup\partial C$ is expressed in terms of a summation along the entire boundary ∂C, excluding its corner nodes. The wavefield $u^{\partial C}$ on the nodes ∂C is multiplied by the background Green's function from one node to the interior to the node y where the scattered field is computed. The wavefield $u^{\partial C+1}$ one layer to the interior of ∂C is multiplied by the background Green's function from one node to the interior of ∂C to the node y where the scattered field is computed. Eq. (7) is similar to a typical reciprocity theorem. Details of the derivation and interpretation can be found in Appendix A.

Eq. (7) is used to assemble a system of equations that solves for the exact wavefield within the truncated domain for the perturbed model m. This means that all the orders of scattering between the inhomogeneities in m_0 outside of the truncated domain and the perturbations δm inside the truncated domain are computed correctly.

2.2 Computing the perturbed wavefield in the truncated domain

In this section, we set up a system of equations to compute the perturbed wavefield $u(x,\omega)$ in the truncated domain, $B\cup\partial C$. To do this, we first introduce our node numbering scheme. Then we introduce three concepts that relate the perturbed wavefield in the truncated domain to the perturbed model $m(x)$ and the scattered wavefield $u_s(x,\omega)$. Using the concepts that were introduced earlier we setup the system of equations needed to compute the perturbed wavefield in the truncated domain. We then explain how the solution of this system of equations, which is the perturbed wavefield in the truncated domain, is used to efficiently obtain data residuals at the receivers. Using these data residuals at the receivers and the previously set up system of equations, the FWI gradient is calculated within the truncated domain.

The size of the full domain Ω is $n_x\times n_z$ and that of the truncated domain is $n'_x\times n'_z$. The number of boundary nodes n'_b in the truncated domain is, therefore:

$$n'_b = 2(n'_x + n'_z - 2). \tag{8}$$

The number of nodes in the truncated domain is n'_n:

$$n'_n = n'_x \times n'_z \tag{9}$$

A spiralling node numbering scheme is used in the truncated domain. The first node is the top left node on ∂C. Node numbering for the truncated domain then continues in a counter-clockwise inward spiralling fashion. This is illustrated in Fig. 1. The scattered field in the truncated domain is found from the following three relations:

List 1

(i) For the nodes on the boundary ∂C, the difference between the wavefield in the perturbed model u and the wavefield in the background model u_0 is the scattered wavefield u_s, that is, $u_s(x) = u(x) - u_0(x)$.

(ii) Also at each node on the boundary ∂C, the scattered wavefield $u_s(y,\omega)$ depends on the perturbed wavefield $u(x,\omega)$ at every node of ∂C, excluding the corner nodes. It also depends on $u^{\partial C+1}(x,\omega)$ one layer to the interior of ∂C. This relationship is expressed by eq. (7).

(iii) In the interior of the truncated domain, the Helmholtz equation in model $m(x)$ describes the relationship between the values of the wavefield at adjacent nodes.

Along with the node numbering scheme, the three relations in List 1 are used to assemble a system of equations:

$$\left(\begin{array}{c|c|c} -I & I & 0 \\ \hline I & G_0^{\partial C+1} & -G_0^{\partial C} \\ \hline \mathbf{0} & -\mathbf{m}(x)\omega^2 & -\Delta_h \end{array}\right)\left(\begin{array}{c} \mathbf{u_s}^{\partial C} \\ \mathbf{u}^{\partial C} \\ \mathbf{u}^{\partial C+1} \\ \mathbf{u}^{\partial C+2} \\ \vdots \end{array}\right) = \left(\begin{array}{c} \mathbf{u_0}^{\partial C} \\ \hline \mathbf{0} \\ \hline \mathbf{0} \\ \mathbf{0} \\ \vdots \end{array}\right), \tag{10}$$

where the two horizontal lines on the right-hand side separate the three relations. The first and the second relation both result in n'_b rows. The third relation, which is the interior Helmholtz equation, results in $(n'_n - n'_b)$ rows, which is the number of interior nodes. The quantities n'_b and n'_n are defined in eqs (8) and (9) respectively. The unknowns are ordered in a layered fashion because of the spiralling node numbering scheme. The vector $\mathbf{u_s}^{\partial C}$ represents the scattered field for all of the nodes on the boundary ∂C. Similarly, the vectors $\mathbf{u}^{\partial C}$ and $\mathbf{u}^{\partial C+1}$ represent the wavefield in the perturbed model m on ∂C and one layer to the interior, respectively. The superscript $\partial C+2$ represents the nodes two layers to the interior, etc. The matrix in eq. (10) is composed of block matrices. The block I is the identity matrix and the blocks $G_0^{\partial C}$ and $G_0^{\partial C+1}$ represent the background Green's functions, from ∂C and one layer to the interior, respectively, to the node in ∂C where the scattered field is evaluated. The block $G_0^{\partial C+1}$ contains the functions $G_0^{\partial C+1}(x,y,\omega)$ from eq. (7) and the block $G_0^{\partial C}$ contains the functions $G_0^{\partial C}(x,y,\omega)$, where y is now the boundary node where the scattered field is computed. These Green's function block matrices are the only dense blocks. They are several times larger than the size of the interior Helmholtz operator in computer memory. $G_0^{\partial C+1}$ is of shape n'_b by n'_b and $G_0^{\partial C}$ is n'_b by $(n'_b - 8)$.

The square matrix of size $n'_n + n'_b$ in eq. (10) is much smaller than the square full domain Helmholtz matrix of size $n_n = n_x\times n_z$. There are far fewer unknowns to solve for, which makes obtaining the truncated wavefield very fast. For the cases investigated in this

paper, the speedup factor is between 10 and 30, but the exact number depends entirely on the size of the truncated domain compared to the size of the full domain. More details on the cost and benefits can be found in the Discussion section.

The wavefield u in the truncated domain is obtained by solving the much smaller system of equations (i.e. eq. 10). This wavefield matches exactly what would have been generated by the full domain Helmholtz solver, as long as the model perturbations are restricted to the interior of the truncated domain. When we say exact, we mean that the relative difference between the two solutions is on the order of $\epsilon \sim 10^{-12}-10^{-15}$ when double precision is used, independent of the shape and amplitude of the model perturbation. We define relative difference as

$$\epsilon = \frac{|u_f - u_t|_2}{|u_f|_2}, \tag{11}$$

where u_f is the wavefield generated by a full domain solver, u_t is the wavefield from the truncated domain solver generated by solving eq. (10) and $|\,|_2$ is the least squares norm. This shows that the only difference is caused by rounding errors. All the orders of scattering between local perturbations in the model and inhomogeneities outside of the truncated domain are included.

We have now described the computation of the perturbed wavefield in the truncated domain. To compute a model update, either for standard FWI or for the salt updating scheme discussed in this paper, we need to compute the FWI gradient. To obtain the gradient in the truncated domain, the adjoint wavefield is required in addition to the perturbed wavefield (Virieux & Operto 2009). The adjoint wavefield requires evaluation of the data residuals in the perturbed model $m(x)$. Once the wavefield in the truncated domain has been solved for, eq. (7) is used to obtain the scattered field at the receivers. The cost of computing the scattered field at all the receiver locations is negligible compared to the cost of computing the truncated wavefield. This remains true for a streamer acquisition instead of the fixed spread acquisition investigated in this paper. The pre-computed background wavefield u_0 is added to u_s to obtain the perturbed wavefield u at the receiver locations, as defined in eq. (4). In this way the data residuals and the objective function for the locally perturbed model $m(x)$ can be evaluated exactly, using only local solves. With the numerically exact residuals available, the adjoint wavefield can be specified. For the least-squares misfit function

$$\chi = \frac{1}{2}\sum_{s,r}(u^{s,r} - d^{s,r})\overline{(u^{s,r} - d^{s,r})}, \tag{12}$$

the adjoint wavefield in the frequency domain for a specific shot s is defined as:

$$\beta^s(x,\omega) = \sum_{i=1}^{n_r}\overline{G(x,x_r,\omega)}\,(u^{s,r} - d^{s,r}), \tag{13}$$

where the bar is complex conjugation and n_r the number of receivers. The superscripts s and r are source and receiver indexes. The wavefield u in the truncated domain is already computed through eq. (10) and it is desirable to use the same system of equations to solve for the adjoint wavefield as well. This is achieved by realizing that the complex conjugate of eq. (13) is formed by letting the complex conjugate of the residuals act as source terms, just like standard FWI. The Green's functions that were used to propagate the scattered field from the truncated domain to the receivers are now used to propagate the contribution of each residual source to the boundary ∂C at no additional cost. Each residual source adds a contribution to the right-hand side term $\mathbf{u}_0^{\partial C}$ in eq. (10). Solving the system of equations now gives exactly the same complex conjugated adjoint wavefield as would have been generated by a full domain solver. Both the forward and the adjoint wavefields have now been efficiently computed in the truncated domain and are combined to get exactly the same FWI gradient as a full domain Helmholtz solver would have returned.

2.3 Full domain Green's function pre-computation

Before the small system of equations (eq. 10) can be solved at high speed, a set of full domain Green's functions must be computed. The background Green's functions that are required are:

List 2

(i) Between each source and its corresponding receivers.
(ii) Between each source and the nodes on ∂C. This gives $\mathbf{u}_0^{\partial C}$ in the right-hand side of eq. (10) when multiplied by the source wavelet.
(iii) Between all the node combinations on ∂C. This builds the block $G_0^{\partial C}$.
(iv) Between the nodes on ∂C and one layer to the interior. With this $G_0^{\partial C+1}$ is constructed.
(v) Between the receivers and ∂C and one layer to the interior. With these background Green's functions the scattered component of the solution of eq. (10) is propagated to the receiver locations and the adjoint wavefield is sent from the receivers to the truncated domain.

A greedy algorithm based on source–receiver reciprocity is used to compute the five sets of Green's functions listed above. For every node, a table is set up that keeps track of how many Green's function connections to other nodes are still left to be computed. The node with the most remaining connections is then used as the source in the next Green's function calculation until all pairs have been computed. The total number of full domain simulations required is favourable compared to what is required for normal full domain inversions. A cost analysis is given in the discussion session. Future studies should focus on reducing the required number of full domain solves to further improve efficiency.

