Received: 22 May 2017 | Revised: 21 March 2018 | Accepted: 21 June 2018

DOI: 10.1002/nla.2206

SPECIAL ISSUE PAPER

WILEY

A multigrid solver to the Helmholtz equation with a point source based on travel time and amplitude

Eran Treister[1] | Eldad Haber[2]

[1]Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

[2]Department of Earth and Ocean Sciences, The University of British Columbia, Vancouver, British Columbia, Canada

Correspondence
Eran Treister, Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel.
Email: erant@cs.bgu.ac.il

Funding information
European Union's Seventh Framework Programme (FP7/2007-2013), Grant/Award Number: 623212

Summary

The Helmholtz equation arises when modeling wave propagation in the frequency domain. The equation is discretized as an indefinite linear system, which is difficult to solve at high wave numbers. In many applications, the solution of the Helmholtz equation is required for a point source. In this case, it is possible to reformulate the equation as two separate equations: one for the travel time of the wave and one for its amplitude. The travel time is obtained by a solution of the factored eikonal equation, and the amplitude is obtained by solving a complex-valued advection–diffusion–reaction equation. The reformulated equation is equivalent to the original Helmholtz equation, and the differences between the numerical solutions of these equations arise only from discretization errors. We develop an efficient multigrid solver for obtaining the amplitude given the travel time, which can be efficiently computed. This approach is advantageous because the amplitude is typically smooth in this case and, hence, more suitable for multigrid solvers than the standard Helmholtz discretization. We demonstrate that our second-order advection–diffusion–reaction discretization is more accurate than the standard second-order discretization at high wave numbers, as long as there are no reflections or caustics. Moreover, we show that using our approach, the problem can be solved more efficiently than using the common shifted Laplacian multigrid approach.

KEYWORDS

factored eikonal equation, fast marching, Helmholtz equation, multigrid, seismic modeling, shifted Laplacian

1 | INTRODUCTION

The acoustic Helmholtz equation is used to model the propagation of a wave within a heterogeneous medium. Assuming constant density, the equation is given by

$$\Delta u + \omega^2\kappa^2(\vec{x})u = q(\vec{x}), \quad \vec{x} \in \Omega, \tag{1}$$

where $u(\vec{x})$ is the pressure wave function in the frequency domain, $\omega = 2\pi f$ is the angular frequency, and $\kappa(\vec{x})$ is the "slowness" of the medium—the inverse of its velocity. The right-hand side $q(\vec{x})$ is used to incorporate sources into the equation. In this work, we consider the case where $q(\vec{x}) = \delta(\vec{x} - \vec{x}_0)$, which models a point source at location x_0 ($\delta(\cdot)$ is the Dirac delta function). The Helmholtz equation with a point source is common in geophysical applications, for example, seismic modeling and full-waveform inversion (FWI).[1–8]

The Helmholtz equation is accompanied with boundary conditions (BCs), which can be Neumann or Dirichlet for example. In many cases, the equation is involved with absorbing BCs that mimic the propagation of a wave in an open domain. One option for this is the Sommerfeld BC

$$\mathbf{n} \cdot \nabla u - i\omega\kappa u = 0. \tag{2}$$

A more effective way to absorb the waves is by using a boundary layer. This can be achieved by either a perfectly matched layer[9] or an absorbing boundary layer.[10–12] To implement the latter layer, for example, we add an attenuation term to Equation (1), as follows:

$$\Delta u + \omega^2\kappa^2(\vec{x})u - i\omega\gamma\kappa^2(\vec{x})u = q(\vec{x}), \quad \vec{x} \in \Omega,$$

where $\gamma(\vec{x}) \geq 0$ is a function that quadratically goes from 0 to ω toward the boundaries of the domain, which attenuates the waveform toward these boundaries.[12] The thickness of this layer is usually chosen to be about one wavelength. The same γ parameter can be used to impose attenuation all over the domain but is quite small in most realistic scenarios. To ease the derivations in this paper, we henceforth ignore the boundary layer and attenuation, that is, assume that $\gamma = 0$, and focus on Equation (1).

We are mostly interested in problems where the frequency ω (or the wave number $\kappa\omega$) is high. In this case, the resulting linear system that arises from the discretization of (1) is highly indefinite. While 2D solutions can be obtained using direct methods, solving the discretized equation in 3D is challenging. That is because the discretization of the problem requires a very fine mesh and a large number of unknowns, in addition to the indefiniteness of the associated matrix.[13–15]

In recent years, there has been a great effort to develop efficient solvers for systems arising from (1), using several different approaches to tackle the problem. One of the most common approaches is the shifted Laplacian multigrid preconditioner,[12,13,16–22] which modifies the equation by adding complex values to the diagonal of the matrix. The modified system is then solved using a multigrid method and is used as a preconditioner for the nonshifted system. This preconditioner is used with a proper Krylov method to obtain the solution of the nonshifted problem (usually, fGMRES[23] or BiCGSTAB[24] are used). Another approach was recently proposed in the works of Poulson et al.[14] and Engquist and Ying[25] for solving (1) in 2D and 3D, respectively. The approach can be viewed as a domain decomposition method with particular BCs and can be effective in terms of iterations but requires a large setup time and storage. This can impose challenges if the solution of (1) is required for multiple frequencies. Other iterative approaches include those in the works of Gordon et al.[26,27] as well as Haber and Maclachlan,[15] Brandt and Livshits,[28] Olson and Schroder,[29] and Livshits,[30] which are multigrid based.

In this paper, we develop a new approach for solving the Helmholtz equation based on the work of Haber and Maclachlan.[15] Rather than solve the discrete (1), we reformulate the problem by using the Rytov decomposition of the solution[*]

$$u(\vec{x}) = a(\vec{x}) \exp(-i\omega\tau(\vec{x})). \tag{3}$$

Here, the waveform $u(\vec{x})$ is decomposed into an amplitude $a(\vec{x})$ and a phase involving $\tau(\vec{x})$. To reformulate (1) according to (3), we first use the chain rule to obtain

$$\begin{aligned} \nabla u &= (\nabla a - i\omega\nabla\tau)\exp(-i\omega\tau), \\ \Delta u &= (\Delta a - 2i\omega\nabla\tau \cdot \nabla a - i\omega\Delta\tau - \omega^2 a|\nabla\tau|^2)\exp(i\omega\tau). \end{aligned}$$

Then, after plugging Δu into (1) and multiplying the equation by $\exp(i\omega\tau)$, we get the following complex-valued advection–diffusion–reaction (ADR) equation:

$$\Delta a - 2i\omega\nabla\tau \cdot \nabla a - i\omega(\Delta\tau)a - \omega^2(|\nabla\tau|^2 - \kappa(\vec{x})^2)a = \hat{q}(\vec{x}), \tag{4}$$

where $\hat{q}(\vec{x}) = q(\vec{x})\exp(i\omega\tau)$. This equation is also equivalent to

$$\Delta a - i\omega\nabla\tau \cdot \nabla a - i\omega\nabla \cdot (\nabla\tau a) - \omega^2(|\nabla\tau|^2 - \kappa(\vec{x})^2)a = \hat{q}(\vec{x}), \tag{5}$$

which does not contain the term $\Delta\tau$.

