Frequency domain tomographic full waveform inversion

Rahul Sarkar and Biondo Biondi

ABSTRACT

We develop the theory of frequency domain tomographic full waveform inversion (TFWI) with the goal of alleviating some of the computational bottlenecks present in its time domain implementation. Fourier transforming the time coordinate converts the extended Born modeling operator into a pointwise multiplication operator in the frequency domain, which is computationally cheap as compared to its time domain counterpart that involves computing expensive convolutions in time. This transformation leads to significant computational gains for the inner loop sub-problem in TFWI, where for each frequency band being inverted for, the factorization of the sparse Helmholtz matrix only needs to be performed once for the whole sub-problem. The theory is developed for the acoustic case in a constant density medium, ans we provide examples of some preliminary 2D numerical tests.

INTRODUCTION

TFWI, introduced by Biondi and Almomin (2014), belongs to a class of algorithms based on the idea of model extensions developed to overcome the issue of non-convexity of full waveform inversion (FWI). It is a widely held belief that the success of these methods is intimately linked to the ability of the extended modeling operator in each case to model the recorded wave field. While TFWI attempts to perform this using a time lag extension, other forms of extensions have also been tried successfully, for example extensions in subsurface offset or equivalently subsurface angles (Rickett and Sava, 2002; Sava and Fomel, 2003), and source extensions (Huang and Symes, 2015; Huang et al., 2016). When viewed as an optimization problem, these algorithms relax the original FWI problem with an extended model, which plays the role of artificial variables introduced into the problem that allow for the solution of an easier extended inverse problem. The physical variables or the velocities are eventually recovered by solving a sequence of such problems using a strategy that gradually removes the need for the artificial variables, for example by using regularization.

In the case of TFWI, Biondi and Almomin (2014) proposes to minimize the following objective function in order to achieve the above mentioned objectives

$$J(\mathbf{c}, \delta\mathbf{c}) = \sum_{i=1}^{N_s} \frac{1}{2} ||\mathbf{R}_i(\mathbf{u}_i + \delta\mathbf{u}_i) - \mathbf{d}_i||_2^2 + \frac{\epsilon^2}{2} || \, t\delta\mathbf{c}||_2^2 \, , \tag{1}$$

where N_s is the number of sources, $\delta\mathbf{c}$ is the non-physical model; and for the i^{th} source $\mathbf{d}_i$ is the recorded data, $\mathbf{R}_i$ is the operator that samples the wave field at the receiver locations, and $\mathbf{u}_i$ and $\delta\mathbf{u}_i$ are the physical and non-physical wave fields respectively, corresponding to the solutions of the following wave equations with zero initial Cauchy data

$$\begin{aligned} &\left(\frac{1}{\mathbf{c}^2(\mathbf{x})}\frac{\partial^2}{\partial t^2} - \nabla^2\right)\mathbf{u}_i(\mathbf{x},t) = \mathbf{f}_i(\mathbf{x},t)\ , \\ &\left(\frac{1}{\mathbf{c}^2(\mathbf{x})}\frac{\partial^2}{\partial t^2} - \nabla^2\right)\delta\mathbf{u}_i(\mathbf{x},t) = \int_{\mathbb{R}} 2\frac{\partial^2 \mathbf{u}_i}{\partial t^2}(\mathbf{x},t-\tau)\frac{\delta\mathbf{c}(\mathbf{x},\tau)}{\mathbf{c}^3(\mathbf{x})}\ d\tau, \end{aligned} \tag{2}$$

In equation (2), $\mathbf{c}$ denotes a physical velocity model, $\mathbf{f}_i$ represents the source function for the i^{th} source, ∇^2 denotes the Laplacian operator, $\mathbf{x} \in \mathbb{R}^d$ represents the physical space coordinate and t represents time. We will mostly be interested in the cases $d = 2, 3$. The second equation involves a convolution in time for each source separately, which makes both the forward and adjoint of the extended Born modeling operator expensive to compute. In the original version of the algorithm, the optimization problem in (1) is solved using the adjoint state method treating the problem as a non-linear optimization problem with respect to $\mathbf{c}$ and $\delta\mathbf{c}$ simultaneously, and using scale mixing. It is easily seen that for fixed $\mathbf{c}$, the minimization step for $\delta\mathbf{c}$ is a linear least squares problem (also called the inner-loop subproblem for TFWI), which is typically solved using some variant of the conjugate gradient (CG) algorithm. In fact, this observation has already been suggested previously by Barnier et al. (2017) in the context of variable-projection TFWI. However as CG is an iterative method, this requires us to repeatedly apply the extended Born modeling operator and its adjoint and hence perform the convolutions for each shot and for every CG iteration. This is the main bottleneck of TFWI, and it is clear that one must be able to perform the minimization step over $\delta\mathbf{c}$ more efficiently in order for the overall algorithm to be practical. The main contribution of this report is to demonstrate that the inner-loop subproblem can be made really efficient by solving an equivalent problem in the frequency domain. In the next sections, we develop the complete theoretical framework of this technique, and then provide some 2D synthetic examples.

THEORY

For reasons that will become clear, we will consider a slightly different problem from equation (1) given by the minimization of the following objective function

$$J(\mathbf{c},\delta\mathbf{c}) = \sum_{i=1}^{N_s} \frac{1}{2}||\mathbf{R}_i(\mathbf{u}_i + \delta\mathbf{u}_i) - \mathbf{d}_i||_2^2 + \frac{\epsilon^2}{2}||\ t\delta\mathbf{c}||_2^2 + \frac{\gamma^2}{2}||\delta\mathbf{c}||_2^2\ , \tag{3}$$

where $\gamma \neq 0$, but not too large. In this section we develop the theory, discuss the algorithm details, and then discuss how we obtain computational savings in the frequency domain algorithm as compared to the time domain algorithm.

Frequency domain transformation

The first step involves transforming all the quantities involved to the frequency domain. In this paper, any Fourier transformed quantity that will appear will be assumed to belong to the Schwartz class of functions in time, and hence also in frequency. We will denote the frequency domain coordinate by ω. Denoting by $\mathcal{F}$ the Fourier transform operator acting on the time coordinate, we first introduce the following notation where all variables with a "hat" are in the frequency domain, and are thus complex :

$$\begin{aligned}
&\hat{\mathbf{u}}_i = \mathcal{F}(\mathbf{u}_i) \ , \ \delta\hat{\mathbf{u}}_i = \mathcal{F}(\delta\mathbf{u}_i) \ , \\
&\hat{\mathbf{d}}_i = \mathcal{F}(\mathbf{d}_i) \ , \ \hat{\mathbf{f}}_i = \mathcal{F}(\mathbf{f}_i) \ , \\
&\delta\hat{\mathbf{c}} = \mathcal{F}(\delta\mathbf{c}) \ .
\end{aligned} \tag{4}$$

It should be noted that as all the time domain quantities $\mathbf{u}_i, \delta\mathbf{u}_i, \mathbf{d}_i, \mathbf{f}_i$ and $\delta\mathbf{c}$ are real valued, by property of the Fourier transform we also have the following symmetry conditions

$$\begin{aligned}
&\hat{\mathbf{u}}_i(\mathbf{x}, -\omega) = \overline{\hat{\mathbf{u}}_i(\mathbf{x}, \omega)} \ , \ \delta\hat{\mathbf{u}}_i(\mathbf{x}, -\omega) = \overline{\delta\hat{\mathbf{u}}_i(\mathbf{x}, \omega)} \ , \\
&\hat{\mathbf{d}}_i(\mathbf{x}, -\omega) = \overline{\hat{\mathbf{d}}_i(\mathbf{x}, \omega)} \ , \ \hat{\mathbf{f}}_i(\mathbf{x}, -\omega) = \overline{\hat{\mathbf{f}}_i(\mathbf{x}, \omega)} \ , \\
&\delta\hat{\mathbf{c}}(\mathbf{x}, -\omega) = \overline{\delta\hat{\mathbf{c}}(\mathbf{x}, \omega)} \ .
\end{aligned} \tag{5}$$