2.4 Combining the local solver with the salt boundary inversion method

Estimating the shape of the salt body in a velocity model is labour intensive (Etgen *et al.* 2014) and generally involves the following steps. When an initial estimate is available, a migration is run on the velocity model. The focusing properties of the migrated image contain information about the quality of the salt body estimate. Interpreters use this information to alter the shape of the salt body in the velocity model. New migrations are run and this process is repeated iteratively until an adequate image is obtained. This process can take months because of the manual guidance that is required and the computational cost of migration. This illustrates the desirability of an automatic method for determining the correct salt geometry. Lewis *et al.* (2012) introduced a method to do this based on a level set approach. The method requires the FWI gradient at the boundary of the salt as input. Intuitively, the sign of the gradient tells whether the salt boundary should move outwards or inwards. In addition to the sign of the gradient, the salt updating code also uses the amplitude of the gradient to obtain a search direction for salt body updates. When performing a line search with this search direction, the method returns a new salt body shape for every attempted step

length. The salt shape resulting in the lowest value of the objective function is accepted and a new gradient and a new search direction are computed. By iterating over this procedure the salt body estimate is expected to converge towards the true shape. Appendix B explains the salt boundary inversion method in more detail.

The salt boundary inversion method requires the FWI gradient only at the boundary of the salt, which makes using a local solver desirable. Instead of computing the full domain gradient and discarding the gradient everywhere except for the salt boundary, we now have the option to compute the gradient exactly and efficiently for this desired region with the local solver. For the truncated domains investigated in this paper, the speedup factor is between 10 and 30. Larger truncated domains have a smaller speedup factor, while smaller truncated domains have higher speedup factors. The speedup from the local solver makes the inversion much more interactive. For example, instead of having to wait six days, the inversion finishes in approximately 7 hr on a single core. The short feedback time of the local inversions and the reduced computational resource requirements make it possible to investigate many more scenarios. For instance, we can try many different sets of inversion parameters and also different initial salt shapes to investigate the robustness of the inversion.

3 RESULTS

In this section, we apply the local solver to the automatic salt updating scheme. The complexity of the salt body inversion is raised incrementally to show its characteristics. A finite difference code is used to generate the data and the background Green's functions. No noise is added to the data.

3.1 Verification of the local solver

Before starting the experiments we first verify the numerical exactness of the local solver. This is tested by generating wavefields on the extended Pluto velocity model in Fig. 2, which is perturbed from the true model in region C indicated by the black box. During a typical local inversion, the model within the truncated domain will change from the initial model on which the Green's functions were generated. To simulate this, we investigate the general case where the model $m(x)$ in Fig. 2 differs within the truncated domain from the initial model $m_0(x)$ on which the Green's functions are generated. The perturbed wavefield in the truncated domain $u(x, \omega)$ is obtained by solving eq. (10). Because of the non-zero model perturbation, the solution vector contains a non-zero scattered field vector $\mathbf{u}_s{}^{\partial C}$.

Perfectly Matched Layers (PMLs) are placed on all four boundaries of Fig. 2, although this is not required by the local solver. The exactness of the local solver would continue to hold with free surface or other boundary conditions on any of the boundaries. Fig. 3(a) shows the real part of the wavefield generated on the velocity model of Fig. 2 using a full domain Helmholtz solver. Fig. 3(b) enlarges the wavefield within the black box, which represents the truncated domain. Fig. 3(c) shows the result obtained using the local solver. The wavefields within the truncated region match exactly. The relative difference as defined by eq. (11) is $\epsilon \sim 10^{-12}$–10^{-16}. Even though the model is perturbed from the initial model within the truncated domain, the local solver still matches the full domain wave solve exactly. This means that all of the interactions between the locally perturbed model and the inhomogeneous, but unperturbed, exterior are correctly modelled.

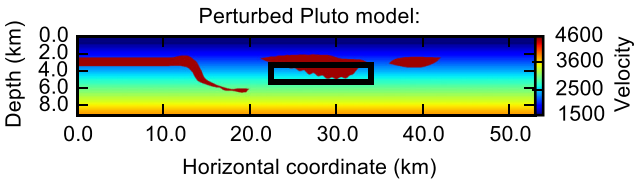

Figure 2. The 3480 × 600 nodes velocity model on which the wavefields are generated. The 758 × 133 nodes truncated domain is indicated by the box. Within this box the model is perturbed from the initial model on which the Green's functions are generated, and the wavefields are then compared in this domain.

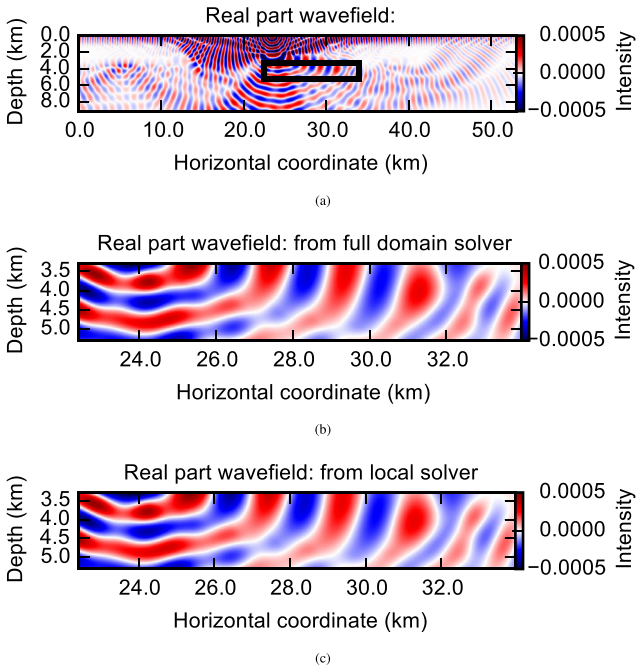

Figure 3. Verification of the local solver using the Pluto model. (a) Real part of the wavefield from the full domain Helmholtz solver. (b) The result from the full domain solver is enlarged within the black box. (c) Real part of the wavefield in the black box from the truncated solver. An equally good fit is obtained for the imaginary part.

The same accuracy levels are maintained when propagating the solved local wavefield to the receiver locations. Therefore, data residuals are the same as a full domain solver would have obtained up to numerical accuracy. These residuals form sources for the adjoint wavefield. Similar tests show that the adjoint wavefield generated by the local solver also matches the full domain equivalent with a relative difference $\epsilon \sim 10^{-12}$–10^{-16}. This shows that the local solver introduces no artefacts in the forward wavefield, objective function evaluation and adjoint wavefield. Therefore, all the components for a local inversion are available, but now at increased speed.

3.2 Application to salt boundary inversion

When the initial model differs from the true model only within the truncated domain, a very good reconstruction is expected for the noiseless case. We investigate this scenario in the extended Pluto model. The initial model is equal to the true model everywhere except for the black box in Fig. 2. A fixed spread acquisition geometry is simulated with 200 equally spaced sources and 500 equally spaced receivers 40 m below the surface. We perform a frequency

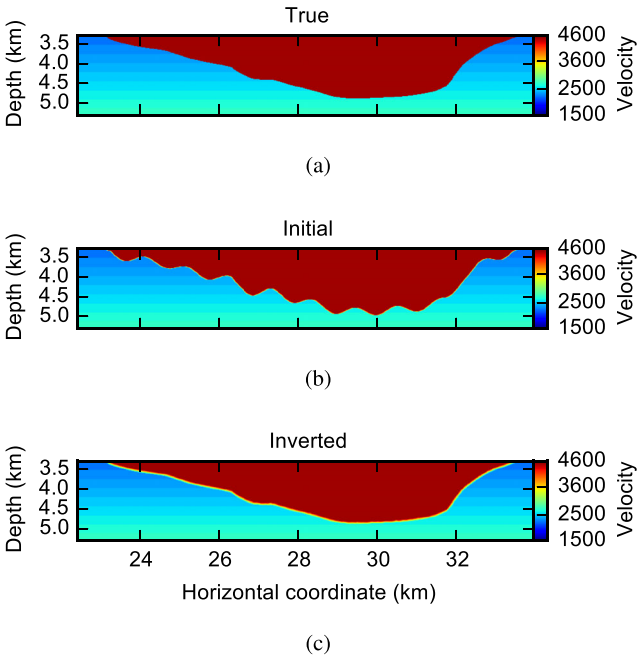

Figure 4. Salt boundary inversion results for the Pluto model where the true and initial models only differ within the interior of the truncated domain. (a) True truncated domain. (b) Initial truncated domain. (c) Inverted truncated domain.

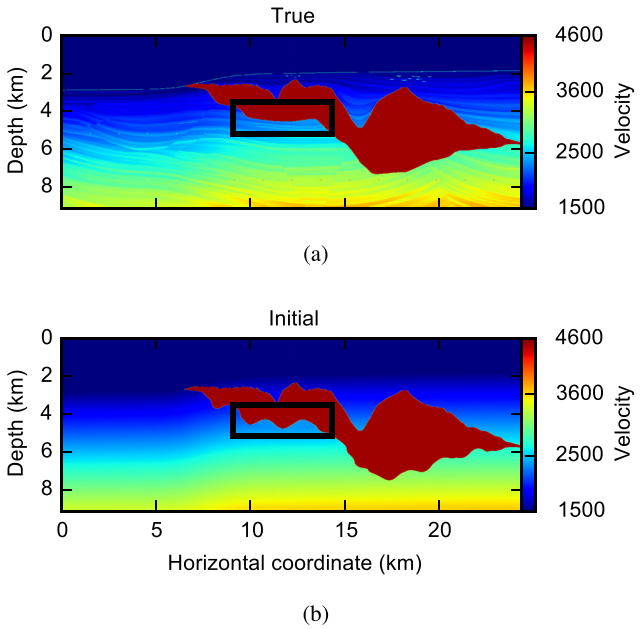

Figure 5. (a) True 3201 × 1201 nodes Sigsbee2b velocity model with a 691 × 221 nodes truncated domain. (b) Initial velocity model with MVA sediment and an incorrect salt bottom everywhere, not just inside the truncated domain indicated by the black box. The top of the salt is correct in this example.

sweep with inversion taking place at the frequencies 2.5, 3.0, 3.5, 4.5, 5.5, 7.0, 8.5 and 10.0 Hz. We use the same inversion frequencies in the other inversion experiments in this paper. For the Pluto example we perform five inversion iterations at each frequency. The FWI gradient is generated within the truncated domain, and this gradient is used by the salt updating code to generate a search direction. For each step length in the line search a different salt geometry is suggested. A very simple line search is used in the inversion in order to minimize the number of evaluations of the objective function. At the start of each inversion frequency, a step length on the higher end of the scale is used. If this results in an increase of the objective function, the step length is cut back drastically for the remaining iterations at that frequency. In this way the number of Helmholtz solutions in most iterations is only three times the number of shots. We do this because generating the gradient costs two times the number of shots; one forward simulation and one adjoint simulation for each shot. Only one step length normally has to be evaluated, at a cost proportional to the number of shots.