We note that the function $a(\vec{x})$ does not fully correspond to a real amplitude, as usually assumed when using the Rytov decomposition. For example, $a(\vec{x})$ is not necessarily positive or even real valued. In the process above, we artificially

[*] The Rytov decomposition is usually given by $u(\vec{x}) = a(\vec{x})\exp(i\omega\tau(\vec{x}))$. Here, we use its complex conjugate, which is equivalent to using the standard decomposition.

doubled the unknowns of Equation (1) using the Rytov decomposition. Hence, for any given $\tau(\vec{x})$ and frequency ω, the function $a(\vec{x})$ is unique according to (3), and the resulting Equations (4) and (5) are equivalent to (1). Their numerical solutions using (3) are equivalent to the numerical solution of (1) up to discretization and roundoff errors *only*, given equivalent BCs (to be discussed later). For example, if we choose $\tau = 0$, then we retain Equation (1), and $a(\vec{x}) = u(\vec{x})$. The work of Haber and Maclachlan[15] aims to get a multigrid preconditioner for (1) and chooses $\tau(\vec{x})$ to be a plane, so that problem (4) becomes positive definite and easy to solve. In this work, we suggest new discretizations for the Helmholtz problem based on (4) or (5). Our first goal is to choose $\tau(\vec{x})$ such that the amplitude $a(\vec{x})$ is as smooth as possible, so we get small numerical errors when we discretize (4) or (5). Our second goal is to get a linear system that can be solved efficiently by multigrid methods.

To fulfill our first goal, we aim to capture most of the oscillatory behavior of the solution $u(\vec{x})$ by choosing $\tau(\vec{x})$ appropriately, so that the corresponding amplitude $a(\vec{x})$ is smooth. As motivation, consider the solution of (1) with a point source for a constant medium $\kappa = 1$. In 3D, the analytical solution is $u(\vec{x}) = \frac{1}{4\pi r}\exp(-i\omega r)$, where $r = \|\vec{x} - \vec{x}_0\|_2$ is the Euclidean distance from the source $\vec{x}_0$. In this case, if we choose $\tau(\vec{x}) = r$, we get $a(\vec{x}) = \frac{1}{4\pi r}$, which is a rather smooth function (except at x_0). Figure 1 demonstrates this case in 2D. For a heterogeneous medium, a similar result can be obtained by solving the eikonal equation

$$|\nabla\tau|^2 = \kappa(\vec{x})^2, \tag{6}$$

where $|\cdot|$ is the Euclidean norm. This is an advection equation that requires a known initial value at some subregion, and in the context of wave propagation, the solution τ has the meaning of travel time, or first-arrival time, of the wave. In this work, we consider the wave propagation from a point source at location $\vec{x}_0$, for which the travel time is 0, and hence, $\tau(\vec{x}_0) = 0$. By choosing τ according to (6), we eliminate the last term on the left-hand side of (4).

The idea of using travel time and amplitude for modeling wave propagation from a point source was studied recently in a different approach than in this paper. The Rytov decomposition (3) is used to get a geometrical-optics $O\left(\frac{1}{\omega}\right)$ ansatz for the solution of the Helmholtz equation in the high-frequency regime.[31–34] There, similarly to our approach, the eikonal equation is used to eliminate the ω^2 term in (4) yielding the travel time, and the amplitude is obtained by eliminating the ω term in (4), by solving the real-valued transport equation

$$\nabla\tau\cdot\nabla a + \frac{1}{2}(\Delta\tau)a = \frac{1}{2}(\nabla\tau\cdot\nabla a + \nabla\cdot((\nabla\tau)a)) = 0. \tag{7}$$

The resulting approximation includes only the first-arrival information of the wave propagation, and since (7) does not depend on ω, this approximation aims to be valid for multiple frequencies. Our approach is different in the way that the amplitude is defined, according to (4) instead of (7). This way, the amplitude is specific for a given frequency and contains all the information of the wave propagation, for example, it includes reflections and interferences.

To compute the phase for (3), the eikonal equation is solved. This equation is nonlinear and, hence, may have many solutions. Out of these solutions, the one that corresponds to the first-arrival time can be computed efficiently.[35,36] One of the most effective ways to compute it is by fast marching methods,[37–39] which solve (6) directly using first- or second-order schemes in $\mathcal{O}(n\log n)$ operations, based on the monotonicity of the solution along the characteristics. Alternatively, (6)

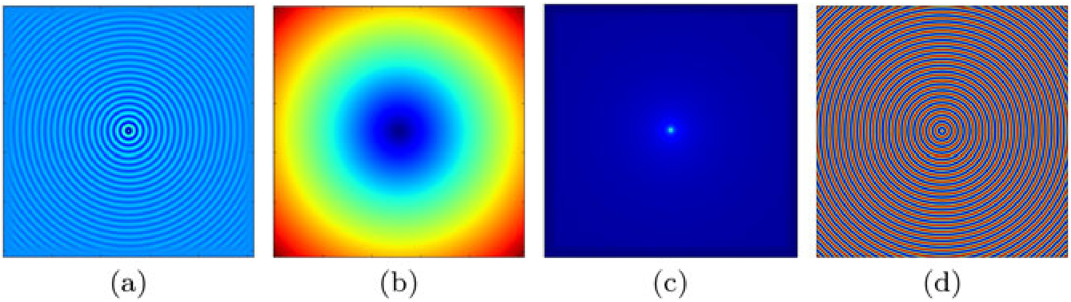

FIGURE 1 The solution of the Helmholtz equation with homogeneous media for a point source with absorbing BCs. The oscillatory waveform is given by $u = a\exp(-i\omega\tau)$, whereas the $a(\vec{x})$ and $\tau(\vec{x})$ that are calculated numerically are smooth. (a) The waveform $u(\vec{x})$, (b) The travel time $\tau(\vec{x})$, (c) The amplitude $a(\vec{x})$, (d) The phase $\exp(-i\omega\tau)$

can be solved iteratively by fast sweeping methods with first- or higher-order accuracy.[40–42] Equation (6) can also be solved using a Lax–Friedrichs scheme,[43,44] which involves adding artificial viscosity to (6).

However, the methods mentioned above are not suitable for solving (6) for our purpose. To get an accurate solution for the amplitude in (7), the numerical approximation for $\tau(\vec{x})$ has to be very accurate.[33] Because the analytical $\tau(\vec{x})$ is nonsmooth at the point source, the numerical solution of Equation (6) is polluted with errors when it is computed using the aforementioned standard finite difference methods.[45] To overcome this,[32,33] use the factored version of the eikonal equation, which was originally suggested in the work of Pica.[46] The new equation is obtained by setting $\tau = \tau_0\tau_1$ in (6), where the function $\tau_0(\vec{x})$ is known and its derivatives are computed analytically. Using the chain rule, we get the factored eikonal equation for τ_1 as

$$|\tau_0\nabla\tau_1 + \tau_1\nabla\tau_0|^2 = \kappa(\vec{x})^2. \tag{8}$$

The most common choice for τ_0 is the distance function from the point source, that is, $\tau_0 = \|\vec{x} - \vec{x}_0\|_2$. That is the analytical solution for (6) in the case where $\kappa(x) = 1$. The function τ_0 is nonsmooth at the location of the source, but the factor τ_1, which needs to be computed numerically, is expected to be very smooth at the surrounding of the source. Similarly to its original version, Equation (8) can be solved directly by the fast marching method[47] or iteratively by the fast sweeping methods with first-order accuracy[45,48,49] or by a Lax–Friedrichs scheme up to third-order accuracy.[32,33,49] The works of Luo et al.[49] and Noble et al.[50] suggest hybrid schemes where the factored eikonal equation is solved at the neighborhood of the source, and the standard eikonal equation is solved in the rest of the domain.