By Parseval's theorem, the L^2 norm is preserved by the Fourier transform, and hence Fourier transforming the right hand side of (3) allows us to rewrite the objective function as

$$J(\mathbf{c}, \delta\mathbf{c}) = J(\mathbf{c}, \delta\hat{\mathbf{c}}) = \sum_{i=1}^{N_s} \frac{1}{2} ||\mathbf{R}_i(\hat{\mathbf{u}}_i + \delta\hat{\mathbf{u}}_i) - \hat{\mathbf{d}}_i||_2^2 + \frac{\epsilon^2}{2} \left\| \frac{\partial}{\partial\omega} \delta\hat{\mathbf{c}} \right\|_2^2 + \frac{\gamma^2}{2} ||\delta\hat{\mathbf{c}}||_2^2 \ . \tag{6}$$

We also need to express the wave equations in (2) in the frequency domain, and so Fourier transforming them in time lead to the following Helmholtz equations with radiation boundary conditions

$$\begin{aligned}
&\left(\nabla^2 + \frac{\omega^2}{\mathbf{c}^2(\mathbf{x})}\right) \hat{\mathbf{u}}_i(\mathbf{x}, \omega) = -\hat{\mathbf{f}}_i(\mathbf{x}, \omega) \ , \\
&\left(\nabla^2 + \frac{\omega^2}{\mathbf{c}^2(\mathbf{x})}\right) \delta\hat{\mathbf{u}}_i(\mathbf{x}, \omega) = \frac{2\omega^2}{\mathbf{c}^3(\mathbf{x})} \hat{\mathbf{u}}_i(\mathbf{x}, \omega) \delta\hat{\mathbf{c}}(\mathbf{x}, \omega) \ .
\end{aligned} \tag{7}$$

Problem discretization

The next crucial step involves discretization of physical space, time and frequency that allows us to formulate and solve the inverse problem on a computer. We will proceed with a rectangular gridding strategy. Let $\Delta\mathbf{x} \in \mathbb{R}^d_{>0}$, $\Delta t \in \mathbb{R}_{>0}$ and $\Delta\omega \in \mathbb{R}_{>0}$ denote the grid spacings in space, time and frequency respectively. Let the number of grid points in space be given by $\mathbf{N} \in \mathbb{N}^d$, and in frequency be given by $N_\omega \in \mathbb{N}$, where $\mathbb{N}$ is the set of positive integers. Also letting N_i denote the components of $\mathbf{N}$ we will denote the total number of spatial grid points by $N = \prod_{i=1}^d N_i$. With this notation, the quantities $\hat{\mathbf{u}}_i, \delta\hat{\mathbf{u}}_i, \hat{\mathbf{f}}_i$ and $\delta\hat{\mathbf{c}}$ become elements of a finite dimensional vector space of dimension NN_ω, and $\mathbf{c}$ becomes an element of a vector space of dimension N. Finally, if N_{r_i} is the number of receivers present for the i^{th} source, then $\hat{\mathbf{d}}_i$ becomes an element of a vector space of dimension $N_{r_i}N_\omega$.

We solve the Helmholtz equations in (7) using the numerical scheme outlined in Liu and Ying (2016), where the radiation boundary condition is imposed using perfectly matched layers (Johnson, 2008; Sommerfeld, 1949). We have previously demonstrated numerical results from 2D experiments implementing this numerical scheme in Sarkar and Biondi (2017), and as was also remarked in that paper, the numerical scheme extends naturally to three and higher spatial dimensions. Thus for the k^{th} frequency ω_k, solving the equations in (7) reduce to solving the following linear systems of equations :

$$\begin{aligned} \mathbf{A}(\omega_k)\hat{\mathbf{u}}_i(\mathbf{x},\omega_k) &= -\hat{\mathbf{f}}_i(\mathbf{x},\omega_k) \ , \\ \mathbf{A}(\omega_k)\delta\hat{\mathbf{u}}_i(\mathbf{x},\omega_k) &= \frac{2\omega_k^2}{\mathbf{c}^3(\mathbf{x})}\hat{\mathbf{u}}_i(\mathbf{x},\omega_k)\delta\hat{\mathbf{c}}(\mathbf{x},\omega_k) \ , \end{aligned} \tag{8}$$

where $\mathbf{A}(\omega_k)$ is a complex matrix of dimensions $N \times N$ that additionally depends on $\mathbf{c}(\mathbf{x})$. In order to simplify the notation, we will furthermore denote all the matrix and vector quantities involved for each frequency ω_k, indexed by k as follows :

$$\begin{aligned} &\mathbf{A}_k = \mathbf{A}(\omega_k) \ , \ \mathbf{\Lambda}_{ki} = \text{diag}\left(\frac{2\omega_k^2}{\mathbf{c}^3(\mathbf{x})}\hat{\mathbf{u}}_i(\mathbf{x},\omega_k)\right) \ , \\ &\hat{\mathbf{u}}_{ki} = \hat{\mathbf{u}}_i(\mathbf{x},\omega_k) \ , \ \delta\hat{\mathbf{u}}_{ki} = \delta\hat{\mathbf{u}}_i(\mathbf{x},\omega_k) \ , \\ &\hat{\mathbf{d}}_{ki} = \hat{\mathbf{d}}_i(\mathbf{x},\omega_k) \ , \ \hat{\mathbf{f}}_{ki} = \hat{\mathbf{f}}_i(\mathbf{x},\omega_k) \ , \\ &\delta\hat{\mathbf{c}}_k = \delta\hat{\mathbf{c}}(\mathbf{x},\omega_k) \ . \end{aligned} \tag{9}$$

where $\mathbf{A}_k, \mathbf{\Lambda}_{ki} \in \mathbb{C}^{N\times N}$ are matrices, $\hat{\mathbf{u}}_{ki}, \delta\hat{\mathbf{u}}_{ki}, \hat{\mathbf{f}}_{ki}, \delta\hat{\mathbf{c}}_k \in \mathbb{C}^N$ are vectors, and $\hat{\mathbf{d}}_{ki} \in \mathbb{C}^{N_{r_i}}$ is a vector. The complete vectors $\hat{\mathbf{u}}_i, \delta\hat{\mathbf{u}}_i, \hat{\mathbf{f}}_i, \delta\hat{\mathbf{c}}$ and $\hat{\mathbf{d}}_i$ are obtained by stacking the vectors $\hat{\mathbf{u}}_{ki}, \delta\hat{\mathbf{u}}_{ki}, \hat{\mathbf{f}}_{ki}, \delta\hat{\mathbf{c}}_k$ and $\hat{\mathbf{d}}_{ki}$ respectively, for all the frequencies $\omega_1, \dots, \omega_{N_\omega}$. So for example, we have $\hat{\mathbf{u}}_i = [\hat{\mathbf{u}}_{1i}^T \dots \hat{\mathbf{u}}_{N_\omega i}^T]^T$. Using this notation, we may finally write the system of Helmholtz equations in (8) in the following compact form

$$\begin{aligned} \mathbf{A}_k\hat{\mathbf{u}}_{ki} &= -\hat{\mathbf{f}}_{ki} \ , \\ \mathbf{A}_k\delta\hat{\mathbf{u}}_{ki} &= \mathbf{\Lambda}_{ki}\delta\hat{\mathbf{c}}_k \ , \end{aligned} \tag{10}$$

and note that if $\Delta\mathbf{x}$ is chosen properly, then $\mathbf{A}_k$ admits an inverse (Liu and Ying, 2016), which allows us to be able to invert the linear system of equations in (10) and obtain the quantities $\hat{\mathbf{u}}_{ki}$ and $\delta\hat{\mathbf{u}}_{ki}$. We will formally write the solution to this system as

$$\begin{aligned} \hat{\mathbf{u}}_{ki} &= -\mathbf{A}_k^{-1}\hat{\mathbf{f}}_{ki} \ , \\ \delta\hat{\mathbf{u}}_{ki} &= \mathbf{A}_k^{-1}\mathbf{\Lambda}_{ki}\delta\hat{\mathbf{c}}_k \ , \end{aligned} \tag{11}$$

but note that we will never invert the matrix $\mathbf{A}_k$ in order to solve (10). The actual solution process in principle involves factorizing the matrix $\mathbf{A}_k$ into its lower and upper triangular factors, and then performing a forward substitution followed by a backward substitution to solve the systems of equations. We will provide more details of this procedure in later sections.