The true local velocity is displayed in Fig. 4(a). Note that the transition between the salt and sediment happens in a staircase manner. The salt updating code will only return models where the transition is a velocity gradient. Therefore, the inversion cannot give a perfect reconstruction of the true velocity model with the staircase transition. The initial velocity model in Fig. 4(b) contains an oscillatory perturbation of the salt boundary with an amplitude exceeding 100 m. No updates are performed on the boundary nodes. The local solver only returns the correct wavefield when model updates are limited to the interior nodes, so we set the initial velocity equal to the true velocity on the boundary of the truncated domain. The inversion result obtained after performing the frequency sweep is displayed in Fig. 4(c). Because the differences between the true and initial models are restricted to the interior of the truncated domain, a very good inversion result is obtained. The only difference is that the inverted result has a thin 'halo' around the boundary of the salt instead of the pixelated staircase pattern of the true model.

In a realistic inversion, the initial model will differ from the true model everywhere, and not just in the truncated domain. It is interesting to investigate if the local solver, in combination with the salt updating code, can enhance the salt boundary estimate locally, despite the presence of inaccuracies outside of the truncated domain. To investigate this we use the Sigsbee2b velocity model. The true velocity model is shown in Fig. 5(a). The initial velocity model in Fig. 5(b) has the true sediment velocity replaced by the MVA sediment velocity. The top of the salt is correct, but the bottom of the salt is incorrect everywhere and not just within the truncated domain indicated by the black box. Just as in the Pluto model, 200 shots are used, but instead of using all 500 equally spaced receiver locations, only those within a 10 km offset range are used.

We perform three iterations at each inverted frequency. Fig. 6 shows the true, initial and inverted models for this scenario. An increased number of inversion iterations does not substantially change the resulting velocity model. Unlike the Pluto test, the initial velocity model now has an incorrect sediment velocity and salt bottom perturbations with amplitude larger than 100 m outside of the truncated domain. Despite these inaccuracies in the initial model, inversion in the truncated domain still yields a more accurate local velocity model. A thin halo-shaped feature is present on the inverted model, which is an artefact of the current implementation of the salt updating code. Because the salt boundary is only updated in the truncated domain, a model discontinuity may appear when the salt moves along the truncation boundary. This can be seen in Fig. 7, where the truncated inverted model is embedded in the unchanged full domain initial model. Because of this it is desirable to let the salt boundary intersect the domain truncation boundary on the sides. This causes a vertical discontinuity that has a smaller effect on the data than a horizontal discontinuity would. A horizontal reflector would have been generated had the salt cut the bottom or top of the

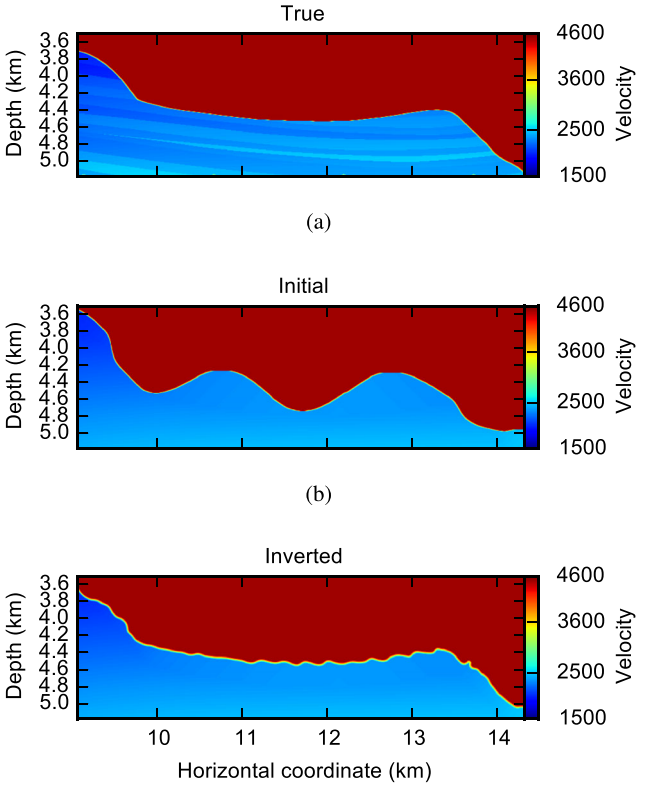

Figure 6. (a) True Sigsbee2b truncated velocity model from Fig. 5(a). (b) Initial truncated model from Fig. 5(b). (c) Inverted truncated model from local solver. Significant improvement can be observed.

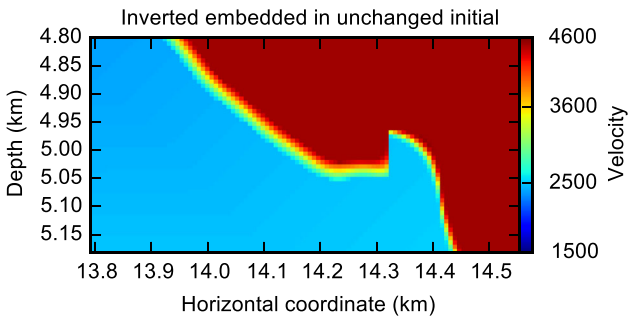

Figure 7. Truncated domain inversion result embedded in the unchanged full domain initial model illustrating that abrupt changes can take place at the boundary of the truncated domain.

truncated domain. In practice, the salt boundary should be tapered to avoid the discontinuity that is observed in Fig. 7. This can be achieved naturally in the framework of the salt updating code by increasing the surface tension near the boundaries of the truncated domain. The current implementation of the code did not yet allow for spatially varying surface tension.

An improved salt bottom estimate has beneficial effects for subsalt migration. Fig. 8 shows sections of the migrated images in the region below the truncated domain on the true (a), initial (b) and inverted (c) velocity models, respectively. The migration in the initial velocity model severely distorts the migrated image. It is very hard to find any of the features that are present in the true model. The truncated domain inversion improves the salt bottom estimate locally, which improves the image so that many of the features in

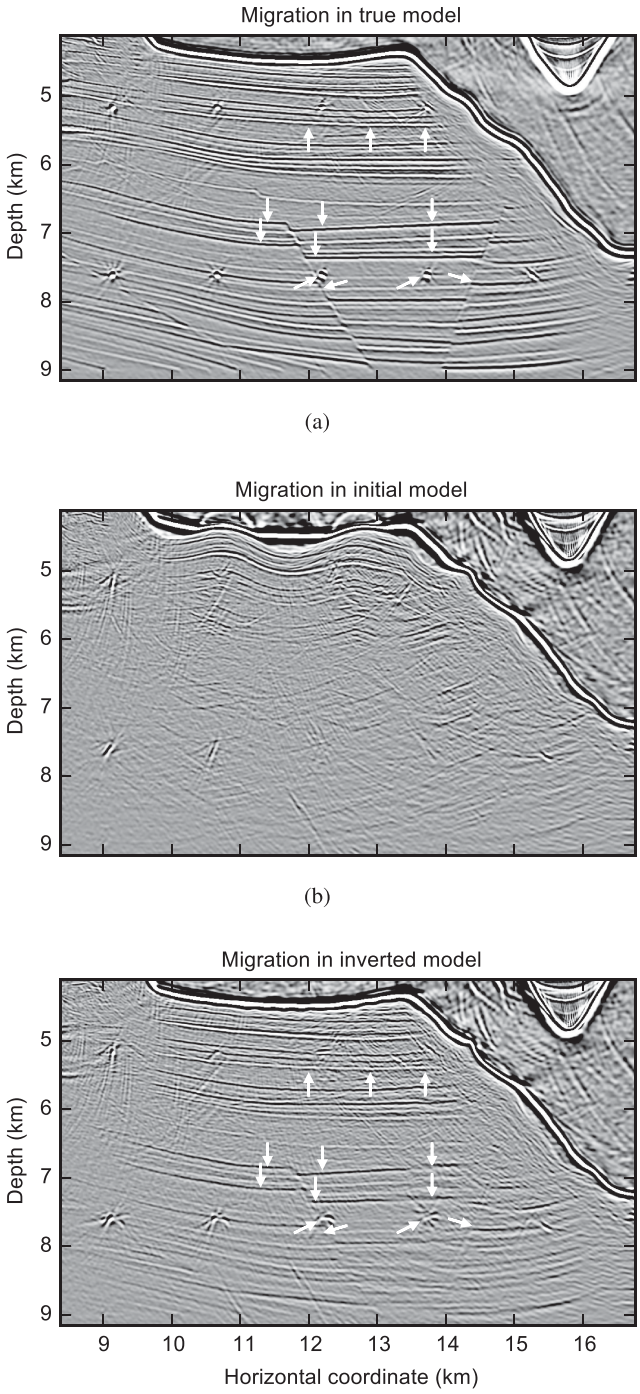

Figure 8. Migration in the true, initial and inverted models of Fig. 6. The initial model uses an MVA sediment velocity and has a very incorrect salt bottom. The arrows show the locations of interesting features that appear after the salt bottom estimate has been improved.

the true model can now be seen. The inverted image shows the horst-graben structure and most of the deep point diffractors.

3.3 Multidomain inversion

In the previous section, the assumption was made that the top of the Sigsbee2b salt body was pixel-perfect in the velocity model. Even

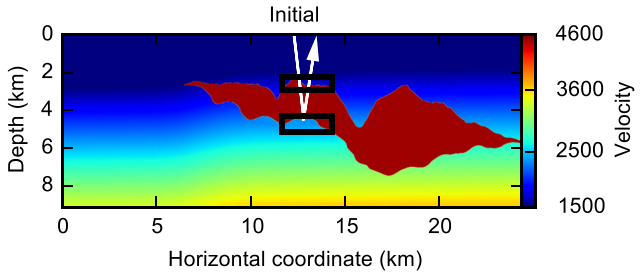

Figure 9. The initial model for the multidomain inversion. The top truncated domain is 347 × 91 nodes and the bottom domain is 347 × 111 nodes. The true model remains the same, shown in Fig. 5. The white ray shows that a primary reflection on the salt bottom is influenced by the top domain twice. This demonstrates that it is important that the local solver correctly accounts for waves re-entering the truncated domain. Waves reflecting multiple times benefit from this even more.

though it is much easier to estimate the top of the salt, there will always be slight inaccuracies. These inaccuracies make accurate reconstruction of the bottom salt a difficult task. Inaccuracies in both the top and the bottom will distort the subsalt image. In a scenario where the migrated image needs to be enhanced in a subsalt region, it is desirable to get a good approximation of the salt body above it. For this purpose, we derive a multidomain extension of eq. (10). Details can be found in Appendix C. This multidomain solver generates the identical wavefield as a full domain solver would have in multiple disjoint regions, as long as the model is only updated within those regions. There are no restrictions on the number of domains or on their shape.