Similarly to the work of Luo et al.,[33] in this work, we use the factored eikonal Equation (8) to get an accurate solution for the Helmholtz equation based on (4) and (5). We apply the fast marching method suggested in the work of Treister and Haber[47] for solving (8), but in principle, all the methods mentioned earlier for solving the factored eikonal equation can be used in our approach. Note that the equations for the amplitude that include $\Delta\tau$ now include $\Delta\tau_1$ in the factored case. That is a numerical approximation of the Laplacian of the factor τ_1, which may be nonsmooth in areas away from the source due to discontinuities in κ or due to caustics. $\nabla\tau_1$ is not continuous but is bounded because of (8). Approximating the second derivative of a nonsmooth function may yield high numerical errors. For this reason, the third-order Lax–Friedrichs scheme, which adds smoothness to τ_1, is used in the works of Luo and Qian[32] and Luo et al.[33] However, both Equations (7) and (5) show that the problems can be formulated without $\Delta\tau_1$, and there is no real need to approximate it numerically.

To summarize, in this work, we reformulate the Helmholtz equation using ideas from the works of Haber and Maclachlan,[15] Luo and Qian,[32] Luo et al.,[33] and Treister and Haber[47] to allow a more efficient numerical solution of the equation for a point source. We aim that the majority of the solution is represented by smooth functions, which have physical meanings of amplitude and travel time. Similarly to the work of Haber and Maclachlan,[15] we use the full ADR Equation (4) or (5) for the amplitude and solve it by multigrid methods. However, instead of choosing the travel time τ as a plane, we solve (8) to find it, so that the amplitude $a(x)$ is smooth. The smoothness of $a(x)$ contributes to the efficiency of multigrid methods for solving (4), as smooth functions can be approximated well on coarser grids. The rest of this paper is organized as follows. In Section 2, we present the options that we consider for discretizing the ADR equation and examine the similarity between the discretized Helmholtz and ADR equations. Then, in Section 3, we present the multigrid methods that we use to solve the ADR and Helmholtz equations. Finally, in Section 4, we present numerical results, which first compare the accuracy of the two approaches for the Helmholtz problem and then the computational effort needed to solve the equations using multigrid.

2 | THE DISCRETIZATION OF THE REFORMULATED ADR EQUATION

The Helmholtz Equation (1) is usually discretized using the finite difference method on a regular mesh. The standard approach involves a second-order scheme, resulting in 5- and 7-point stencils for two and three dimensions, respectively. We denote the corresponding linear system by

$$H\mathbf{u} = \mathbf{q}. \tag{9}$$

We note that using this discretization, one has to use a mesh that is fine enough, having at least 10–15 grid points per wavelength[12–15]; otherwise, the numerical solution is polluted by dispersion errors. This requirement leads to rather large matrices, and therefore, higher-order discretizations were developed to minimize the numerical dispersion phenomenon and require fewer grid points per wavelength on the expense of larger stencils.[51–54] In this work, we discretize (4) and (5) using second-order stencils as in the work of Haber and Maclachlan,[15] but note that extensions to higher-order stencils can be obtained as well. In particular, we consider two types of discretizations for the ADR equations, which are different in the scheme used for the advection operator in (4). The first discretization includes a central difference scheme for the advection operator. We denote the resulting linear system

$$\hat{H}_{\text{cen}}\mathbf{a} = \hat{\mathbf{q}}. \tag{10}$$

In the second discretization, we use a second-order upwind scheme for the advection term and denote the resulting linear system

$$\hat{H}_{\text{2up}}\mathbf{a} = \hat{\mathbf{q}}. \tag{11}$$

We now present how we define (10) and (11) and later show a relation between the matrices H and $\hat{H}_{\text{cen}}$. We present all the finite difference operators in 1D, and extensions to 2D and 3D are straightforward.

2.1 | The coefficients of the equation

First, we define the coefficients of Equations (4) and (5), involving the travel time τ. For this, we assume that the travel time factor τ_1 is calculated by fast marching[47] in second-order accuracy on the same mesh. Using the chain rule, we define

$$\nabla\tau = \tau_0\nabla\tau_1 + \tau_1\nabla\tau_0 \quad \text{and} \quad \Delta\tau = \tau_1\Delta\tau_0 + 2\nabla\tau_0\cdot\nabla\tau_1 + \tau_0\Delta\tau_1, \tag{12}$$

where τ_0, $\nabla\tau_0$, and $\Delta\tau_0$ are the known analytic solution of (6) for a constant medium and its analytic derivatives. To approximate $\nabla\tau_1$ in (12), we use the second-order central difference operator

$$\left(\frac{\partial\tau_1}{\partial x}\right)_j \approx \frac{(\tau_1)_{j+1} - (\tau_1)_{j-1}}{2h},$$

while reverting to first-order operators on the domain boundaries. As an alternative, we may use the discrete gradients that were used to calculate τ_1 in the factored fast marching algorithm. To approximate $\Delta\tau_1$ in (12), we use the standard second-order central difference operator and use the one-sided second-order second derivative operator on the boundaries, as there are no natural BCs to τ.

2.2 | The discretization of the ADR equation

Once the coefficients of (4) and (5) are known, we discretize the equations using second-order finite difference operators. The Laplacian operator is discretized using a standard second-order central difference operator, just as in the second-order discretization of (1). As noted before, for the advection term $\nabla\tau\cdot\nabla a$, we have two options.

1. Using a second-order central difference:
$$\left(\frac{\partial\tau}{\partial x}\cdot\frac{\partial a}{\partial x}\right)_j \approx \left(\frac{\partial\tau}{\partial x}\right)_j\cdot\frac{a_{j+1}-a_{j-1}}{2h}. \tag{13}$$
This operator is used in (10).
2. Using a second-order upwind scheme:
$$\left(\frac{\partial\tau}{\partial x}\cdot\frac{\partial a}{\partial x}\right)_j \approx \begin{cases} \left(\frac{\partial\tau}{\partial x}\right)_j\cdot\frac{3a_j-4a_{j-1}+a_{j-2}}{2h}, & \text{if } \left(\frac{\partial\tau}{\partial x}\right)_j > 0 \\ \left(\frac{\partial\tau}{\partial x}\right)_j\cdot\frac{-3a_j+4a_{j+1}-a_{j+2}}{2h}, & \text{if } \left(\frac{\partial\tau}{\partial x}\right)_j < 0. \end{cases} \tag{14}$$
This operator is used in (11).

One important observation should be made regarding (14): At the immediate neighborhood of the source point (distance of one node), we cannot use the second-order upwind scheme because $\nabla\tau$ changes signs. Therefore, for those nodes, we simply use (13). This situation is similar to the treatment of the same points when calculating τ using fast marching. There, the algorithm reverts to first-order operators for these points, but still achieves second-order accuracy overall (see the work of Treister and Haber[47]). For discretizing the advection term in (5), we use similar operators.

2.2.1 | Boundary conditions

The BCs of the ADR Equation (4) are identical to those of (1) and are defined using the chain rule. For example, for the Neumann BC, we have

$$\mathbf{n}\cdot\nabla u = 0 \Rightarrow \mathbf{n}\cdot(\nabla a - i\omega\nabla\tau)\exp(-i\omega\tau) = 0 \Rightarrow \mathbf{n}\cdot\nabla a - i\omega\mathbf{n}\cdot\nabla\tau = 0. \tag{15}$$

The Sommerfeld BC in Equation (2) is treated in a similar way, and the absorbing boundary layer in γ is kept as a mass matrix similarly to the case of (1). The BC in Equation (15) or its Sommerfeld version is needed for the Laplacian operator in (4) and (5) and can also be used for the advection term in this equation. Alternatively, the advection term can be discretized without BCs. Since the source is inside the domain, the wave propagate from the source outward, and the upwind discretization at the boundary points will always use internal neighboring points in (14). Therefore, the upwind advection term in (14) does not require BCs. This principle is used in fast marching to calculate τ. On the other hand, the scheme (13) does require a different treatment at the boundaries and may be replaced with the upwind operators there (which is what we do here). Other BCs for (1) can be adapted to (4) similarly to (15).