Finally, it is to be noted that the objective function in (6) is now transformed to the following form involving the vector quantities

$$J(\mathbf{c},\delta\hat{\mathbf{c}}) = \sum_{i=1}^{N_s}\frac{1}{2}||\mathbf{R}_i(\hat{\mathbf{u}}_i + \delta\hat{\mathbf{u}}_i) - \hat{\mathbf{d}}_i||_2^2 + \frac{\epsilon^2}{2} \parallel \mathbf{D}\delta\hat{\mathbf{c}}\|_2^2 + \frac{\gamma^2}{2}||\delta\hat{\mathbf{c}}||_2^2 \ , \tag{12}$$

with the understanding that the norm denoted by $||\cdot||$ now refers to the norm induced by the standard Hilbert space inner product, in the corresponding vector spaces for each of the different quantities, and $\mathbf{D}$ is the derivative operator along the frequency axis. For our convenience, we will take $\mathbf{D}$ to be the forward difference derivative operator.

The normal equations

In the time domain inner-loop sub-problem of TFWI, one seeks to find an extended model that is able to model the residuals using the extended Born modeling operator, for a fixed background velocity model. Posed an an optimization problem, this is equivalent to finding a minimizer $\delta\mathbf{c}^*$ of $J(\mathbf{c}, \delta\mathbf{c})$ for a fixed value of $\mathbf{c}$, that is

$$\delta\mathbf{c}^* = \underset{\delta\mathbf{c}}{\arg\min}\; J(\mathbf{c}, \delta\mathbf{c}) \; . \tag{13}$$

Because of the equivalence of the time domain and frequency domain objective functions, as shown in equation (6), the corresponding frequency domain solution $\delta\hat{\mathbf{c}}^*$ is given as

$$\delta\hat{\mathbf{c}}^* = \underset{\delta\hat{\mathbf{c}}}{\arg\min}\; J(\mathbf{c}, \delta\hat{\mathbf{c}}) \; , \tag{14}$$

with the time and frequency domain solutions for the extended model being related by $\delta\hat{\mathbf{c}}^* = \mathcal{F}(\delta\mathbf{c}^*)$. This correspondence will be shown shortly as we will show that the solution to problem (14) is unique. It is also clearly seen from the first equation in (11) that the vectors $\hat{\mathbf{u}}_{ki}$ are completely determined once we fix $\mathbf{c}$, and thus $\hat{\mathbf{u}}_i$ is also completely determined, because after all it is just the physical wavefield modeled using the background velocity model which has been Fourier transformed to the frequency domain. We will denote the misfit between the recorded data and the modeled physical wave field in the frequency domain as $\hat{\mathbf{r}}_i$, and define it as

$$\hat{\mathbf{r}}_i = \hat{\mathbf{d}}_i - \mathbf{R}_i \hat{\mathbf{u}}_i \; , \tag{15}$$

which allows us to rewrite the objective function for the inner-loop TFWI problem as

$$J_{\mathbf{c}}(\delta\hat{\mathbf{c}}) = \sum_{i=1}^{N_s} \frac{1}{2} ||\mathbf{R}_i \delta\hat{\mathbf{u}}_i - \hat{\mathbf{r}}_i||_2^2 + \frac{\epsilon^2}{2} \parallel \mathbf{D}\delta\hat{\mathbf{c}}||_2^2 + \frac{\gamma^2}{2} ||\delta\hat{\mathbf{c}}||_2^2 \; , \tag{16}$$

where the notation $J_{\mathbf{c}}(\delta\hat{\mathbf{c}})$ denotes the implicit dependence on the background velocity $\mathbf{c}$. As was previously remarked, the inner-loop TFWI sub-problem is a linear least squares problem, which is also true in the frequency domain. We can see this directly from the form of the objective function in (16), by additionally noting that $\delta\hat{\mathbf{u}}_i$ is related to $\delta\hat{\mathbf{c}}$ through a linear transformation which we next make explicit.

Using the fact that $\hat{\mathbf{u}}_i = [\hat{\mathbf{u}}_{1i}^T \dots \hat{\mathbf{u}}_{N_\omega i}^T]^T$, $\delta\hat{\mathbf{c}} = [\delta\hat{\mathbf{c}}_1^T \dots \delta\hat{\mathbf{c}}_{N_\omega}^T]^T$, and the second equation in (11), we can express the relationship between $\delta\hat{\mathbf{u}}_i$ and $\delta\hat{\mathbf{c}}$ using a block diagonal matrix as follows

$$\begin{bmatrix} \delta\hat{\mathbf{u}}_{1i} \\ \vdots \\ \delta\hat{\mathbf{u}}_{N_\omega i} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^{-1}\mathbf{\Lambda}_{1i} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{A}_{N_\omega}^{-1}\mathbf{\Lambda}_{N_\omega i} \end{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{c}}_1 \\ \vdots \\ \delta\hat{\mathbf{c}}_{N_\omega} \end{bmatrix} \; , \tag{17}$$

where the non-zero blocks $\mathbf{A}_k^{-1}\mathbf{\Lambda}_{ki}$ are of dimensions $N \times N$, and we will compactly write this relationship as $\delta\hat{\mathbf{u}}_i = \mathbf{B}_i \delta\hat{\mathbf{c}}$, with $\mathbf{B}_i$ representing the block diagonal matrix which again is completely determined for fixed $\mathbf{c}$. This allows us to write the objective function in (16) entirely in terms of $\delta\hat{\mathbf{c}}$, which can be further simplified as follows

$$\begin{aligned} J_{\mathbf{c}}(\delta\hat{\mathbf{c}}) &= \sum_{i=1}^{N_s} \frac{1}{2}||\mathbf{R}_i\mathbf{B}_i\delta\hat{\mathbf{c}} - \hat{\mathbf{r}}_i||_2^2 + \frac{\epsilon^2}{2} \parallel \mathbf{D}\delta\hat{\mathbf{c}}\|_2^2 + \frac{\gamma^2}{2}||\delta\hat{\mathbf{c}}||_2^2 \\ &= \frac{1}{2}\delta\hat{\mathbf{c}}^H \left(\sum_{i=1}^{N_s} \mathbf{B}_i^H\mathbf{R}_i^H\mathbf{R}_i\mathbf{B}_i + \epsilon^2\mathbf{D}^H\mathbf{D} + \gamma^2\mathbf{I} \right) \delta\hat{\mathbf{c}} \\ &\quad - \frac{1}{2}\sum_{i=1}^{N_s} \left(\hat{\mathbf{r}}_i^H\mathbf{R}_i\mathbf{B}_i\delta\hat{\mathbf{c}} + \delta\hat{\mathbf{c}}^H\mathbf{B}_i^H\mathbf{R}_i^H\hat{\mathbf{r}}_i \right) + \frac{1}{2}\sum_{i=1}^{N_s} \parallel \hat{\mathbf{r}}_i\|_2^2 \ , \end{aligned} \tag{18}$$

where $\mathbf{I}$ represents the identity matrix. The next observation is that the matrix $\mathbf{M} = \left(\sum_{i=1}^{N_s} \mathbf{B}_i^H\mathbf{R}_i^H\mathbf{R}_i\mathbf{B}_i + \epsilon^2\mathbf{D}^H\mathbf{D} + \gamma^2\mathbf{I}\right)$ is Hermitian and positive definite for any $\gamma \neq 0$, and therefore by Theorem 1 in Appendix A, $J_{\mathbf{c}}(\delta\hat{\mathbf{c}})$ has a unique minimizer $\delta\hat{\mathbf{c}}^*$ satisfying the following invertible linear system

$$\mathbf{M}\ \delta\hat{\mathbf{c}}^* = \mathbf{b}\ ,\ \text{where } \mathbf{b} = \sum_{i=1}^{N_s} \mathbf{B}_i^H\mathbf{R}_i^H\hat{\mathbf{r}}_i\ , \tag{19}$$

which is the normal equation for the inner-loop TFWI sub-problem.

Solving the normal equations

As the matrix $\mathbf{M}$ is Hermitian positive definite, the method of choice for solving the linear system is the conjugate gradient (CG) algorithm (Hestenes and Stiefel, 1952), each iteration of which requires us to form matrix vector products. From equation (A-8), it is clear that $\mathbf{M}$ consists of the sum of two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ of which the first is trivial to apply to a vector and poses no challenge. The second matrix $\mathbf{M}_2$ is more complicated, but it has a nice block diagonal structure, and we seek to exploit its structure to efficiently compute the result of its application to a vector. A necessary step towards achieving this goal is to precompute and store certain invariant quantities for the entire duration of solving the TFWI inner-loop subproblem.