Using this multidomain solver, both the top and the bottom of the salt are inverted in order to create a 'window' through the salt. The computational cost of the multidomain solver is slightly better than the cost of a single domain with the same number of boundary nodes. The reason for this is that for a fixed number of boundary nodes the single domain contains more nodes and, therefore, more unknowns. For this experiment the bottom is perturbed in a similar way to the previous example. In addition, the top is also perturbed, but less than the bottom to represent that it is easier to obtain the correct top of salt than it is to obtain the correct bottom of salt.

Fig. 9 shows the initial geometry with the black boxes indicating the two simultaneous inversion domains. Individually, both domains are slightly smaller than the single domain in Fig. 5(b), so that the multidomain local solver has a similar speedup factor over the full domain Helmholtz solver as did the single domain local solver in Fig. 5(b). The two windows are moved eastwards by a small distance so that they have a relatively large depth separation. In this case, the multidomain inversion has a significant advantage over using a big single domain enclosing both boxes on the top and the bottom. Such a big single domain would solve for many unnecessary unknowns, such as the wavefield in the interior of the salt, where no model updates take place.

We perform ten iterations at each inverted frequency. Inversion takes place simultaneously at the top and the bottom. Rather good inversion results are obtained after only five iterations, with slight improvements continually accruing up to ten iterations in this noiseless investigation. Differences in illumination between the top and bottom of the salt can be compensated by applying pre-conditioning to the gradient (Shin *et al.* 2001). The modified gradient has similar amplitude above and below the salt, promoting salt boundary updates of similar magnitudes. The improvement in the top domain of Fig. 9 is seen in Fig. 10. Just as in the previous inversions, the

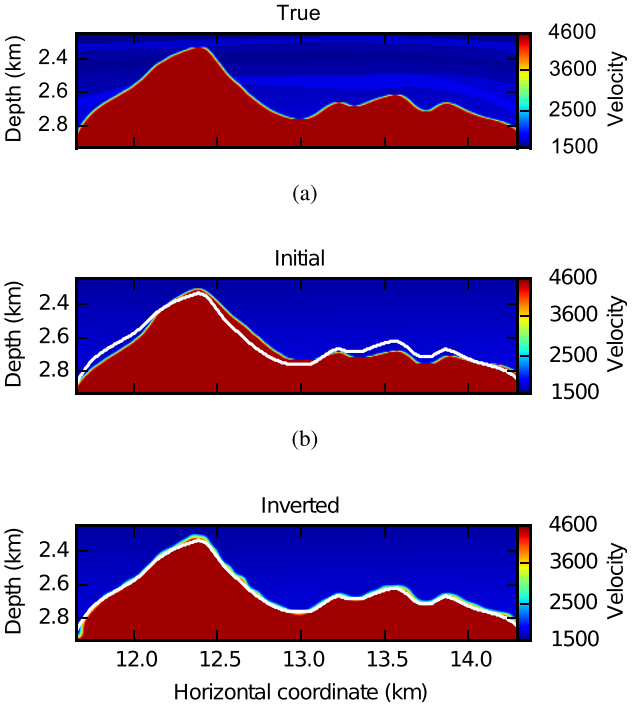

Figure 10. This figure shows the improvement in the top salt estimate. Inversion takes place simultaneously in both of the truncated domains in Fig. 9. The initial and true models differ significantly everywhere. The white line shows the true salt boundary.

halo-like feature around the inverted salt body is slightly too wide. Reinitializing the implicit function as it evolves periodically may alleviate this problem. Despite this artefact, the initial model has improved significantly. The inverted topography on the right half is much closer to the true topography. With the exception of the left corner, the salt flank on the left has also improved significantly.

The results of inversion in the bottom domain, which are computed simultaneously with the inversion in the top domain, are shown in Fig. 11. The bottom domain initially has larger perturbations of the salt boundary than the top domain. The multidomain inversion improves the salt bottom estimate significantly even though the top of salt starts out incorrect everywhere, and is only improved right above the bottom of salt. The interior of the bottom domain sees the greatest improvement, although the sides also move in the correct direction.

Many waves corresponding to long offsets travel through the incorrect top of salt outside of the top window before reaching the bottom truncated domain. Especially at higher frequencies, their contributions to the gradient may be out of phase and will not add constructively. The narrow offsets for sources above the domains will travel through the top domain before reaching the bottom domain. Therefore, they will benefit more from the improved top of salt estimate and are more likely to add constructively to the gradient in the bottom domain. It is worth mentioning that the exact nature of the local solver is very important for this specific problem. Imagine a wave travelling from a source above the domains to the bottom domain and then finally to a receiver above the domains. This wave path is represented by the ray in Fig. 9. The wave will first enter the top domain where it experiences the updated velocity model. It then moves down to the bottom domain, where it is reflected. It then re-enters the top domain before it moves to the

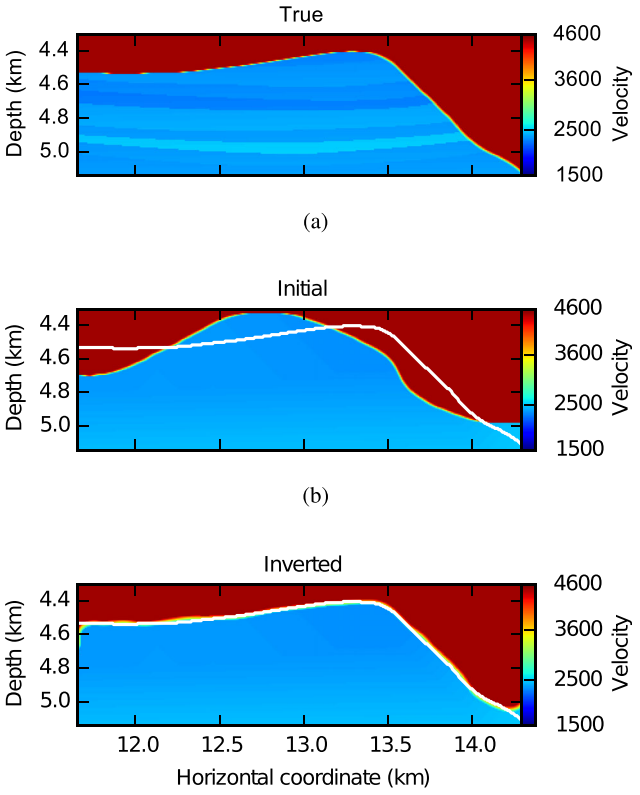

Figure 11. This figure shows the improvement in the bottom salt estimate. Inversion takes place simultaneously in both of the truncated domains in Fig. 9. The initial and true models differ significantly everywhere. The white line shows the true salt boundary.

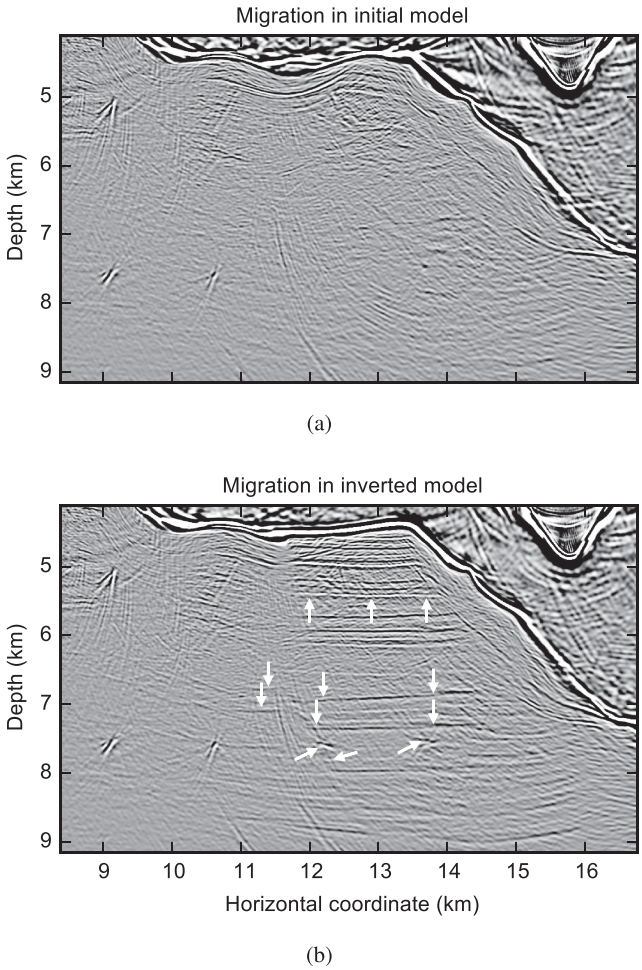

Figure 12. Migration artefacts around the salt bottom in the initial model still give an indication of the correct salt shape. Less distorted initial models have been investigated where the correct salt bottom was not recognizable from migrated images, but simultaneous inversion still improved the salt bottom estimate. The large salt perturbations in Figs 10(b) and 11(b) make it easier to visualize improvements in the velocity model and the migrated image. Arrows show interesting features. The location of these arrows is slightly perturbed from those in the true model in Fig. 8. But they show the same geological features.

receiver. This primary reflection on the bottom of salt is influenced by model updates in the top of salt twice. Because no assumptions are made, these multiple interactions between the waves with the perturbations in the top domain are handled correctly. Waves can enter and exit any perturbed domain an arbitrary number of times. Multiples between the top and bottom of salt are influenced by perturbations as they pass through the salt even more times, and the local solver still handles these waves correctly.

Figs 12(a) and (b) show migration with the initial and the inverted velocity models. Because now the top of salt also starts out incorrectly and the truncated domains are narrower than in the 'bottom only' scenario, the migrated image is less accurate. There is still great improvement, with sections of the layers and the graben structure becoming visible because of the window that is created in the salt by improving the salt boundary on the top and the bottom simultaneously.

Instead of using the multidomain implementation introduced in this section, we can also elect a sequential inversion approach. In this approach, we first generate the Green's functions for the top truncated domain in Fig. 9. After using this single domain local solver to update the top salt, we would then generate the Green's functions for the bottom domain in Fig. 9. The bottom salt is then updated, changing the arrival time of the bottom salt primary reflection in Fig. 9. If we would want to invert for the top salt again, now that the bottom salt primary reflection has changed significantly, we would have to recompute the Green's functions used by the top single domain solver. The reason for this is that the salt bottom update changed the model outside of the top truncated domain. Generating wavefields using the single domain solver is slightly faster, because the matrix is smaller. But the advantage of the multidomain solver is that all the offline work of computing the Green's functions can be done once, before the inversion starts. All orders of multiple interactions between model changes in the different subdomains can also be computed naturally without having to recompute Green's functions each time a subdomain is changed.