2.3 | The relation between the discretized versions of the ADR and Helmholtz equations

Equations (1) and (4) are diagonally scaled versions of each other.[15] For the matrix in (9) and both matrices $\hat{H}$ in (10) and (11), we have that

$$M^{-1}HM \approx \hat{H}, \tag{16}$$

where M is a diagonal matrix such that $M_{jj} = \exp(-i\omega\tau(x_j))$. We now show a comparison between the discrete matrices $M^{-1}HM$ and $\hat{H}_{\text{cen}}$ in 1D, and again, the extensions to 2D and 3D are straightforward.

Assume that we know $\tau(x)$ for which (6) is held. A general equation of the system $M^{-1}HM\mathbf{a} = \hat{\mathbf{q}}$ is given by

$$\frac{1}{h^2}(a_{j-1}\exp(i\omega(\tau_j-\tau_{j-1})) - 2a_j + a_{j+1}\exp(i\omega(\tau_j-\tau_{j+1}))) + \omega^2\kappa_j^2 a_j = \hat{q}_j. \tag{17}$$

Using the Taylor expansion, we set $\tau_{j\pm1} - \tau_j \approx \pm\tau'_j h + \frac{1}{2}\tau''_j h^2 + O(h^3)$ and get

$$\frac{1}{h^2}\left(a_{j-1}\exp\left(i\omega\left(-\tau'_j h + \frac{1}{2}\tau''_j h^2 + O(h^3)\right)\right) - 2a_j\right) + \frac{1}{h^2}\left(a_{j+1}\exp\left(i\omega\left(\tau'_j h + \frac{1}{2}\tau''_j h^2 + O(h^3)\right)\right)\right) + \omega^2\kappa_j^2 a_j = \hat{q}_j. \tag{18}$$

Next, we use the Taylor expansion $\exp(\varepsilon) = 1 + \varepsilon + \frac{1}{2}\varepsilon^2 + \frac{1}{6}\varepsilon^3 + O(\varepsilon^4)$, for which we take many terms because the expression inside the exponent is of magnitude ωh, which is typically small but much larger than h. We get

$$\begin{aligned}\frac{1}{h^2}(a_{j-1} - 2a_j + a_{j+1}) &- \frac{i\omega}{h^2}\left(\left(h + \frac{1}{6}\omega^2 h^3\right)\tau'_j(a_{j+1} - a_{j-1}) + h^2\tau''_j\frac{1}{2}(a_{j+1} + a_{j-1})\right)\\ &- \frac{\omega^2}{h^2}\left(\frac{1}{2}\left(-\tau'_j h\right)^2 a_{j-1} + \frac{1}{2}\left(\tau'_j h\right)^2 a_{j+1} - \omega^2 h^2\kappa_j^2 a_j\right) = \hat{q}_j,\end{aligned} \tag{19}$$

while neglecting terms that are of magnitude lower than $O(\omega^2 h^2)$ relatively to the magnitude of the entries of H.

The first and second lines of (19) are identical to those in our discretization of (4) using (13). The second line suggests that the mass matrix that multiplies $\Delta\tau$ should be discretized with an averaging operator $\frac{1}{2}(a_{j+1} + a_{j-1})$ if we wish to make the two discretizations more similar. Looking into the third row of (19), neglecting everything except the leading $O(h^2)$ terms, we get $\left(\tau'_j\right)^2\frac{1}{2}(a_{j-1} + a_{j+1}) - \kappa_j^2 a_j$, which leads to the eikonal equation. If we assume that eikonal equation $\left(\tau'_j\right)^2 = \kappa_j^2$ is held, we can replace this term with $\frac{1}{2}\omega^2\kappa_j^2(a_{j-1} - 2a_j + a_{j+1})$, which is the discrete Laplacian of a multiplied by some factor. To conclude, if we neglect the third term of the Taylor expansion for *exp*(), then the discretization of (4) that is close to $M^{-1}HM$ given an exact τ is

$$\frac{1}{h^2}(a_{j-1} - 2a_j + a_{j+1})\left(1 - \frac{1}{2}h^2\omega^2\kappa_j^2\right) - i\omega\left(2\tau'_j\left(\frac{a_{j+1} - a_{j-1}}{2h}\right) + \tau''_j\frac{1}{2}(a_{j+1} + a_{j-1})\right) = \hat{q}_j. \tag{20}$$

In light of this comparison, we use an averaging mass matrix in the mass term $i\omega(\Delta\tau)a$.

The comparison above shows that there is an $O(\omega^2 h^2)$ difference between the two discretizations, which means that the solution of the two systems will be similar only if ωh is small enough. This is also a requirement that we have for discretizing (1) using standard methods. In fact,[55] states that the discretization error in the Helmholtz computations is of size $O(\omega^3 h^2)$, which imposes an even stronger requirement on the mesh size than a small ωh.

3 | MULTIGRID SOLVERS FOR THE ADR EQUATIONS

In this section, we describe the multigrid approaches that we use for solving (10) and (11) and for solving (9) using the shifted Laplacian method. Generally, multigrid approaches aim at solving linear systems

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

iteratively by using two complementary processes: relaxation and coarse grid correction. The relaxation is obtained by a standard iterative method like Jacobi or Gauss–Seidel, which is only effective for reducing error that is spanned by the eigenvectors of A that correspond to relatively high eigenvalues (in magnitude). The remaining error, called "algebraically smooth," is spanned by the eigenvectors of A corresponding to small eigenvalues (in magnitude), that is, vectors $\mathbf{e}$ subject to

$$\|A\mathbf{e}\| \ll \|A\|\|\mathbf{e}\|. \tag{21}$$

To reduce this algebraically smooth error, multigrid methods use a coarse grid correction, where the error $\mathbf{e}$ for some iterate $\mathbf{x}^{(k)}$ is estimated by solving a coarser system

$$A_c\mathbf{e}_c = \mathbf{r}_c = P^{\top}(\mathbf{b} - A\mathbf{x}^{(k)}).$$

The matrix A_c is an approximation of matrix A on a coarser grid (the subscript c denotes coarse components). Matrix P is a transfer operator that is used for projecting the residual onto the coarser grid and interpolating $\mathbf{e}_c$—the solution of the coarse system—back onto the fine grid, that is,

$$\mathbf{e} = P\mathbf{e}_c. \tag{22}$$

This process is effective if any algebraically smooth error $\mathbf{e}$ satisfying (21) can be represented in the range of the interpolation P. The coarse operator A_c can be obtained by either rediscretizing the problem on a coarser grid or by the Galerkin operator

$$A_c = P^{\top}AP. \tag{23}$$

Algorithm 1 summarizes the process using two grids. By treating the coarse problem recursively, we obtain the multigrid V-cycle, and by treating the coarse problem recursively twice (by two recursive calls to the V-cycle), we obtain a W-cycle. For more information, see the works of Briggs et al.,[56] Trottenberg et al.,[57] and Yavneh[58] and the references therein.