Precomputing quantities

The quantities $\mathbf{A}_k^{-1}$ only depend on the frequencies ω_k and is independent of the shots. We start by forming the LU factors of the matrices $\mathbf{A}_k = \mathbf{L}_k\mathbf{U}_k$ and $\mathbf{A}_k^H = \tilde{\mathbf{L}}_k\tilde{\mathbf{U}}_k$ for each $k = 1, \ldots, N_\omega$ and store them. For each shot and each frequency ω_k, we can compute the primary wavefields $\hat{\mathbf{u}}_{ki}$, and hence also the quantities $\mathbf{\Lambda}_{ki}$. As $\mathbf{\Lambda}_{ki}$ is a diagonal matrix, we only need to store the diagonal entries of the matrix. Then for any vector $\mathbf{y}$ of appropriate dimensions, all of the following operations can be easily performed :

$$\mathbf{A}_k^{-1}\mathbf{y}, (\mathbf{A}_k^{-1})^H\mathbf{y}, \mathbf{\Lambda}_{ki}\mathbf{y} \text{ and } \mathbf{\Lambda}_{ki}^H\mathbf{y}\ ,\ \forall\, k = 1, \ldots, N_\omega,\ \text{and } \forall\, i = 1, \ldots, N_s\ . \tag{20}$$

The first two operations can be performed by forward and backward substitutions involving the precomputed LU factors, while the last two operations are trivial multiplications by precomputed diagonal matrices. The other quantity that we will precompute are the matrices $\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i$ for all shots $i = 1, \ldots, N_\omega$. As mentioned in Appendix B, $\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i$ is a diagonal $\{0, 1\}$ matrix with 1s only in the positions corresponding to the receiver locations. Thus $\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i$ can be stored easily by storing just the node indexes of the receivers corresponding to each shot, and quantities of the form $\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{y}$ for a vector $\mathbf{y}$ of appropriate dimension can be calculated as follows

$$(\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{y})_\alpha = \begin{cases} y_\alpha & \text{if a receiver is present at node index } \alpha \text{ ,} \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

The storage cost of the precomputed quantities is given in Table 1, where density refers to the fraction of non-zero entries of a matrix. It should be noted that all the matrices $\mathbf{L}_k$ are structurally same for all k, and the same is true for $\mathbf{U}_k$. Moreover, $\mathbf{U}_k^H$ is structurally same as $\tilde{\mathbf{L}}_k$, and similarly $\mathbf{L}_k^H$ is structurally same as $\tilde{\mathbf{U}}_k$. Also for 2D problems the density of the matrices $\mathbf{L}_k, \mathbf{U}_k, \tilde{\mathbf{L}}_k$ and $\tilde{\mathbf{U}}_k$ is approximately $\mathcal{O}(1/N)$.

Precomputed Quantity	Subscript Ranges	Storage Cost (number of floats)
$\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i$	$i = 1, \ldots, N_s$	$\sum_{i=1}^{N_s} N_{r_i}$
$\mathbf{\Lambda}_{ki}$	$k = 1, \ldots, N_\omega, \ i = 1, \ldots, N_s$	$N_\omega N_s N$
$\mathbf{L}_k, \mathbf{U}_k$	$k = 1, \ldots, N_\omega$	$N_\omega N^2(\text{density}(\mathbf{L}_1) + \text{density}(\mathbf{U}_1))$
$\tilde{\mathbf{L}}_k, \tilde{\mathbf{U}}_k$	$k = 1, \ldots, N_\omega$	$N_\omega N^2(\text{density}(\tilde{\mathbf{L}}_1) + \text{density}(\tilde{\mathbf{U}}_1))$

Table 1: Storage costs of the precomputed quantities in terms of the number of floating point variables.

Computing the matrix vector product

Once we have the precomputed quantities, the task of forming matrix vector products with the matrix $\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2$ is straightforward. The steps are outlined in Algorithm 1 where we apply $\mathbf{M}$ to $\delta\hat{\mathbf{c}} = [\delta\hat{\mathbf{c}}_1^T \ldots \delta\hat{\mathbf{c}}_{N_\omega}^T]^T$ for illustration. The output is a vector $\mathbf{y}$ which naturally also admits a block decomposition of the form $\delta\hat{\mathbf{c}}$, i.e. $\mathbf{y} = [\mathbf{y}_1^T \ldots \mathbf{y}_{N_\omega}^T]^T$. In the algorithm, it is assumed that the precomputed quantities in Table 1 are already available.

It is possible to write down the amount of work needed to perform the steps in Algorithm 1. Computing the $\mathbf{M}_1 \delta\hat{\mathbf{c}}$ updates costs $N(3N_\omega - 2)$ additions and $2NN_\omega$ multiplications. For the $\mathbf{M}_2 \delta\hat{\mathbf{c}}$ computation, all the updates excluding the linear solves cost $3NN_\omega N_s$ multiplications and $NN_\omega N_s$ additions. For each linear solve per shot per frequency, using forward and backward substitution with the cached LU factors costs roughly $N^2 \mathcal{O}(\text{density}(\mathbf{L}_1) + \text{density}(\mathbf{U}_1))$ multiplications and additions. Thus the total cost of the $\mathbf{M}_2 \delta\hat{\mathbf{c}}$ update is $N_\omega N_s N^2 \mathcal{O}(\text{density}(\mathbf{L}_1) + \text{density}(\mathbf{U}_1))$.

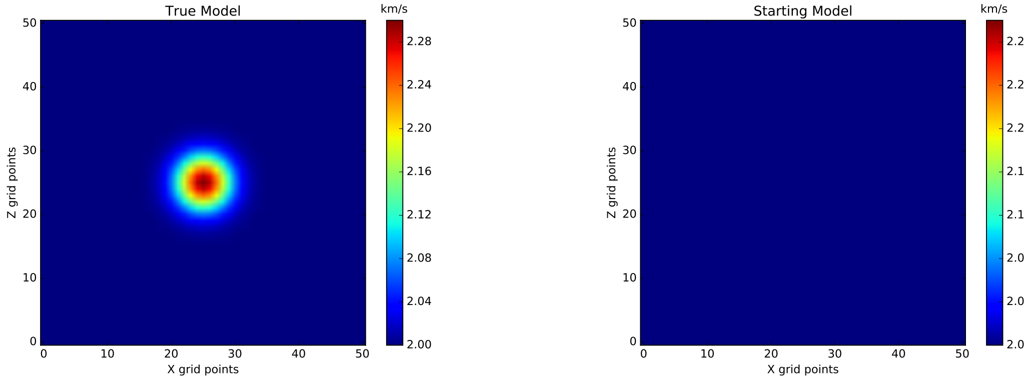
NUMERICAL EXPERIMENTS

In this section we provide some numerical results to test the performance of the algorithm proposed in this paper. We will use a simple model for our test cases that has a Gaussian anomaly in the center as the true model. We will denote this model as $\mathbf{c}_{\text{true}}$. The incorrect

Algorithm 1 Application of $\mathbf{M}$ to a vector

```
procedure MATRIX VECTOR PRODUCT ($\delta\hat{\mathbf{c}}$)

    // Initialize the output to zeros
    $\mathbf{y} \leftarrow 0$

    // Compute $\mathbf{M}_2 \delta\hat{\mathbf{c}}$
    for $k = 1$ to $N_\omega$ do
        for $i = 1$ to $N_s$ do
            $\mathbf{z} \leftarrow \mathbf{\Lambda}_{ki} \delta\hat{\mathbf{c}}_k$
            Solve: $\mathbf{L}_k \mathbf{U}_k \mathbf{w} = \mathbf{z}$
            $\mathbf{z} \leftarrow \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{w}$
            Solve: $\tilde{\mathbf{L}}_k \tilde{\mathbf{U}}_k \mathbf{w} = \mathbf{z}$
            $\mathbf{z} \leftarrow \mathbf{\Lambda}_{ki}^H \mathbf{w}$
            $\mathbf{y}_k \leftarrow \mathbf{y}_k + \mathbf{z}$
        end for
    end for

    // Compute $\mathbf{M}_1 \delta\hat{\mathbf{c}}$
    $\mathbf{y}_1 \leftarrow \mathbf{y}_1 + (\epsilon^2 + \gamma^2) \delta\hat{\mathbf{c}}_1 - \epsilon^2 \delta\hat{\mathbf{c}}_2$
    for $k = 2$ to $N_\omega - 1$ do
        $\mathbf{y}_k \leftarrow \mathbf{y}_k + (2\epsilon^2 + \gamma^2) \delta\hat{\mathbf{c}}_k - \epsilon^2 (\delta\hat{\mathbf{c}}_{k-1} + \delta\hat{\mathbf{c}}_{k+1})$
    end for
    $\mathbf{y}_{N_\omega} \leftarrow \mathbf{y}_{N_\omega} + (\epsilon^2 + \gamma^2) \delta\hat{\mathbf{c}}_{N_\omega} - \epsilon^2 \delta\hat{\mathbf{c}}_{N_\omega - 1}$

    return $\mathbf{y}$
end procedure
```

starting model will be a constant velocity background model, that we will denote using the already introduced notation $\mathbf{c}$. Both these models are plotted in Figures 1a and 1b respectively. The grid spacing in both X and Z directions is 10 m. The number of grid points for the models are given by the parameters $N_x = 50$ and $N_z = 50$. We will use a Ricker wavelet of peak frequency 10 Hz in our experiments.