4 DISCUSSION

Local inversion may not converge towards the true truncated model, even when the data are noiseless. To see this, assume that the initial model differs from the true model outside of the truncated domain. The model space is constrained to the truncated domain. Reconstruction of the true model within the truncated domain may not coincide with a minimum of the objective function. Interactions between the inverted true truncated model and the incorrect exterior will leave an imprint on the synthetic data that cannot be removed

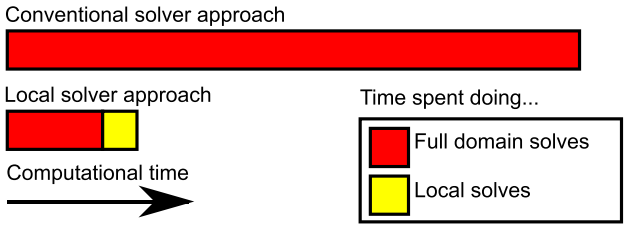

Figure 13. Schematic cost comparison for running five simultaneous inversions using different inversion parameters and initial conditions. The length of the bars is to scale.

with local updates. Other truncated models may match the noiseless data better. Despite this problem, the local inversion experiments on the Sigsbee2b velocity model show significant improvement.

All the inversion results shown started from an initial model that had oscillatory perturbations. The reason why oscillatory perturbations were chosen is that they are easy to visualize in plots. We have investigated other perturbations, such as uniform depth errors in the range of 100 m, and similar results were obtained. As with any least-squares inversion, there is a limit on how far the initial model can differ from the true model, as the inversion can get stuck in a local minimum. There are objective functions that avoid this problem, for example, those based on cross-correlation (Diaz & Sava 2013).

The motivation for the combination of the local solver with the salt updating code is that inverting the salt shape currently requires much manual work. In some of the examples we tested, the salt shape of the initial model was so far from the correct model that the migrated images contained clear hints of the correct salt shape, as can be seen in Fig. 8(b). The migrated image shows semi-circular artefacts on the salt bottom. The perfect top salt in Fig. 5 would realistically result in a rather accurate initial estimate for the salt bottom through salt-flooding. The reason why an unreasonably bad initial model was chosen was to demonstrate the inversion capabilities of the local salt inversion strategy. The initial model for multidomain inversion in Fig. 9 is much more realistic. The incorrect top of salt makes it harder to estimate the initial salt bottom. The migrated image in the initial velocity model in Fig. 12 still contains hints about the correct salt bottom because the error is still large. However, salt flooding would not have resulted in a close to perfect salt-bottom estimate in this case. We have also performed multidomain inversion on initial models with smaller bottom of salt errors. In that scenario, the migrated image in the incorrect velocity model did not show clear hints about the true salt bottom shape. The multidomain inversion still improved the salt body estimate and the subsalt image.

4.1 Cost and benefit

The Green's functions that must be pre-computed are described in the Theory section. In both of the Sigsbee2b tests, the number of Green's functions that need to be computed at each frequency is approximately 2500, when using reciprocity and a greedy algorithm. These pre-computations require an equal number (2500) of full domain Helmholtz solves. The pre-computations for all the frequencies can be performed before the frequency sweep starts.

Fig. 13 schematically compares the work of the local solver with that of using a full domain solver to generate the gradients in the regions of interest. The comparison is made for a single frequency, but the use of multiple frequencies would simply result in a multiplier on both bars. The pre-computation for multiple frequencies can be done before the frequency sweep starts. The figure compares the cost of running five inversions with different inversion parameters and initial conditions. Each inversion performs five iterations, and each iteration requires the computation of the gradient and the evaluation of the objective function for at least one step length. Generating the gradient requires two times the number of shots (2 times 200), with the cost split over forward and adjoint simulations. Evaluating the objective function in the line search requires another 200 simulations. If all this work is done using a full domain solver, the total cost would be $5 \times 5 \times 600 = 15\,000$ full domain simulations, which somewhat optimistically assumes that only one objective value had to be evaluated for each line search. The computational cost of these 15 000 full domain solves is represented by the large red bar for this conventional approach. The truncated domain solvers require approximately 2500 full domain simulations to pre-compute the Green's functions, represented by the smaller red bar. No full domain simulations are required after the initial setup.

For the Pluto model the local solver is more than 10 times faster than the full domain solver. The local solves in both the single domain and the multidomain Sigsbee2b scenarios are slightly less than 30 times faster than using a full domain solver. The difference in speedup factors is caused by differences in the relative size of the truncated domain(s) and the full domain. In the comparison in Fig. 13, an intermediate speedup factor of 20 is used. Just like the full domain analysis, 15 000 simulations are performed. But these are now much faster. This is represented by the shorter yellow bar in Fig. 13. In addition to the decrease in total computation, the local solver approach has the advantage that the time between the start and the end of the inversion is 10–30 times shorter. The memory requirements are also substantially lower because of the reduced model space, allowing for more simultaneous inversions on the same hardware. The full domain Helmholtz equation includes PMLs on the sides. It is solved using the direct solver package UMFPACK (Davis 2004), which is feasible in 2-D. The single truncated domain system of equations (eq. 10) and the multidomain version (C2) are also solved with UMFPACK. The memory required for factorizing the local solver matrix (10) is much smaller than for the full domain Helmholtz matrix. This could potentially enable the use of direct solvers in scenarios where they would otherwise fail due to lack of memory. The advantage of a direct solver is that the computational cost of computing the wavefield associated with each source is very small after the factors have been computed.

Similar to the local solver introduced in this paper, the time domain local solver of van Manen *et al.* (2007) also computes the local wavefield exactly. But the computational complexity of the domain truncating boundary condition is less favourable in their implementation. Similar to the frequency domain implementation introduced in this paper, the wavefield at each boundary node depends on the wavefield at all the other n_b' boundary nodes. Both methods therefore scale quadratically with the number of boundary nodes. The major difference between the two methods is that the time domain method also scales quadratically with the number of time steps n_t. The reason for this scaling is that at each time step the results of all the previous time steps are required. This scaling can make the time domain local solver inefficient, because the computational cost of a full domain solver depends only linearly on the number of time steps. The frequency domain implementation introduced in this paper depends only linearly on the number of frequencies n_f used in the inversion, where n_f is usually also much smaller than n_t. The usual difference between time and frequency

domain solvers applies to the computation of the wavefield in the interior of the truncated domain.

The truncated domains in this study have been rectangular, but this is by no means necessary. In fact, the domains can have arbitrary shape. The local solver could have been made even faster for this salt updating application had the salt boundary been followed more closely, reducing the number of unknowns. Figs 6 and 11 show that a substantial volume of sediment and salt interior are included in the truncated domain, while the salt boundary never moved to many of those locations during the inversion, so the wavefields and gradients were unnecessarily computed there.

For the experiments in this paper a fixed grid spacing is used during the frequency sweep. This means that an unnecessarily high number of grid points per wavelength is used at low frequencies. If the grid density is decreased at lower frequencies, the number of Green's functions that must be pre-computed decreases as well. This reduces the setup cost of the local solver and makes it more favourable.

4.2 Future work

There are three clear avenues of future work. First, the local solver introduced in this paper generates the local wavefields at increased speed, but before the local solver can be used, a large number of Green's functions must be computed. This requires many costly full domain simulations. This number can be reduced if the Green's functions between two points could be estimated with high accuracy. Approximations based on traveltimes or interferometric approximations could, for instance, be used to construct a dictionary of basis matrices for the Green's function blocks in eq. (10). The idea of matrix probing (Chiu & Demanet 2012) could then be used to fit the Green's function blocks with the dictionary of basis matrices. A similar matrix-probing approach has previously been used to approximate the effects of perfectly matched layers (Bélanger-Rioux & Demanet 2015). Such an innovation will be required when applying the local solver to 3-D problems. The number of boundary nodes will grow drastically, as will the number of required Green's functions. Second, the derivation of the local solver was based on the second-order accurate Helmholtz equation. Numerical dispersion motivates adaptation of the algorithm to other discretization schemes. Finally, it will be interesting to apply the local solver to problems other than salt boundary updating.

5 CONCLUSIONS

In this paper, we introduced a numerically exact local solver. We demonstrated that the local solver generates exactly the same wavefield as a full domain solver in a computationally efficient manner. This property is preserved when the model is perturbed in the interior of the truncated domain. All interactions between model perturbations in the interior of the truncated domain and the exterior are included automatically. The local solver computes the local wavefield by solving a relatively small system of equations. The small size of the matrix before and after factorization may allow for a direct solver in situations where the full domain cannot be factorized because of the size of the factors. We combined the local solver with an automatic salt updating code. The locally improved salt boundary estimate resulted in significant improvements in the subsalt migrated image.

ACKNOWLEDGEMENTS

We acknowledge Marlies Vasmel from ETH Zurich and Laurent Demanet from MIT for insightful discussions. We are thankful to Schlumberger for giving permission to publish this research. In addition, we want to thank Johan Robertsson and Paul Williamson for their careful review and helpful suggestions. Finally, we would also like to thank MIT and the ERL consortium members for funding and the support of the National Science Foundation under Grant No. DMS-1115406. The Python Seismic Inversion Toolbox (PySIT) was used for the numerical experiments in this paper.

REFERENCES

Arts, R., Trani, M., Chadwick, R., Eiken, O., Dortland, S. & van der Meer, L., 2009. Acoustic and elastic modeling of seismic time-lapse data from the Sleipner CO_2 storage operation, in *Carbon Dioxide Sequestration in Geological Media—State of the Science: AAPG Studies in Geology,* vol. 59, pp. 391–403, eds Grobe, M., Pashin, J.C. & Dodge, R.L., American Association of Petroleum Geologists.

Bansal, R. *et al.*, 2013. Simultaneous-source full-wavefield inversion, *Leading Edge,* **32**(9), 1100–1108.

Berryhill, J., 1984. Wave-equation datuming before stack, *Geophysics,* **49,** 2064–2066.

Bélanger-Rioux, R. & Demanet, L., 2015. Compressed absorbing boundary conditions via matrix probing, *SIAM J. Numer. Anal.,* **53**(5), 2441–2471.

Bevc, D., 1995. Imaging under rugged topography and complex velocity structure, *PhD thesis,* Stanford University, Stanford, California, USA.

Brandsberg-Dahl, S., de Hoop, M.V. & Ursin, B., 1999. Velocity analysis in the common scattering-angle/azimuth domain, *SEG Technical Program Expanded Abstracts 1999,* pp. 1715–1718, doi:10.1190/1.1820865.