Algorithm 1. Two-grid cycle

```
Algorithm $\mathbf{x} \leftarrow TwoGrid(A, \mathbf{b}, \mathbf{x})$
1. Apply pre-relaxations: $\mathbf{x} \leftarrow Relax(A, \mathbf{x}, \mathbf{b})$.
2. Define and restrict the residual $\mathbf{r}_c = P^T(\mathbf{b} - A\mathbf{x})$.
3. Define $\mathbf{e}_c$ as the solution of the coarse-grid problem $A_c\mathbf{e}_c = \mathbf{r}_c$.
4. Prolong $\mathbf{e}_c$ and apply coarse grid correction: $\mathbf{x} \leftarrow \mathbf{x} + P\mathbf{e}_c$.
5. Apply post-relaxations: $\mathbf{u} \leftarrow Relax(A, \mathbf{x}, \mathbf{b})$.
```

3.1 | The shifted Laplacian multigrid method

The shifted Laplacian multigrid method is one of the most common approaches to solve the Helmholtz equation. This method is implemented in efficient software packages.[21,59] To solve the linear system (9) using the shifted Laplacian approach, one introduces a shifted system by adding a complex negative mass matrix to (9), as follows:

$$H_s = H - i\omega^2\alpha\ \mathrm{diag}\,(\kappa^2), \tag{24}$$

where $\alpha > 0$ is a shifting parameter. From a physical point of view, this term is equivalent to adding attenuation to the equation, which means that the waves in the shifted problem decay rapidly if α is large enough, resulting in a *local* approximation of the waveform. In the shifted Laplacian approach, the shifted matrix is used as a preconditioner for the Helmholtz linear system (9) inside a Krylov method, which is usually chosen to be (flexible) GMRES.[23] The preconditioning is obtained by applying a multigrid cycle for approximately inverting the shifted matrix (24). The larger α is, the more efficient the solution of the *shifted* system using multigrid is, but the quality of (24) as a preconditioner deteriorates. The compromise suggested in the work of Erlangga et al.[12] is to use $\alpha = 0.5$, together with rather standard multigrid cycles, whereas the works of Calandra et al.[13] and Treister and Haber[8] suggest applying more elaborate cycles.

We define the prolongation P to be a bilinear interpolation operator, because the Laplacian operator in (1) is homogeneous (does not have varying coefficients). As relaxation, the damped Jacobi method is often chosen, and its damping parameter needs to be chosen differently for each level, as on coarse grids, the wave number becomes larger compared to the mesh size. Consequently, the matrix becomes more indefinite and even negative definite at some level.[13,60] Another choice of relaxation is the GMRES method, which automatically adapts to the matrix at each level. This relaxation method was originally suggested in the work of Elman et al.[60] for the Helmholtz equation and was recently used in the works of Cools et al.[19] and Calandra et al.[13]

Apart from the type of relaxation and prolongation, one has to choose the number of levels used in the multigrid hierarchy. Unlike many other multigrid scenarios, the algebraically smooth error modes of the Helmholtz operator have a sign-changing behavior at a high wave number and, therefore, cannot be represented accurately on very coarse grids. Hence, the performance of the solver deteriorates when using more levels. For example, the results in the work of Calandra et al.[13] show that the best performance is achieved using three levels only, which is also what we get in our experience. The works of Calandra et al.[13] and Erlangga and Nabben[18] invest more work on the second grid, as oscillatory errors are significantly better represented on this grid than on the other coarser grids. However, when using only a few levels, we get a rather large coarsest grid problem, which requires a relatively accurate solution. Factorizing the coarsest grid matrices for large-scale 3D problems is memory consuming and limiting. The work of Calandra et al.[13] suggests an inexact solution of the coarsest grid using GMRES, which is the approach that we adopt in this work.

Another acceleration technique, suggested in the work of Erlangga and Nabben,[18] uses a recursive multilevel Krylov solver for the coarse grid problems. Such a cycle is called the Krylov cycle and has a rather elaborated recursive structure, depending on the number of Krylov iterations at each level. If this number is 2 (which is the common choice), we get the structure of a W-cycle—see Algorithm 2.

Algorithm 2. Krylov multigrid cycle

```
Algorithm $\mathbf{x} \leftarrow Kcycle(A, \mathbf{b}, \mathbf{x})$
1. Apply pre-relaxations: $\mathbf{x} \leftarrow Relax(A, \mathbf{x}, \mathbf{b})$.
2. Define and restrict the residual $\mathbf{r}_c = P^T(\mathbf{b} - A\mathbf{x})$.
3. If coarsest level is reached - solve $A_c\mathbf{e}_c = \mathbf{r}_c$, possibly inexactly.
   Otherwise, apply FGMRES(2) for $A_c\mathbf{e}_c = \mathbf{r}_c$ starting from 0,
   with Kcycle() as a preconditioner.
4. Prolong $\mathbf{e}_c$ and apply coarse grid correction: $\mathbf{x} \leftarrow \mathbf{x} + P\mathbf{e}_c$.
5. Apply post-relaxations: $\mathbf{u} \leftarrow Relax(A, \mathbf{x}, \mathbf{b})$.
```

3.2 | The solution of the ADR linear system

We have two ADR linear systems in (10) and (11). The work of Haber and Maclachlan[15] suggests to use $M\hat{H}M^{-1}$ as a preconditioner to (9), where $\hat{H}$ is one of the ADR matrices. From the results and local Fourier analysis in the work of Haber and Maclachlan,[15] we learn the following.

1. Even at high frequency, the operator $M\hat{H}_{\mathrm{cen}}M^{-1}$ is a good preconditioner to H, and the two matrices are spectrally similar. The linear system (10) is hard to solve using multigrid, similarly to the standard (9).
2. The ADR operator $M\hat{H}_{\mathrm{2up}}M^{-1}$ is not a good preconditioner to H, suggesting that the operators are indeed different. However, the ADR linear system (11) can be efficiently solved using multigrid.

Even though we use a significantly different τ than that in the work of Haber and Maclachlan,[15] we observed the same properties in our case as well. In addition, the ADR system with a *first*-order upwind advection operator is solved very

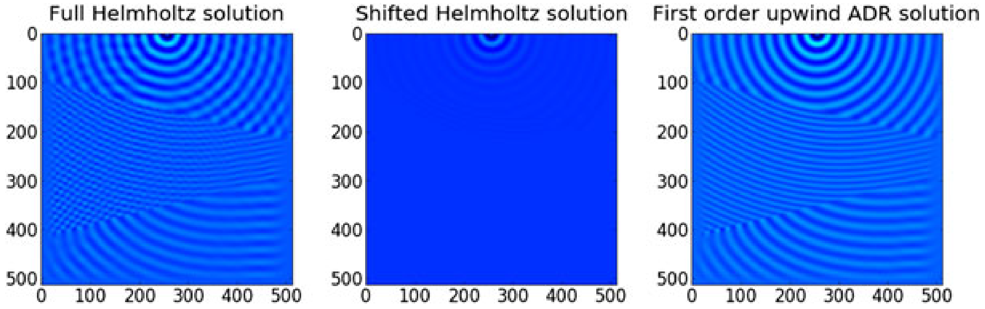

FIGURE 2 A demonstration of the local and global approximations of the Helmholtz solution. On the left, there is a solution to (1) for point source with a reflective model (the Wedge model that appears later). Indeed, reflections are evident in the middle and upper parts of the domain. In the middle figure, we show the solution obtained by solving the shifted problem (24) for the same point source, for $\alpha = 0.2$. It is clear that the waves decay rapidly, and only two to three wavelengths are approximated well. This is the reason why the shifted Laplacian method is very sensitive to the number of wavelengths in the domain. On the left, we see the solution of (25) for the point source, which can be obtained almost as easily as the solution of (24). This time, we see the global main wave function on the entire domain, but the discretization eliminates all the reflections

efficiently by multigrid. This is not surprising because the first-order advection operator can be obtained as a sum of a second-order advection operator and a Laplacian operator, which is well represented on coarser grids (for both upwind and central schemes). This is similar to having a "Laplacian shift" term of $-i\omega h\Delta a$. Unlike the standard shift in (24), the "Laplacian shift" strongly damps oscillatory modes in a but hardly influences spatially smooth modes. Because we solve the Helmholtz equation for a point source and use an accurate travel time, most of the amplitude a is smooth (up to reflections that are oscillatory). This results in a *global* approximation, which has the global behavior of the solution corresponding to the first arrival of the wave but has almost no reflections. In Figure 2, we demonstrate such a global approximation compared with a local approximation obtained by the shifted Laplacian preconditioner.