(a)

(b)

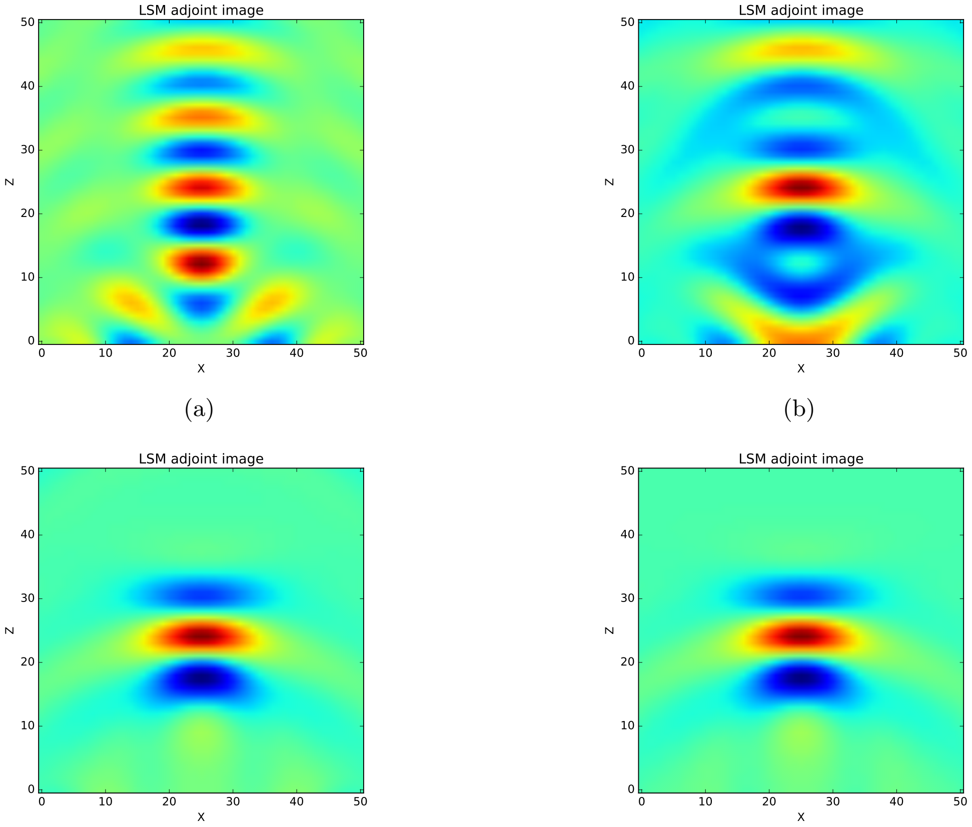
Figure 1: a) The true velocity model $\mathbf{c}_{\text{true}}$, b) The incorrect starting velocity model $\mathbf{c}$. [**ER**]

Effect of sampling in frequency on adjoint images

In the first experiment we study the effect of sampling in the frequency domain on the zero-lag adjoint extended image, which is obtained by stacking all the adjoint extended images along the frequency axis. The adjoint extended image for various frequencies is simply the right hand side $\mathbf{b}$ of equation (19). For this study, we choose a Ricker wavelet of peak frequency $f_p = 10$ Hz, which gives roughly a maximum frequency $f_{\text{max}} = 20$ Hz. We divide the frequency range $[0, f_{\text{max}}]$ into N_ω uniformly spaced points, for $N_\omega = 2, 4, 8, 16$ and compute $\mathbf{b}$ for each value of N_ω, and finally compute the adjoint zero-lag extended image by stacking the images at all the frequencies. The results are plotted in Figures 2a, 2b, 2c and 2d respectively. A fact that is clearly seen from this example is that the zero-lag images are almost the same for $N_\omega = 8$ and $N_\omega = 16$. This suggests a general direction, which remains to be fully explored, that not too many frequencies are needed to reconstruct the time domain adjoint extended image. Our experiments with the given model suggests that this fact continues to be true even for the adjoint images with non-zero lags.

Variation of adjoint images with frequency

In the next illustration we show how the adjoint extended images vary with different frequencies, and so give the reader some intuition of how these fields look like. We choose the case $N_\omega = 8$ for this illustration. The results are plotted in Figures 3a, 3b and Figures 4a, 4b.



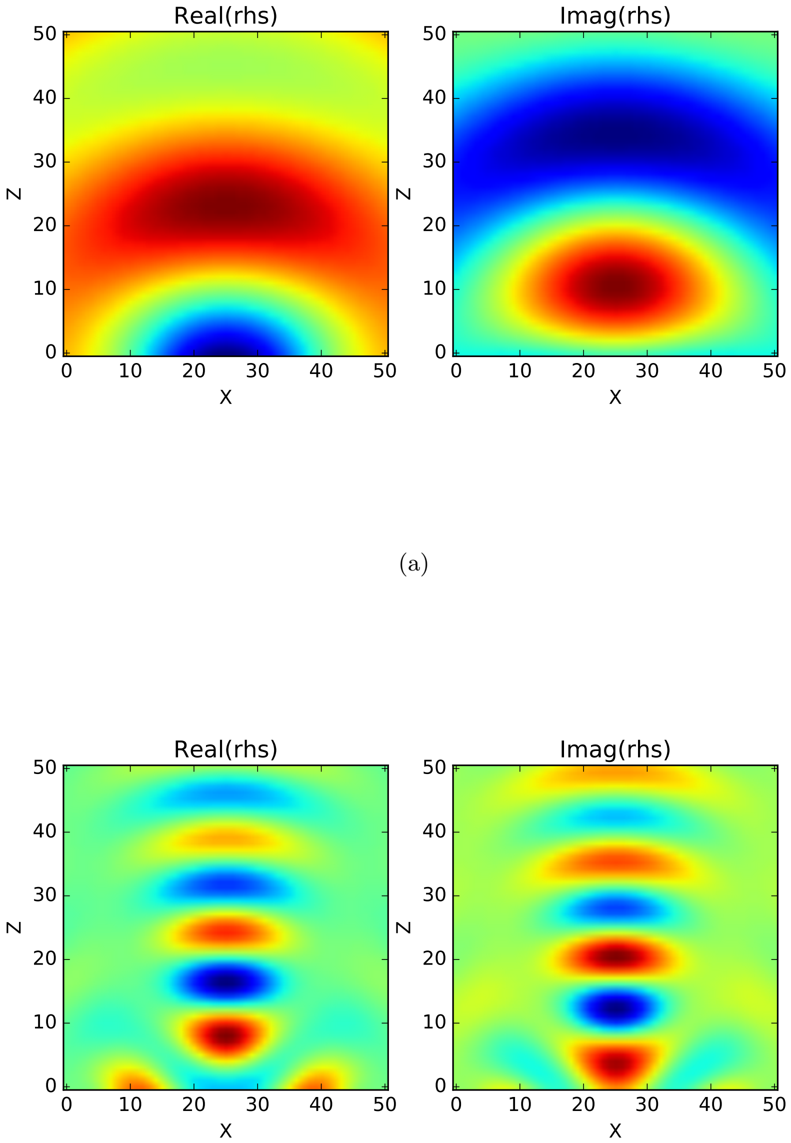
(a)

(b)

(c)

(d)