Broggini, F., Snieder, R. & Wapenaar, K., 2014. Data-driven wavefield focusing and imaging with multidimensional deconvolution: numerical examples for reflection data with internal multiples, *Geophysics,* **79**(3), WA107–WA115.

Byun, J., Yu, J. & Seol, S., 2010. Crosswell monitoring using virtual sources and horizontal wells, *Geophysics,* **75**(3), SA37–SA43.

Chiu, J. & Demanet, J., 2012. Matrix probing and its conditioning, *SIAM J. Numer. Anal.,* **50,** 171–193.

Davis, T., 2004. Algorithm 832: Umfpack v4.3 - an unsymmetric-pattern multifrontal method, *ACM Trans. Math. Softw.,* **30,** 196–199.

Diaz, E. & Sava, P., 2013. Data-domain and image-domain wavefield tomography, *Leading Edge,* **32**(9), 1064–1071.

Dix, C., 1955. Seismic velocities from surface measurements, *Geophysics,* **20**(1), 68–86.

Dong, S., Luo, Y., Xiao, X., Chvez-Prez, S. & Schuster, G., 2009. Fast 3D target-oriented reverse-time datuming, *Geophysics,* **74**(6), WCA141–WCA151.

Dorn, O., Miller, E. & Rappaport, C., 2000. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets, *Inverse Probl.,* **16,** 1119–1156.

Etgen, J.T., Chu, C., Yang, T. & Vyas, M., 2014. Adaptive image focusing, *SEG Technical Program Expanded Abstracts 2014,* pp. 3774–3778.

Fedkiw, R.P., Aslam, T., Merriman, B. & Osher, S., 1999. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.,* **152,** 457–492.

Fichtner, A., 2010. *Full Seismic Waveform Modelling and Inversion,* Advances in Geophysical and Environmental Mechanics and Mathematics, Springer.

Fokkema, J. & van den Berg, P., 2013. *Seismic Applications of Acoustic Reciprocity,* Elsevier Science.

Gill, P.E., Murray, W. & Wright, M.H., 1981. *Practical Optimization,* Academic Press.

Gillman, A., Barnett, A. & Martinsson, P.-G., 2013. A spectrally accurate direct solution technique for frequency-domain scattering problems with variable media, *BIT Numer. Math.,* **55,** 141–170.

Grote, M. & Sim, I., 2011. Local nonreflecting boundary condition for time-dependent multiple scattering, *J. Comput. Phys.,* **230,** 3135–3154.
Guenther, R. & Lee, J., 1988. *Partial Differential Equations of Mathematical Physics and Integral Equations,* Dover Publications, Inc..
Haffinger, P., 2012. Seismic broadband full waveform inversion by shot/receiver refocusing, *PhD thesis,* TU Delft, Delft, The Netherlands.
Herrmann, F., Hanlon, I., van Leeuwen, T., Li, X., Smithyman, B. & Wason, H., 2013. Frugal full-waveform inversion: from theory to a practical algorithm, *Leading Edge,* **32**(9), doi:10.1190/tle32091082.1.
Jugnon, V. & Demanet, L., 2013. Interferometric inversion: a robust approach to linear inverse problems, *SEG Technical Program Expanded Abstracts 2013,* pp. 5180–5184.
Keys, R. & Weglein, A., 1983. Generalized linear inversion and the first born theory for acoustic media, *J. Math. Phys.,* **24,** 1444–1449.
Krebs, J., Anderson, J., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A. & Lacasse, M., 2009. Fast full-wavefield seismic inversion using encoded sources, *Geophysics,* **74**(6), WCC177–WCC188.
Lailly, P., 1983. The seismic inverse problem as a sequence of before-stack migrations, in *Conference Proceeding on Inverse Scattering: Theory and Application,* vol. 11, SIAM, Philadelphia, pp. 206–220.
Lewis, W., Starr, B. & Vigh, D., 2012. A level set approach to salt geometry inversion in full-waveform inversion, *SEG Technical Program Expanded Abstracts 2012,* pp. 1–5.
Masson, Y., Cupillard, P., Capdeville, Y. & Romanowicz, B., 2013. On the numerical implementation of time-reversal mirrors for tomographic imaging, *Geophys. J. Int.,* **196,** 1580–1599.
Métivier, L., Brossier, R., Virieux, J. & Operto, S., 2013. Full waveform inversion and the truncated newton method, *SIAM J. Sci. Comput.,* **35**(2), B401–B437.
Mulder, W., 2005. Rigorous redatuming, *Geophys. J. Int.,* **161,** 401–415.
Osher, S. & Fedkiw, R., 2003. *Level Set Methods and Dynamic Implicit Surfaces (Applied Mathematical Sciences),* Springer.
Osher, S. & Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.,* **79**(1), 12–49.
Plessix, R., Baeten, G., de Maag, J.W., Klaassen, M., Rujie, Z. & Zhifei, T., 2010. Application of acoustic full waveform inversion to a low-frequency large-offset land data set, *SEG Technical Program Expanded Abstracts 2010,* pp. 930–934.
Pratt, R.G., 1999. Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model, *Geophysics,* **64**(3), 888–901.
Prieux, V., Brossier, R., Operto, S. & Virieux, J., 2013. Multiparameter full waveform inversion of multicomponent ocean-bottom-cable data from the Valhall field. Part 1: Imaging compressional wave speed, density and attenuation, *Geophys. J. Int.,* **194,** 1640–1664.
Robertsson, J. & Chapman, C., 2000. An efficient method for calculating finite-difference seismograms after model alterations, *Geophysics,* **65**(3), 907–918.
Sava, P. & Biondo, B., 2004. Wave-equation migration velocity analysis. I. Theory, *Geophys. Prospect.,* **52,** 593–606.
Schuster, G., 2009. *Seismic Interferometry,* Cambridge Univ. Press.
Schuster, G. & Zhou, M., 2006. A theoretical overview of model-based and correlation-based redatuming methods, *Geophysics,* **71**(4), SI103–SI110.
Shin, C., Jang, S. & Min, D.-J., 2001. Improved amplitude preservation for prestack depth migration by inverse scattering theory, *Geophys. Prospect.,* **49,** 592–606.
Stork, C. & Clayton, R., 1991. Linear aspects of tomographic velocity analysis, *Geophysics,* **56,** 483–495.
Tang, Y. & Biondi, B., 2011. Target-oriented wavefield tomography using synthesized born data, *Geophysics,* **76**(5), WB191–WB207.
Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation, *Geophysics,* **49**(8), 1259–1266.
Teng, Z.-H., 2003. Exact boundary condition for time-dependent wave equation based on boundary integral, *J. Comput. Phys.,* **190,** 398–418.
van Leeuwen, T. & Mulder, W.A., 2007. *Data Correlation for Velocity Inversion,* chap. 364, pp. 1800–1804, Society of Exploration Geophysicists.
van Manen, D.-J., Robertsson, J.O. & Curtis, A., 2005. Modeling of wave propagation in inhomogeneous media, *Phys. Rev. Lett.,* **94**(16), 164301, doi:10.1103/PhysRevLett.94.164301.
van Manen, D.-J., Curtis, A. & Robertsson, J.O., 2006. Interferometric modeling of wave propagation in inhomogeneous elastic media using time reversal and reciprocity, *Geophysics,* **71**(4), SI47–SI60.
van Manen, D.-J., Robertsson, J. O.A. & Curtis, A., 2007. Exact wave field simulation for finite-volume scattering problems, *J. acoust. Soc. Am.,* **122**(4), EL115–EL121.
Vasmel, M., Robertsson, J.O.A., van Manen, D.-J. & Curtis, A., 2013. Immersive experimentation in a wave propagation laboratory, *J. acoust. Soc. Am.,* **134**(6), EL492–EL498.
Virieux, J. & Operto, S., 2009. An overview of full-waveform inversion in exploration geophysics, *Geophysics,* **74**(6), WCC1–WCC26.
Wang, S., de Hoop, M.V. & Xia, J., 2011. On 3D modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver, *Geophys. Prospect.,* **59**(5), 857–873.
Wiggins, J., 1984. Kirchhoff integral extrapolation and migration of nonplanar data, *Geophysics,* **49,** 1239–1248.
Yang, D., Zheng, Y., Fehler, M. & Malcolm, A., 2012. Target-oriented time-lapse waveform inversion using virtual survey, *SEG Technical Program Expanded Abstracts 2012,* pp. 1–5.
Zepeda-Núñez, L. & Demanet, L., 2014. The method of polarized traces for the 2D Helmholtz equation, doi:10.1016/j.jcp.2015.11.040.

APPENDIX A: DERIVATION OF NUMERICALLY MATCHING SCATTERED FIELD

Model perturbations δm act as scattering sources in the Lippmann–Schwinger equation (Keys & Weglein 1983) for the scattered field u_s. Eq. (2) shows that the model perturbations δm are limited to C. Therefore, outside of C, the model is the background model and the scattered wavefield is sourceless:

$$-m_0(x)\omega^2 u_s(x,\omega) - \Delta_h u_s(x,\omega) = 0, \quad x \in \Omega \setminus C. \tag{A1}$$

Recall that the model is unperturbed on the boundary ∂C of the truncated domain. It is also known that the numerical Green's function in the background model m_0 satisfies

$$-m_0(x)\omega^2 G_0(x,y,\omega) - \Delta_h G_0(x,y,\omega) = \delta(x-y), \quad x,y \in \Omega. \tag{A2}$$

Multiplying eq. (A1) by $G_0(x,y,\omega)$ and subtracting eq. (A2) multiplied by $u_s(x,\omega)$ gives

$$-G_0(x,y,\omega)\Delta_h u_s(x,\omega) + u_s(x,\omega)\Delta_h G_0(x,y,\omega) = -u_s(x,\omega)\delta(x-y), \quad x \in \Omega \setminus C, y \in \Omega. \tag{A3}$$

Both sides of eq. (A3) will be summed over the nodes in $B \cup \partial C$ to get a convenient expression for the scattered field u_s. Summation by parts is used to turn eq. (A3) into an expression resembling a boundary integral.

Figure A1. Zoomed in section of the black dotted box in Fig. 1.