We solve system (10) in two stages. In the first stage, we compute the global approximation of the solution that corresponds to the first arrival of the wave without treating the reflections well. This is efficiently obtained by approximately solving

$$\hat{H}_{1\text{up}}\mathbf{a} = \hat{\mathbf{q}} \tag{25}$$

up to a quite low accuracy level. Once (25) is approximately solved, we finalize the solution and add the missing reflections. To this end, we use the approximate solution of (25) as an initial guess and apply the shifted Laplacian approach to the ADR system rescaled as Helmholtz, that is,

$$M\hat{H}_{\text{cen}}M^{-1}\mathbf{u} = \mathbf{q}, \tag{26}$$

which is the opposite of (16). Solving (25) inaccurately is achieved by Algorithm 2 in very few iterations; hence, most of the work in obtaining the solution is invested in the second stage. We note that, in principle, we can use only the shifted Laplacian approach to solve (10) without the first stage, and the opposite is also true—we can use the two-stage approach to treat (9) using opposite rescaling.

Algorithm 3. The two-stage solution of the central difference ADR equation

```
Preprocessing: Solve the factored eikonal equation and obtain the travel time $\tau$.
# Stage 1: Define a global approximate solution
1. Compute $\mathbf{a}_{1\text{up}}$ as a low-accuracy solution of $\hat{H}_{1\text{up}}\mathbf{a} = \hat{\mathbf{q}}$.
2. Define $M = \text{diag}(\exp(-i\omega\tau))$, $\mathbf{u}_{1\text{up}} = M\mathbf{a}_{1\text{up}}$.

# Stage 2: Complete the solution using the shifted Laplacian method
1. Define the shifted operator: $H_s^{\text{cen}} = M\hat{H}_{\text{cen}}M^{-1} - i\omega^2\alpha\text{diag}(\kappa^2)$.
2. Solve $M\hat{H}_{\text{cen}}M^{-1}\mathbf{u} = \mathbf{q}$ using $H_s^{\text{cen}}$ as preconditioner with multigrid, starting from $\mathbf{u}_{1\text{up}}$.
```

To solve the ADR system (11) using second-order upwind discretization, we use a preconditioner matrix

$$(1-\beta)\hat{H}_{2\mathrm{up}} + \beta\hat{H}_{1\mathrm{up}}, \tag{27}$$

where $\hat{H}_{1\mathrm{up}}$ is the ADR system with first-order upwind advection in Equation (25). We treat this preconditioner using Algorithm 2.

4 | NUMERICAL RESULTS

In this section, we compare two aspects of the ADR approach for solving the Helmholtz problem. First, we compare the accuracy of the numerical solution $u(\vec{x})$ of the Helmholtz Equation (1) when it is discretized using the standard and ADR second-order discretizations, where $u(\vec{x})$ is composed of the solutions of (4) and (6) through (3). To obtain the travel time τ, we use the fast marching algorithm in the work of Treister and Haber[47] using a second-order upwind discretization. Second, we compare the computational effort required to obtain the different solutions using multigrid methods.

4.1 | Accuracy comparison

In this section, we empirically compare the accuracy of the different discretizations described in Section 2. We test the accuracy by solving a given problem with all discretizations and comparing the solutions to a reference solution obtained on a four times finer grid using the standard second-order discretization (9). Some of the test cases that we present involve caustics and reflections.

We consider four heterogeneous test cases, all on a 2D unit square, discretized on a nodal regular 513 × 513 grid. Motivated by geophysical applications, we place the source point on the top row of the model (the surface), where we use Neumann BCs. On the bottom and sides of the model, we use an absorbing BC to prevent reflection from the model boundaries, as the waves are supposed to continue spreading from these boundaries. We compare the obtained solutions to a reference solution $\mathbf{u}_{\mathrm{ref}}$ calculated for the same problem on a 2049 × 2049 grid, and then downsampled. The δ function at the source is discretized as $\frac{1}{h^2}$. We consider the four models shown in the top row of Figure 3, where on the bottom are

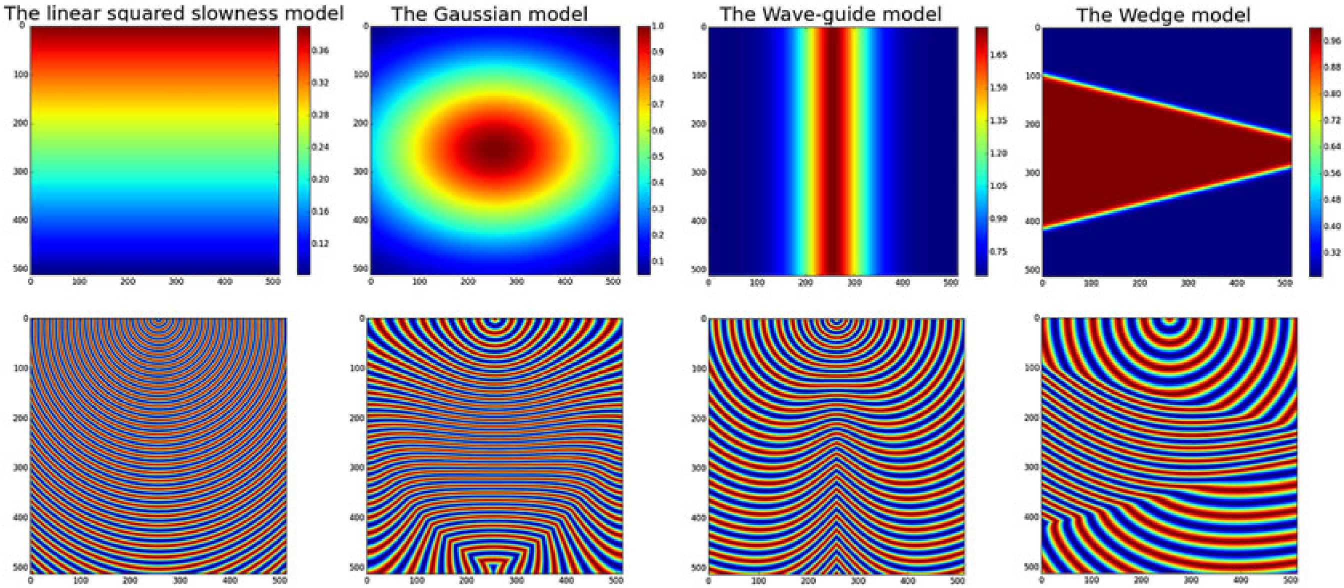

FIGURE 3 The four test cases for the accuracy comparison. In the upper row, we show the model $\kappa(\vec{x})^2$ for each test. From left to right: the linear squared slowness model, the Gaussian model, the waveguide model, and the wedge model. On the bottom row, we show the phase according to the travel time τ corresponding to each model