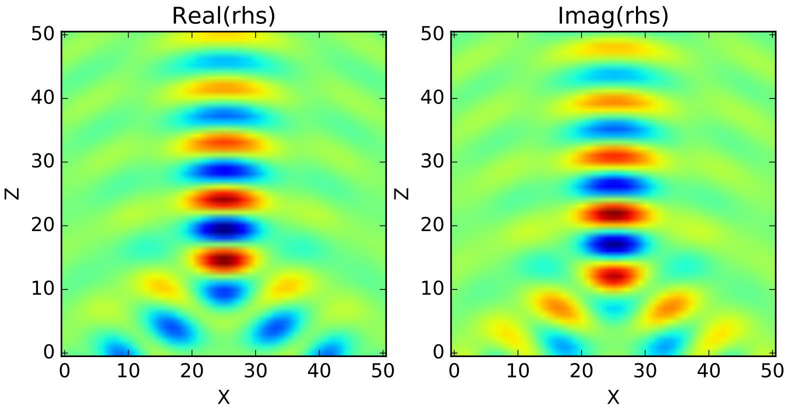
Figure 2: Zero-lag adjoint extended images for different values of N_ω : a) $N_\omega = 2$, b) $N_\omega = 4$, c) $N_\omega = 8$, d) $N_\omega = 16$. [**ER**]



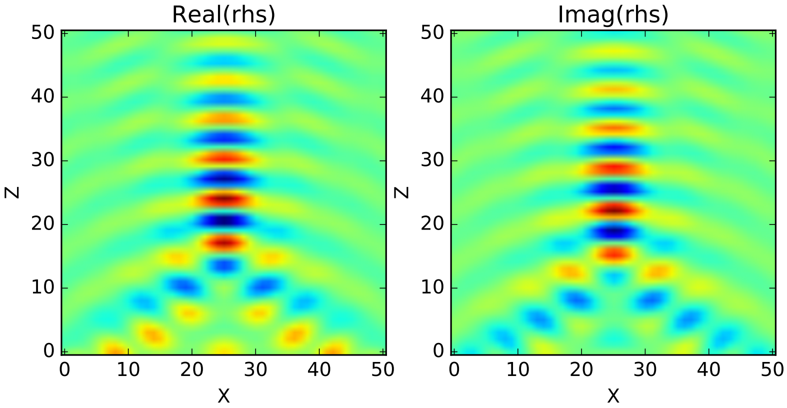
(a)

(b)

Figure 3: Adjoint extended images for the different frequencies for $N_\omega = 8:$ a) $f = f_{\max}/8$, b) $f = 3f_{\max}/8$. [**ER**]



(a)



(b)

Figure 4: Adjoint extended images for the different frequencies for $N_\omega = 8:$ a) $f = 5f_{\max}/8$, b) $f = 7f_{\max}/8$. [**ER**]

Time taken for Helmholtz solves as a function of model size

The final illustration compares the run times needed for factorizing the 2D Helmholtz matrix and the time needed for solving the resulting linear system for different model sizes. This is an important numerical test as it essentially captures the most important aspect of the feasibility of frequency domain TFWI from a computational complexity point of view. The results are shown in Figures 5, 6 and 7, where we have plotted the same figures with and without log scales for the different axes. The plotted values have been averaged over 10 factorizations to compute the time needed for the LU factorization, and averaged over 50 linear solves to compute the time needed for the solves. The graphs show us that the factorization times and the solve times scale approximately linearly with the number of grid points in the model in the log-log scale.

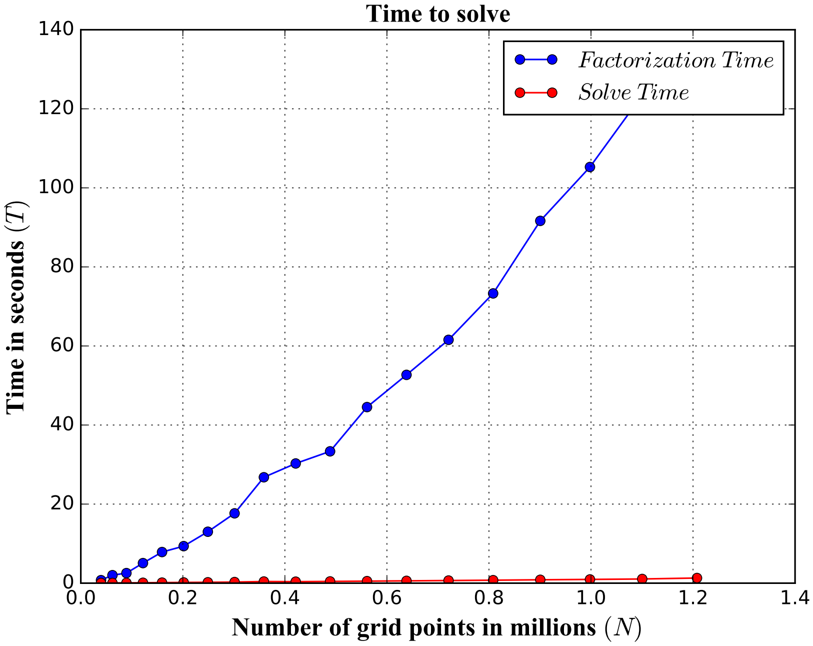


Figure 5: Plot of time taken versus the number of grid points in the model, for performing LU factorization and solution of linear systems using the factorized matrices. [**CR**]

CONCLUSIONS AND FUTURE WORK

We have developed the theory for solving the inner-loop TFWI problem in the frequency domain in this paper, and experimented with solving the normal equations using the conjugate gradient algorithm. The resulting algorithm is much faster than its time domain counterpart in 2D. The LU factorization of the Helmholtz matrix for each frequency needs to be performed only once for the whole problem, and the results are cached. Moreover the same factorization works for all the shots in the survey. It should be noted that the stated matrix factorization is feasible in 2D; however in 3D for higher frequencies and with big model sizes, the factorization of the Helmholtz matrix becomes prohibitively expensive, and moreover the resulting matrix fill-in can be significant to erode away all computational

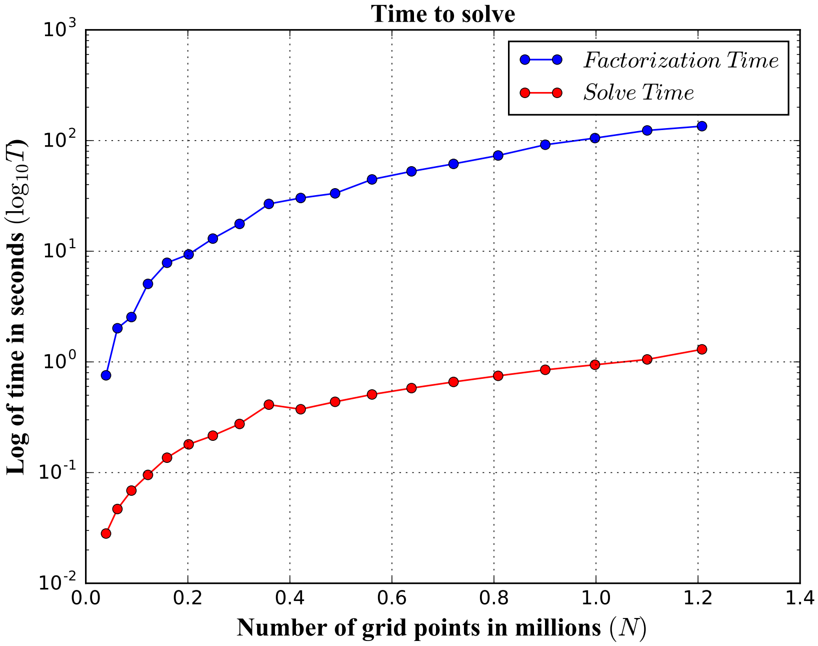


Figure 6: Plot of time taken in logarithmic scale versus the number of grid points in the model, for performing the LU factorization and solution of linear systems using the factorized matrices. [**CR**]