To make the rest of the derivation easier to follow, a 1-D version of eq. (A3) is first summed along a line. The Laplacian operator Δ_h becomes the three-point second-order derivative stencil in 1-D. The left-hand side of eq. (A3) centred around node i is $d^{(i)}$ and in 1-D becomes:

$$d^{(i)} = -G_0^{(i)} \frac{u_s^{(i+1)} - 2u_s^{(i)} + u_s^{(i-1)}}{h^2} + u_s^{(i)} \frac{G_0^{(i+1)} - 2G_0^{(i)} + G_0^{(i-1)}}{h^2}, \tag{A4}$$

where for notational convenience the dependence on x,y, and ω has temporarily been dropped. The detour through the 1-D case will be useful when finding an expression for the 2-D summation of eq. (A3) with the five-point Laplacian stencil. Fig. A1 shows the 1-D setup of the problem. This figure corresponds to the dotted black box in Fig. 1. Eq. (A4) is first written in a more convenient way:

$$d^{(i)} = -G_0^{(i)} \left(a^{(i+1)} - a^{(i)}\right) + u_s^{(i)} \left(b^{(i+1)} - b^{(i)}\right), \tag{A5}$$

with the coefficients $a^{(i)}$ and $b^{(i)}$ defined as:

$$a^{(i)} = \frac{u_s^{(i)} - u_s^{(i-1)}}{h^2}, \quad b^{(i)} = \frac{G_0^{(i)} - G_0^{(i-1)}}{h^2} \tag{A6}$$

Using summation by parts to sum $d^{(i)}$ over its domain of definition (i.e. the red and black nodes in the 1-D version of $B \cup \partial C$ in Fig. A1), the following expression is obtained:

$$\sum_{i=m}^{n} d^{(i)} = -\left[G_0^{(n+1)} a^{(n+1)} - G_0^{(m)} a^{(m)}\right] + \sum_{i=m}^{n} a^{(i+1)} \left(G_0^{(i+1)} - G_0^{(i)}\right) + \left[u_s^{(n+1)} b^{(n+1)} - u_s^{(m)} b^{(m)}\right] - \sum_{i=m}^{n} b^{(i+1)} \left(u_s^{(i+1)} - u_s^{(i)}\right). \tag{A7}$$

If we recognize that $\left(G_0^{(i+1)} - G_0^{(i)}\right) = h^2 b^{(i+1)}$ and $\left(u_s^{(i+1)} - u_s^{(i)}\right) = h^2 a^{(i+1)}$, the sums cancel. With this simplification, eq. (A7) becomes

$$\sum_{i=m}^{n} d^{(i)} = -\left[G_0^{(n+1)} a^{(n+1)} - G_0^{(m)} a^{(m)}\right] + \left[u_s^{(n+1)} b^{(n+1)} - u_s^{(m)} b^{(m)}\right] = \frac{1}{h^2}\left[u_s^{(n)} G_0^{(n+1)} - G_0^{(n)} u_s^{(n+1)}\right] + \frac{1}{h^2}\left[u_s^{(m)} G_0^{(m-1)} - G_0^{(m)} u_s^{(m-1)}\right]. \tag{A8}$$

Eq. (A8) can be manipulated to show that $\sum_{i=m}^{n} d^{(i)}$ can be rewritten in terms of weighted normal derivatives on both boundaries:

$$\sum_{i=m}^{n} d^{(i)} = \frac{1}{h^2}\left[u_s^{(n)} \left(G_0^{(n+1)} - G_0^{(n)}\right) - G_0^{(n)} \left(u_s^{(n+1)} - u_s^{(n)}\right)\right] + \frac{1}{h^2}\left[u_s^{(m)} \left(G_0^{(m-1)} - G_0^{(m)}\right) - G_0^{(m)} \left(u_s^{(m-1)} - u_s^{(m)}\right)\right]. \tag{A9}$$

We now use eq. (A8) in the following derivation for the 2-D summation of eq. (A3). The left-hand side of eq. (A3) is called $d^{(i,j)}$ with index i referring to the grid row and index j referring to the grid column.

$$\begin{aligned} d^{(i,j)} = &\left(-G_0^{(i,j)} \frac{u_s^{(i+1,j)} - 2u_s^{(i,j)} + u_s^{(i-1,j)}}{h^2} + u_s^{(i,j)} \frac{G_0^{(i+1,j)} - 2G_0^{(i,j)} + G_0^{(i-1,j)}}{h^2}\right) \\ &+ \left(-G_0^{(i,j)} \frac{u_s^{(i,j+1)} - 2u_s^{(i,j)} + u_s^{(i,j-1)}}{h^2} + u_s^{(i,j)} \frac{G_0^{(i,j+1)} - 2G_0^{(i,j)} + G_0^{(i,j-1)}}{h^2}\right). \end{aligned} \tag{A10}$$

The elements of the five-point Laplacian stencil have been ordered in such a way that the first line of eq. (A10) contains all the vertical derivatives and the second line contains all the horizontal derivatives. It is important to note that both lines resemble eq. (A4). When finding a simplified expression for the 2-D summation of eq. (A10) over $B \cup \partial C$, it is useful to investigate the summation over both lines of eq. (A10) individually.

First we will focus on the 2-D summation of the first line of eq. (A10) over the domain $B \cup \partial C$ (i.e. the black and red nodes). We compute this 2-D summation within a column and then add all of these column sums. The sum over all the elements within a column is expressed in terms of nodal contributions at the top and bottom boundary of the mesh in Fig. 1. This is visualized in the left blue box in Fig. A2(a) and follows from the 1-D derivation that the first line of (A10) has the form of (A4). Summation over columns crossing C is handled using the same logic from the 1-D derivation. The column sum is split into two parts, which are represented by the two other blue boxes in Fig. A2(a). The first part is summation over the top subcolumn extending from the top of the full mesh to the node on the top boundary of the truncated mesh. The second part is summation over the bottom subcolumn extending from the bottom node on the truncated mesh boundary to the bottom of the full mesh. Summation over both subcolumns is also expressed in terms of weighted normal derivatives at their boundaries. Adding all these weighted normal derivatives, indicated by arrows in Fig. A2(a), equals a sum over all grid points of the first line of eq. (A10).

The same idea can be repeated with the second line of eq. (A10). But now summation is over all the rows. Rows intersecting C are split into two subrows whose contributions are added. The summation over a row or subrow is again expressed in terms of weighted normal derivatives as in the 1-D version eq. (A8). Therefore, summation of the second line of eq. (A10) over $B \cup \partial C$ results in horizontal normal derivatives that are added to the vertical normal derivatives of Fig. A2(a). The summation of eq. (A10) over $B \cup \partial C$ is the combination of both lines of the equation. Therefore, it includes the horizontal and vertical normal derivatives on the boundaries, as is illustrated in Fig. A2(b). One important observation is that the corner nodes of ∂C do not contribute.

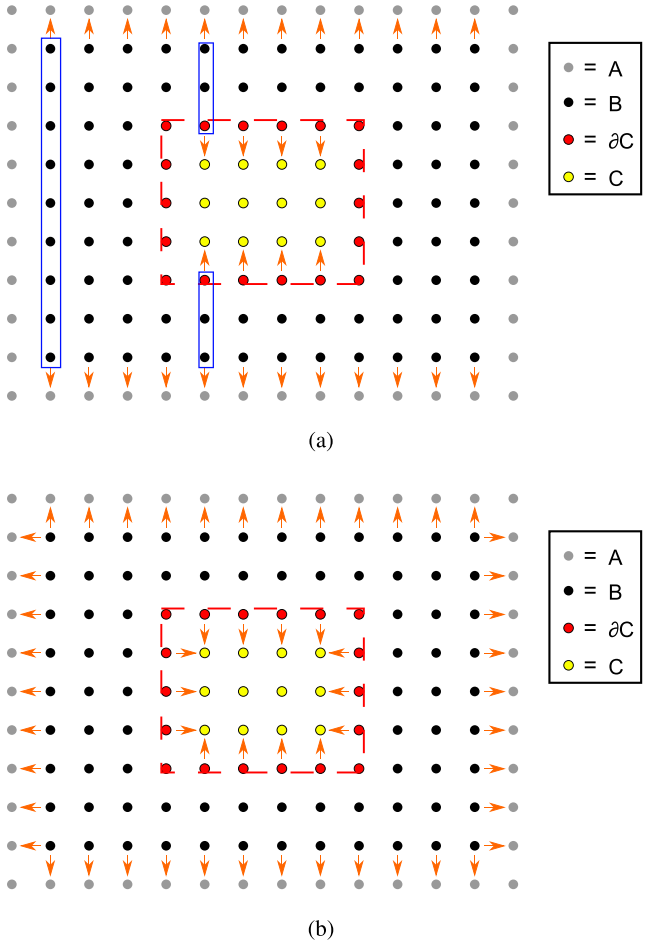

Figure A2. (a) The summation of the first line in eq. (A10) results in weighted normal derivatives indicated by orange arrows. The left blue box represents a column not intersecting C. Summation over this column results in weighted normal derivatives at the top and bottom of the full mesh. The two smaller blue boxes represent subcolumns whose summation also results in weighted normal derivatives at the endpoints. (b) Additional summation over the second line adds the horizontal normal derivatives to the vertical ones of panel (a). The contribution on the outer boundary adds up to zero and can therefore be neglected.

Summing the right-hand side of eq. (A3) over the volume $B \cup \partial C$ gives $-u^s(y, \omega)$ for y in $B \cup \partial C$. Therefore, the summation of all the weighted normal derivatives in Fig. A2(b) gives the scattered field:

$$\sum \frac{1}{h^2}\left(u_s^{(n)}\left(G_0^{(n+1)} - G_0^{(n)}\right) - G_0^{(n)}\left(u_s^{(n+1)} - u_s^{(n)}\right)\right) = -u_s(y, \omega), \quad y \in B \cup \partial C, \tag{A11}$$

where the summation is over all non-corner boundary nodes and the superscript n is the value on the boundary, and the subscript $n + 1$ one node to the exterior from B. One problem with this boundary summation is that it requires explicit knowledge of the scattered wavefield. There is an analogous derivation for the background field u_0, which is generated by a source in B. The left-hand side turns into a weighted normal derivative summation with u_0 replacing u_s, and the right-hand side is zero. Adding that expression to eq. (A11) and using $u = u_0 + u_s$ gives

$$\sum \frac{1}{h^2}\left(u^{(n)}\left(G_0^{(n+1)} - G_0^{(n)}\right) - G_0^{(n)}\left(u^{(n+1)} - u^{(n)}\right)\right) = -u_s(y, \omega), \quad y \in B \cup \partial C, \tag{A12}$$

where the scattered field in $B \cup \partial C$ is now expressed in terms of the total field u at the boundaries. A more detailed investigation would show that the weighted normal derivatives on the outer boundary sum to zero. Therefore, the summation in eq. (A12) can be limited to the boundary of the truncated domain, where the reader is reminded that the corner nodes are not involved as can be seen in Fig. A2(b). The derivation in this appendix can easily be extended to 3-D.