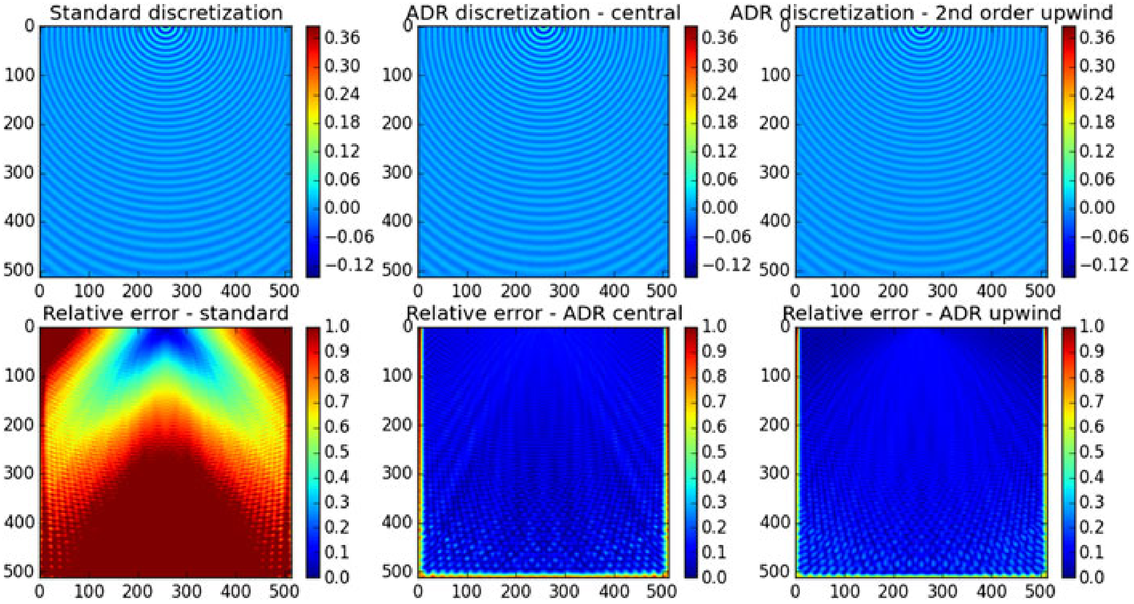

FIGURE 4 Accuracy comparison for the linear model. Note that the error introduced by the standard discretization is mostly a phase error, which is not observed in the other advection–diffusion–reaction (ADR) discretizations

the corresponding phases $\exp(-i\omega\tau)$. For each discretization—the standard second-order finite difference, the ADR with central difference advection, and the ADR with second-order upwind advection—we show the obtained solution $\mathbf{u}$ and the *relative* error

$$e_{ij} = \frac{|u_{ij} - (u_{\text{ref}})_{ij}|}{|(u_{\text{ref}})_{ij}|}.$$

4.1.1 | Linear squared slowness model

In this test case, the squared slowness $\kappa(\vec{x})^2$ is a linear model in the y direction, starting from 0.4 at the top of the model and decreases to 0.08 at the bottom. This model is the first model from the left in Figure 3. For this test case, we use a high value of ω corresponding to at least 11 grid points per wavelength. Figure 4 shows the real value of $\mathbf{u}$ for the three discretizations and the relative absolute error. The solution for this test does not contain reflections or caustics, and therefore, the amplitude and travel time are smooth. Visually, the three solutions are similar, but the error plot shows a very high error for the standard discretization. This is a phase error that is caused by the dispersion phenomenon mentioned earlier. The ADR discretizations do not include the dispersion because the phase is obtained accurately by the relatively smooth τ. This example illustrates that when there are no reflections or caustics and the travel time is smooth (except at the point source), the solution for the amplitude obtained by the ADR equation is more accurate than the standard approach.

4.1.2 | Gaussian model

In this test case, the squared slowness $\kappa(\vec{x})^2$ is a Gaussian function, that is,

$$\kappa(\vec{x})^2 = \exp(-(\vec{x} - 0.5)^{\top}\Sigma(\vec{x} - 0.5)), \quad \Sigma = \begin{bmatrix} 4 & 0 \\ 0 & 8 \end{bmatrix}.$$

The model appears second from the left in Figure 3, and the corresponding travel time τ has a discontinuity at the bottom of the model. For this test case, we choose ω; thus, we have at least 12 grid points per wavelength.

Figure 5 shows the obtained solutions, which are visually similar. Because of the caustics, this time, all approximations have significant errors, but the error is more dominant when using the standard discretization than in the ADR discretizations. The upwind ADR discretization yields the most accurate solution in general but is comparable to the central difference ADR solution. The standard discretization again yields a solution with dispersion errors because the frequency is high.

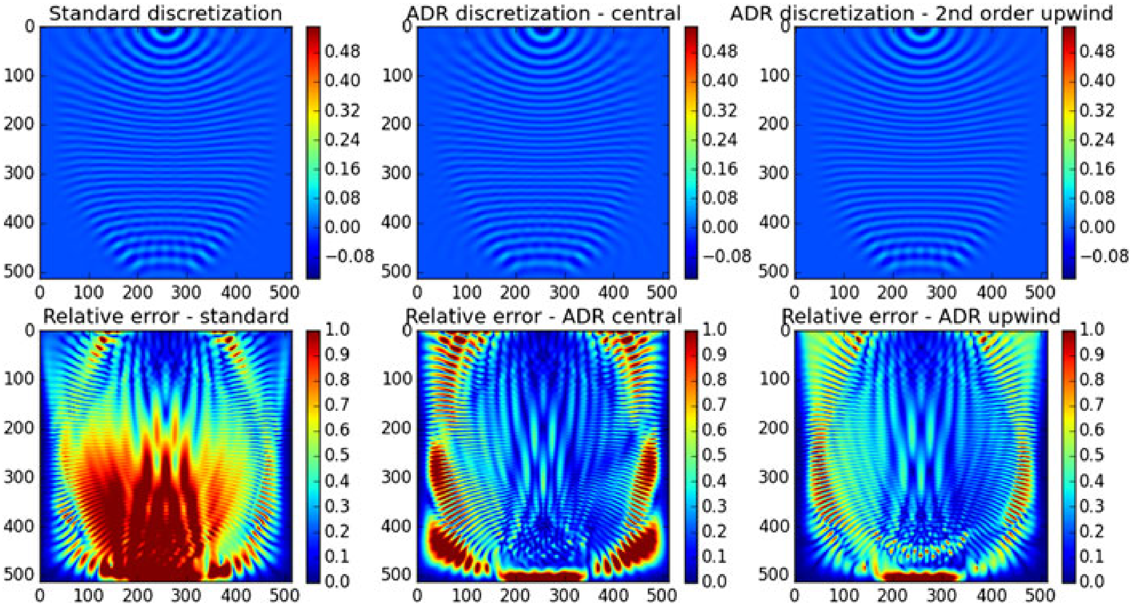

FIGURE 5 Accuracy comparison for the Gaussian model

4.1.3 | The waveguide model

In this test case, the squared slowness $\kappa(\vec{x})^2$ is the waveguide model given by

$$v(\vec{x}) = \exp(1.25 * (1 - 0.4 * \exp(-32 * (x_1 - 0.5)^2))); \quad \kappa(\vec{x})^2 = \frac{1}{v^2}.$$

The model is shown second from the right in Figure 3. Even though the model is very smooth, it generates severe caustics, which leads to inferences in the solution. Since this test case is very complicated, we use a modest ω with at least 20 grid points per wavelength.