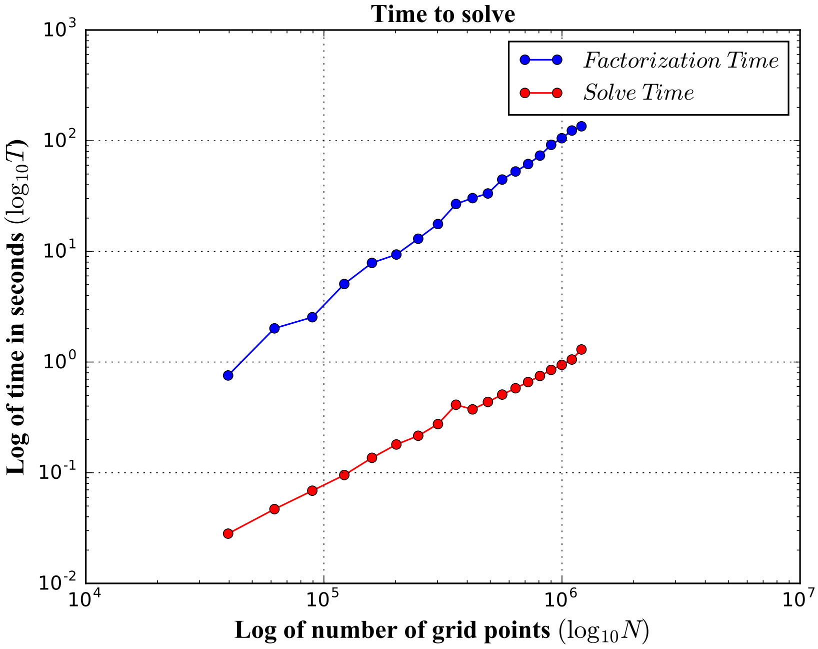


Figure 7: Plot of time taken versus the number of grid points in the model (both in logarithmic scale), for performing LU factorization and solution of linear systems using the factorized matrices in logarithmic scale. [**CR**]

gains. Thus the proposed direct method is only expected to be useful for low frequency inversions and with reasonable model sizes.

The conjugate gradient algorithm used to solve the system requires matrix vector products per iteration. We have shown in the paper that this is easily calculated using the block tridiagonal form of the matrix. Forming the matrix vector products involves some trivial multiplication by diagonal matrices, and two sparse linear solves using the cached LU factors per frequency band per shot. However, performing Gaussian elimination using sparse LU factorization is only proportional to the number of non-zero elements in the LU factors. In 2D this leads to very high savings in computational cost compared to time domain implementations of the inner-loop problem.

We are also currently exploring other linear algebra algorithms that do not involve solving the normal equations. This is in part motivated by the observation that the normal equations have squared condition number compared to the original matrix. In literature there exist methods to solve such problems that do not involve dealing with the normal equations directly. Whether these methods will be useful in our case remains to be investigated.

Irrespective of the methods used to solve the inner-loop subproblem, the next steps involve performing the non-linear update of the background velocity model and eventually extend the method to handle the 3D case. It will be interesting to investigate if frequency domain methods can provide computational savings for the non-linear update as well.

ACKNOWLEDGMENTS

The first author would like to thank Prof. Lexing Ying and Fei Liu for helping him understand several numerical aspects of the solution to the Helmholtz equation. The first author would also like to thank Prof. András Vasy for many illuminating discussions on the solvability theory of elliptic and hyperbolic partial differential equations.

REFERENCES

Barnier, G., E. Biondi, and B. Biondi, 2017, Modified Tomographic Full Waveform Inversion using the variable projection method: SEP-Report, **170**, 97–110.

Biondi, B. and A. Almomin, 2014, Simultaneous inversion of full data bandwidth by tomographic full waveform inversion: Geophysics, **79**, no. 3, WA129–WA140.

Hestenes, M. R. and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: **49**.

Huang, G., W. Symes, R. Nammour, et al., 2016, Matched source waveform inversion: Volume extension: Presented at the 2016 SEG International Exposition and Annual Meeting.

Huang, G. and W. W. Symes, 2015, Full waveform inversion via matched source extension, *in* SEG Technical Program Expanded Abstracts 2015, 1320–1325, Society of Exploration Geophysicists.

Johnson, S. G., 2008, Notes on perfectly matched layers (PMLs): Lecture notes, Massachusetts Institute of Technology, Massachusetts, **29**.

Liu, F. and L. Ying, 2016, Additive sweeping preconditioner for the Helmholtz equation: Multiscale Modeling & Simulation, **14**, 799–822.

Rickett, J. E. and P. C. Sava, 2002, Offset and angle-domain common image-point gathers for shot-profile migration: Geophysics, **67**, 883–889.

Sarkar, R. and B. Biondi, 2017, A 2D Helmholtz equation solver library based on C++ and SuiteSparse: SEP-Report, **170**, 157–174.

Sava, P. C. and S. Fomel, 2003, Angle-domain common-image gathers by wavefield continuation methods: Geophysics, **68**, 1065–1074.

Sommerfeld, A., 1949, Partial differential equations in physics: Academic Press.

APPENDIX A

Proofs

Theorem 1. *Let* $\mathbf{x}, \mathbf{b} \in \mathbb{C}^N$*, and let* $\mathbf{A} \in \mathbb{C}^{N \times N}$ *be a Hermitian positive definite matrix. Consider the following* C^∞ *function* $f : \mathbb{C}^N \to \mathbb{C}$ *given by* $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^H \mathbf{A}\mathbf{x} - \frac{1}{2}(\mathbf{b}^H\mathbf{x} + \mathbf{x}^H\mathbf{b})$*.*

Then we have the following :

a) The map f *is real valued, and so is in fact* $f : \mathbb{C}^N \to \mathbb{R}$*.*
b) The matrix $\mathbf{H} = \begin{bmatrix} \mathbf{A_R} & -\mathbf{A_I} \\ \mathbf{A_I} & \mathbf{A_R} \end{bmatrix}$ *is symmetric positive definite, where* $\mathbf{A} = \mathbf{A_R} + i\mathbf{A_I}$*, with* $\mathbf{A_R}, \mathbf{A_I} \in \mathbb{R}^{N \times N}$*.*
c) f *has a unique stationary point* $\mathbf{x}^*$ *which is also the global minimum of the function, given by the solution to the equation* $\mathbf{A}\mathbf{x}^* = \mathbf{b}$*.*

Proof. As $\mathbf{A}$ is Hermitian we have $\mathbf{A} = \mathbf{A}^H$. Thus $f(\mathbf{x})^H = \overline{f(\mathbf{x})} = f(\mathbf{x})$, proving that f is a real valued function. For the remaining parts of the proof, we will need to decompose $\mathbf{x}$ and $\mathbf{b}$ into their real and imaginary parts. Let $\mathbf{x} = \mathbf{x}_R + i\mathbf{x}_I$, and $\mathbf{b} = \mathbf{b}_R + i\mathbf{b}_I$, where $\mathbf{x}_R, \mathbf{x}_I, \mathbf{b}_R, \mathbf{b}_I \in \mathbb{R}^N$. Let us also define the vectors $\mathbf{z}, \mathbf{y} \in \mathbb{R}^{2N}$ defined as $\mathbf{z} = [\mathbf{x}_R^T \;\; \mathbf{x}_I^T]^T$, and $\mathbf{y} = [\mathbf{b}_R^T \;\; \mathbf{b}_I^T]^T$.

We first observe that as $\mathbf{A} = \mathbf{A}^H$, we immediately have that $\mathbf{A_R} = \mathbf{A_R}^T$ and $\mathbf{A_I} = -\mathbf{A_I}^T$, which implies that $\mathbf{A_R}$ is symmetric and $\mathbf{A_I}$ is skew-symmetric. This then implies that $\mathbf{H}$ is symmetric. The skew-symmetry of $\mathbf{A_I}$ also implies that for any $\mathbf{u} \in \mathbb{R}^N$, we have $\mathbf{u}^T\mathbf{A_I}\mathbf{u} = 0$. Using this fact, a direct computation gives us

$$\begin{aligned} \mathbf{x}^H\mathbf{A}\mathbf{x} &= \mathbf{x}_R^T\mathbf{A_R}\mathbf{x}_R + \mathbf{x}_I^T\mathbf{A_R}\mathbf{x}_I - \mathbf{x}_R^T\mathbf{A_I}\mathbf{x}_I + \mathbf{x}_I^T\mathbf{A_I}\mathbf{x}_R \\ &= \begin{bmatrix} \mathbf{x}_R^T & \mathbf{x}_I^T \end{bmatrix} \begin{bmatrix} \mathbf{A_R} & -\mathbf{A_I} \\ \mathbf{A_I} & \mathbf{A_R} \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_I \end{bmatrix} \\ &= \mathbf{z}^T\mathbf{H}\mathbf{z} \;\; , \end{aligned} \tag{A-1}$$

and thus as $\mathbf{A}$ is positive definite, we also have that $\mathbf{H}$ is positive definite which proves the second part of the theorem.