APPENDIX B: SALT BOUNDARY INVERSION METHOD

The salt boundary inversion method used here follows the one described in Lewis *et al.* (2012). The method is a re-formulation of the traditional full-waveform inversion problem to invert for the salt boundary alone by using a level set approach to parametrize the salt geometry.

The level set representation involves constructing a function $\phi : \mathbb{R}^k \to \mathbb{R}$ ($k = 2$ in 2-D) such that its zero level set is the boundary ∂S of the geobody $S \subset \Omega$ that we want to represent (i.e. the salt boundary). The level set function ϕ is chosen such that it is positive inside the geobody and negative outside the geobody.

The traditional least squares formulation of the full-waveform inversion problem is posed as the minimization of the following data misfit function (Virieux & Operto 2009),

$$\min_m f(m) = \frac{1}{2}\|F(m) - d\|_2^2, \tag{B1}$$

where, $m \in \mathbb{R}^n$ represents the earth model and F is the forward modelling operator that simulates the seismic data given an earth model and d is the observed seismic data.

Using the level set representation to parametrize the salt geometry, we reformulate the inversion as follows:

$$\min_{\hat{m}} \hat{f}(\hat{m}) = \frac{1}{2}\|F(g(\hat{m})) - d\|_2^2, \tag{B2}$$

where, $\hat{m}$ be the implicit model defined using a level set function $\phi \in \Omega$, and $g : \Omega \to \mathbb{R}^n$ is the operator that maps the implicit model onto the grid used by the forward modelling operator. So $F(g(\hat{m}))$ is the modelled data for a particular salt boundary shape.

The gradient of the new objective function is computed with the chain rule,

$$\nabla \hat{f}(\hat{m}) = \sum_{i=1}^{n} \frac{\partial f}{\partial m_i} \nabla g_i(\hat{m}), \tag{B3}$$

At each iteration, the perturbations to the level set boundary are made by evolving the level set function ϕ, using the method introduced by Osher & Sethian (1988). The equation of motion controlling the evolution of the level set function is given by

$$\phi_t + V \cdot \nabla\phi = c\kappa|\nabla\phi|, \tag{B4}$$

where, $\kappa = \nabla \cdot (\frac{\nabla\phi}{|\nabla\phi|})$ is the curvature of the surface, $c \geq 0$ is the curvature weight and V is the directional force acting on the level set surface. By appropriately choosing the directional force V and solving the above differential equation, we can control the perturbations that are made to the level set surface representing the salt geometry.

We use the negative of the gradient as the force acting on the level set surface to evolve ϕ, analogous to the steepest descent method. That is, at iteration k, evolve ϕ according to

$$\phi_t - \nabla \hat{f}_k(\hat{m}_k) \cdot \nabla\phi = c\kappa|\nabla\phi|. \tag{B5}$$

The evolution is stopped when $\alpha_k = t_k - t_{k-1}$, the step length along the time dimension, meets certain conditions, such as the Wolfe conditions (Gill *et al.* 1981), determined by a line-search procedure. The line search ensures that the value of the objective function is reduced at every iteration. At the end of the iteration, the gradient is recomputed using the new solution for ϕ and the method is restarted using the newly computed gradient.

Because the gradient $\nabla \hat{f}_k(\hat{m}_k)$ is defined only on the boundary of the geobody, to be able to use eq. (B5) to evolve the level set function, we must provide a continuous extension of the gradient off ∂S onto Ω. There are several ways in which this can be done, and we choose the method outlined by Fedkiw *et al.* (1999). First, we choose the level set function to be the signed distance function (Osher & Fedkiw 2003),

$$\phi(x) = \inf_{y \in \partial S} \|x - y\|_2 \mathrm{sgn}(\phi_0), \tag{B6}$$

$$\text{where } \mathrm{sgn}(\phi) = \frac{\phi}{\sqrt{\phi^2 + \delta}}, \qquad \phi_0(x) = \begin{cases} 1 & x \in S \\ -1 & x \in S^c \end{cases}.$$

The signed distance function is constructed by solving

$$\phi_t + \mathrm{sgn}(\phi_0)(|\nabla\phi| - 1) = 0. \tag{B7}$$

A constant extrapolation of the gradient along the normal direction to ∂S is then computed by solving the following advection equation:

$$I_t \pm \frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla I = 0, \tag{B8}$$

where I is the quantity being advected, which in our case will be the components of the gradient vector. Because, at every iteration, we can assume that the perturbation to the boundary is small, this extrapolation step can be optimized such that the gradient must be defined only in a small neighbourhood of the boundary ∂S.

APPENDIX C: NUMERICALLY EXACT MULTIDOMAIN SOLVER

It is possible to solve for the identical wavefield a full domain solver would have generated in an arbitrary number of disjoint truncated domains. The derivation of this multidomain local solver is analogous to the single truncated domain case. Instead of having a single truncated domain interior C with boundary ∂C, there is now a collection of truncated domain interiors $C_1, C_2, \ldots, C_N$ with boundaries $\partial C_1, \partial C_2, \ldots, \partial C_N$. The difference in the wavefield due to model perturbations inside the truncated domain interiors is again called u_s. Performing a derivation

entirely analogous to the one in Appendix A, the scattered field is expressed using eq. (7). In this case, the summation takes place along the boundaries of all the truncated domains, again excluding their corner nodes. This means that eq. (7) uses the total wavefield both at the boundaries of the truncated domains and one layer to the interior, and the Green's functions from those nodes to any node not in the interior of the truncated domains to get the scattered field at that location. With this expression a system of equations similar to eq. (10) is derived to truncate the computational domain.

As before, a spiralling node numbering scheme is used with node numbering starting in the top left and increasing counter-clockwise. The vector $\mathbf{u_s}^{\partial C_j}$ refers to the scattered field on the boundary nodes ∂C_j. The wavefield on the boundary nodes ∂C_j in the background model m_0 is $\mathbf{u_0}^{\partial C_j}$. The total field at the same boundary is $\mathbf{u}^{\partial C_j}$. The total field k layers to the interior is $\mathbf{u}^{\partial C_{j,+k}}$. Just like in the single domain case, a system of equations is assembled:

$$A\mathbf{x} = \mathbf{b} \tag{C1}$$

with matrix A defined in eq. (C2)

$$A = \left(\begin{array}{ccc|ccc|ccc|c} -I & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & \ddots & 0 & 0 & \ddots & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & -I & 0 & 0 & I & 0 & 0 & 0 & \dots \\ \hline I & 0 & 0 & G^{\partial C_{1,+1}}_{0,\to\partial C_1} & \dots & G^{\partial C_{N,+1}}_{0,\to\partial C_1} & -G^{\partial C_1}_{0,\to\partial C_1} & \dots & -G^{\partial C_N}_{0,\to\partial C_1} & 0 \ \dots \\ 0 & \ddots & 0 & \vdots & & \vdots & \vdots & & \vdots & 0 \ \dots \\ 0 & 0 & I & G^{\partial C_{1,+1}}_{0,\to\partial C_N} & \dots & G^{\partial C_{N,+1}}_{0,\to\partial C_N} & -G^{\partial C_1}_{0,\to\partial C_N} & \dots & -G^{\partial C_N}_{0,\to\partial C_N} & 0 \ \dots \\ \hline & 0 & & & & -\mathbf{m}(x)\omega^2 - \Delta_h & & & & \end{array}\right) \tag{C2}$$

and

$$\mathbf{x} = \left(\begin{array}{c} \mathbf{u_s}^{\partial C_1} \\ \vdots \\ \mathbf{u_s}^{\partial C_N} \\ \hline \mathbf{u}^{\partial C_1} \\ \vdots \\ \mathbf{u}^{\partial C_N} \\ \hline \mathbf{u}^{\partial C_{1,+1}} \\ \vdots \\ \mathbf{u}^{\partial C_{N,+1}} \\ \hline \mathbf{u}^{\partial C_{1,+2}} \\ \vdots \\ \mathbf{u}^{\partial C_{1,+M_1}} \\ \vdots \\ \mathbf{u}^{\partial C_{N,+2}} \\ \vdots \\ \mathbf{u}^{\partial C_{N,+M_N}} \end{array}\right) \quad \text{and} \quad \mathbf{b} = \left(\begin{array}{c} \mathbf{u_0}^{\partial C_1} \\ \vdots \\ \mathbf{u_0}^{\partial C_N} \\ \hline 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right), \tag{C3}$$

where the number of layers in truncated domain j is M_j. The Green's block $G^{\partial C_i}_{0,\to\partial C_j}$ contains the Green's functions from the boundary ∂C_i to the boundary ∂C_j where the scattered field is computed. Similarly the block $G^{\partial C_{i,+1}}_{0,\to\partial C_j}$ contains Green's functions between nodes one layer to the interior of ∂C_i to the nodes on the boundary of ∂C_j. The 0 subscript emphasizes that the Green's functions correspond to the background velocity state.

The multidomain matrix in eq. (C2) has a similar structure as the single domain case of eq. (10). In fact, it reduces to eq. (10) when N is set to one. The multidomain analogy of the relations in List 1 is used when constructing eq. (C2). Horizontal lines in eq. (C2) and in vector $\mathbf{b}$ in eq. (C3) divide the matrix in three sections which are not drawn to scale. These sections are not related to the sections in vector $\mathbf{x}$ in eq. (C3). The first section expresses that the difference between the perturbed and the background wavefields is the scattered wavefield on the boundary of every truncated domain. The second section expresses that the scattered field on boundary nodes of any domain is expressed in terms of eq. (7), with summation now following the boundary of all domains as explained at the start of this appendix. The third section expresses that in the interior of every domain the regular Helmholtz equation holds.

Solving eq. (C1) for arbitrary model perturbations in $C_1, C_2, \dots, C_N$ gives in every truncated domain the identical local wavefields a full domain solver gives, with relative differences defined by eq. (11) in the order of $\epsilon \sim 10^{-12}-10^{-16}$ regardless of the size of the perturbations.

Because this local solution is numerically equal to the full domain solution, it means that all interactions between the truncated domains are included, in addition to the interactions with the unperturbed background model. The cost of solving eq. (C1) is similar to the cost of the single domain case of eq. (10) as long as the total number of boundary nodes in the multidomain case is similar to the number of boundary nodes of the single domain. The single domain case will be slightly slower, because for an equal number of boundary nodes a single domain has more total nodes. Qualitatively, the matrices in equations (C2) and 10 are otherwise the same.

After the local wavefields are obtained through eq. (C1), the scattered wavefield is propagated to the receiver locations through the multidomain version of eq. (7). The residuals are then obtained and the adjoint wavefields are computed locally. Cross-correlation with the local wavefields gives the local gradients, which exactly match the gradients obtained from a full domain solver. This makes it possible to do simultaneous inversion in multiple disjointed domains.