For the final part of the theorem, another explicit computation gives us

$$\mathbf{b}^H\mathbf{x} + \mathbf{x}^H\mathbf{b} = 2\;(\mathbf{b}_R^T\mathbf{x}_R + \mathbf{b}_I^T\mathbf{x}_I) = 2\mathbf{y}^T\mathbf{z} \;\; , \tag{A-2}$$

and so now treating f as a function of $\mathbf{z}$ given by the expression $f(\mathbf{z}) = \frac{1}{2}\mathbf{z}^T\mathbf{H}\mathbf{z} - \mathbf{y}^T\mathbf{z}$, the goal of minimizing $f(\mathbf{x})$ over $\mathbf{x}$ is equivalent to minimizing $f(\mathbf{z})$ over $\mathbf{z}$. We next note that the Hessian of f with respect to $\mathbf{z}$ is given by $(\nabla^2 f)(\mathbf{z}) = \mathbf{H} \succ 0$, which implies that f is convex in $\mathbf{z}$ (and hence also in $\mathbf{x}$), and also implies that f has a unique global minimizer $\mathbf{z}^*$ that satisfies $\mathbf{H}\mathbf{z}^* = \mathbf{y}$. If $\mathbf{z}^* = [\mathbf{x}_R^{*T} \;\; \mathbf{x}_I^{*T}]^T$, then the corresponding unique global minimizer of f in $\mathbf{x}$ is given by $\mathbf{x}^* = \mathbf{x}_R^* + i\mathbf{x}_I^*$, and stationarity of f at $\mathbf{x}^*$ follows by observing that $f(\mathbf{x})$ is holomorphic in $\mathbf{x}$. It thus only remains to be shown that $\mathbf{A}\mathbf{x}^* = \mathbf{b}$.

Using the relationship $\mathbf{H}\mathbf{z}^* = \mathbf{y}$, the component equations are $\mathbf{A_R}\mathbf{x}_R^* - \mathbf{A_I}\mathbf{x}_I^* = \mathbf{b}_R$, and $\mathbf{A_I}\mathbf{x}_R^* + \mathbf{A_R}\mathbf{x}_I^* = \mathbf{b}_I$. Multiplying the second equation by i and adding to the first equation gives

$$(\mathbf{A_R} + i\mathbf{A_I})\mathbf{x}_R^* + i(\mathbf{A_R} + i\mathbf{A_I})\mathbf{x}_I^* = \mathbf{b}_R + i\mathbf{b}_I \;\; , \tag{A-3}$$

which upon regrouping gives $\mathbf{A}\mathbf{x}^* = \mathbf{b}$, completing the proof of the theorem. □

APPENDIX B

Block structure of M

We give a short derivation of the explicit form of the matrix $\mathbf{M}$, that appears in the normal equation (19). To achieve this, we will first express each term individually in block diagonal form that appears in the expression $\mathbf{M} = \sum_{i=1}^{N_s} \mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i + \epsilon^2 \mathbf{D}^H \mathbf{D} + \gamma^2 \mathbf{I}$, and then aggregate the results. As the term $\gamma^2 \mathbf{I}$ is trivial, we will focus on the other two terms $\sum_{i=1}^{N_s} \mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i$ and $\epsilon^2 \mathbf{D}^H \mathbf{D}$.

As mentioned previously, the derivative operator $\mathbf{D}$ that we use in this paper is the forward difference operator, that is we are using the approximation $\frac{\partial \delta \hat{\mathbf{c}}_k}{\partial \omega} \approx (\delta \hat{\mathbf{c}}_{k+1} - \delta \hat{\mathbf{c}}_k)/\Delta\omega$, for all $k = 1, \ldots, N_\omega - 1$. This leads to the following block diagonal structure for $\mathbf{D}^H \mathbf{D}$

$$\mathbf{D}^H \mathbf{D} = \frac{1}{\Delta\omega^2} \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \ldots & \ldots & 0 \\ -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} & \ddots & \vdots \\ \vdots & -\mathbf{I} & \ddots & -\mathbf{I} & \vdots \\ \vdots & \ddots & -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} \\ 0 & \ldots & \ldots & -\mathbf{I} & \mathbf{I} \end{bmatrix} . \tag{A-4}$$

The matrix $\mathbf{D}^H \mathbf{D}$ has N_ω row blocks and N_ω column blocks, and in addition each $\mathbf{I}$ appearing in (A-4) is an identity matrix of dimensions $N \times N$.

In order to study the term $\sum_{i=1}^{N_s} \mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i$, we first fix a shot i. Then using (17), the block diagonal structure of $\mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i$ is completely determined once we have a block diagonal representation of the matrix $\mathbf{R}_i^H \mathbf{R}_i$. The first trivial observation is that the sampling operator (i.e. the set of receivers) stays the same across all the frequencies. Let us denote this operator as $\mathbf{\Gamma}_i$, which is a $\{0, 1\}$ matrix of dimensions $N_{r_i} \times N$. Then we can write the following expression for $\mathbf{R}_i$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{\Gamma}_i & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \mathbf{\Gamma}_i \end{bmatrix} . \tag{A-5}$$

We assume that no receiver is sampled twice, which implies that $\mathbf{\Gamma}_i^H \mathbf{\Gamma}_i$ is a diagonal matrix with 1s at the diagonal entries corresponding to the receiver locations, and 0s at the diagonal entries corresponding to the locations where there are no receivers present. This in turn implies that $\mathbf{R}_i^H \mathbf{R}_i$ is a $\{0, 1\}$ diagonal matrix given by

$$\mathbf{R}_i^H \mathbf{R}_i = \begin{bmatrix} \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \end{bmatrix} , \tag{A-6}$$

and so using equation (17) we can write

$$\mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i = \begin{bmatrix} (\mathbf{A}_1^{-1} \mathbf{\Lambda}_{1i})^H \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{A}_1^{-1} \mathbf{\Lambda}_{1i} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & (\mathbf{A}_{N_\omega}^{-1} \mathbf{\Lambda}_{N_\omega i})^H \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{A}_{N_\omega}^{-1} \mathbf{\Lambda}_{N_\omega i} \end{bmatrix} . \tag{A-7}$$

It is easy to see from equation (A-7) that the matrix $\mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i$ is block diagonal, and so $\sum_{i=1}^{N_s} \mathbf{B}_i^H \mathbf{R}_i^H \mathbf{R}_i \mathbf{B}_i$ is also block diagonal. Reassembling all the terms and absorbing the factor of $1/\Delta\omega$ into the constant ϵ we can finally write the expression for $\mathbf{M}$ in block matrix form as

$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix}
(\epsilon^2+\gamma^2)\mathbf{I} & -\epsilon^2\mathbf{I} & \dots & \dots & 0 \\
-\epsilon^2\mathbf{I} & (2\epsilon^2+\gamma^2)\mathbf{I} & -\epsilon^2\mathbf{I} & \ddots & \vdots \\
\vdots & -\epsilon^2\mathbf{I} & \ddots & -\epsilon^2\mathbf{I} & \vdots \\
\vdots & \ddots & -\epsilon^2\mathbf{I} & (2\epsilon^2+\gamma^2)\mathbf{I} & -\epsilon^2\mathbf{I} \\
0 & \dots & \dots & -\epsilon^2\mathbf{I} & (\epsilon^2+\gamma^2)\mathbf{I}
\end{bmatrix} \\
&+ \begin{bmatrix}
\sum\limits_{i=1}^{N_s} (\mathbf{A}_1^{-1}\mathbf{\Lambda}_{1i})^H \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{A}_1^{-1}\mathbf{\Lambda}_{1i} & \dots & 0 \\
\vdots & \ddots & \vdots \\
0 & \dots & \sum\limits_{i=1}^{N_s} (\mathbf{A}_{N_\omega}^{-1}\mathbf{\Lambda}_{N_\omega i})^H \mathbf{\Gamma}_i^H \mathbf{\Gamma}_i \mathbf{A}_{N_\omega}^{-1}\mathbf{\Lambda}_{N_\omega i}
\end{bmatrix} \qquad \text{(A-8)} \\
&= \mathbf{M}_1 + \mathbf{M}_2 \ ,
\end{aligned}$$

where $\mathbf{M}_1$ and $\mathbf{M}_2$ are the first and second matrices in the above expression. The matrix $\mathbf{M}_1$ is block tridiagonal, and the matrix $\mathbf{M}_2$ is block diagonal, and hence $\mathbf{M}$ is also block tridiagonal.