Figure 6 shows the obtained solutions, and in particular, it shows the inferences at the bottom part. All solutions are again visually similar. Because of the caustics, there are errors, and this time, it is clear that the ADR discretization with central difference advection has the lowest error of the three, and the ADR with upwind advection has the highest error

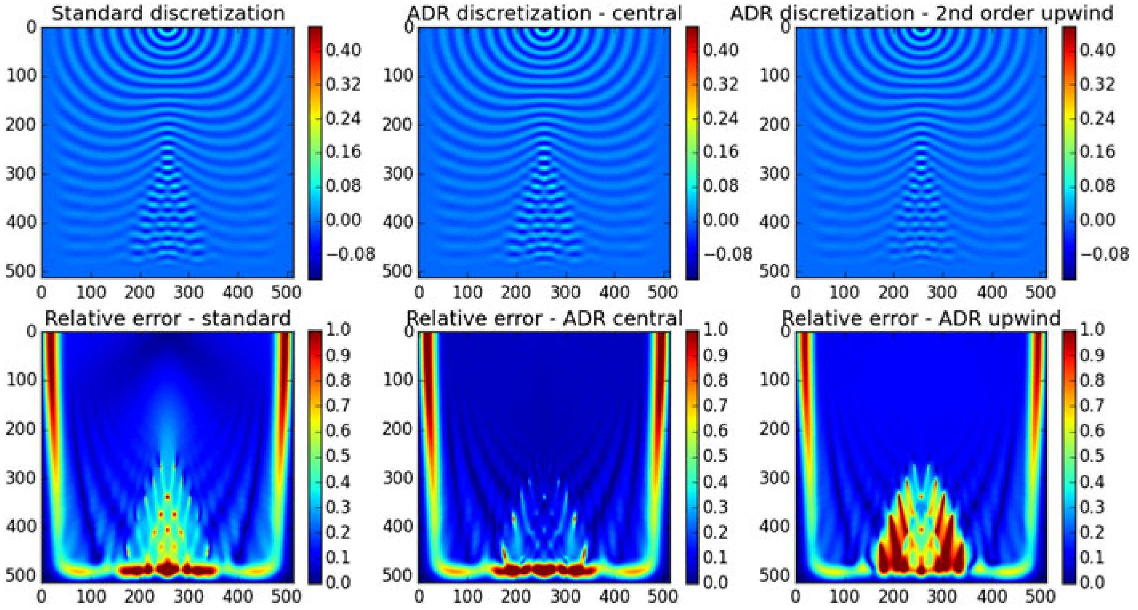

FIGURE 6 Accuracy comparison for the waveguide model

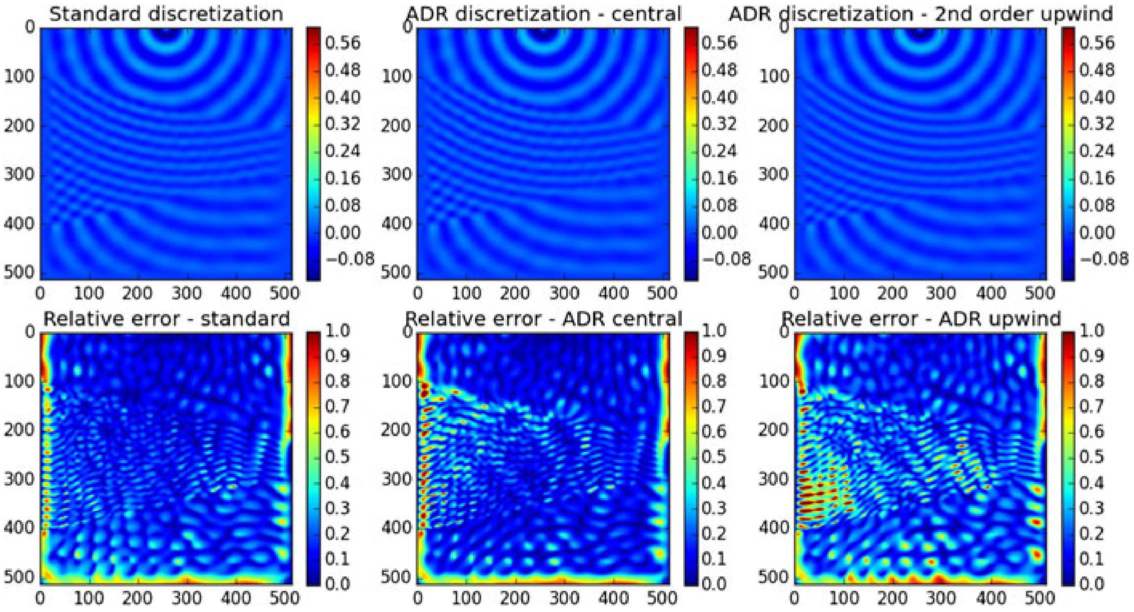

FIGURE 7 Accuracy comparison for the wedge model

compared with the reference solution. Because the frequency is not so high, the standard discretization does not introduce dispersion errors.

4.1.4 | The wedge model

This model, which is the rightmost model in Figure 3, is usually given with a sharp step. Here, we smooth the step to be able to have a somewhat accurate fine approximation of the solution as a reference. The top half of the model is given by the function

$$\kappa(\vec{x})^2 = 0.25 * (\tanh((4 * x_2 - x_1 - 0.75) * 20)) + 0.75,$$

and the bottom half is generated by mirroring the top half. In our experiment, the step is smoothed over approximately 20 grid points, for the coarser 513 × 513 grid. The frequency chosen for the experiment is $f = 20$ Hz ($\omega = 2\pi f$), and the wavelength is at least 25 grid points. This model generates reflections from the wedge and is hard to model accurately because of the sharp change in the model.

Figure 7 shows the obtained solutions, and in particular, it shows the reflections at the top and middle parts of the model. Visually, the two left approximations are similar, but the right one includes somewhat weaker reflections at the middle part of the model. Besides that, all solutions show comparable errors with the lowest error obtained by the standard discretization. We note that particularly in this test case, we are not certain that the reference solution accurately models the reflections.

4.2 | Numerical solution performance

In this section, we compare the computational effort required to solve the Helmholtz problem with the three discretizations described earlier. We present examples that appear in geophysical applications, where, typically, the length of the domain is long, and the depth of the domain is rather short. We consider 2D and 3D examples, for which we test the performance of the solver at high frequency (10–12 grid points per wavelength) and intermediate frequency (about 15–17 grid points per wavelength). For each case, we present three grid sizes to demonstrate the scalability of the solvers as the mesh size and frequency grow together. In all cases, we use flexible GMRES(5) with multigrid as a preconditioner and seek a solution with relative residual accuracy of 10^{-5}, starting from a zero initial guess. For solving the linear system (9) arising from the standard discretization, we use the shifted Laplacian preconditioner with a shift $\alpha = 0.2$. We use a geometric multigrid configuration that includes the Krylov multigrid cycles in Algorithm 2, with Jacobi-preconditioned GMRES as relaxation. We define our coarse grid problems by the Galerkin product $P^T HP$. As the problem becomes more indefinite on coarser grids, we found that it is worth applying more relaxations on the coarser grids, and therefore, on level l, we apply $l + 1$ pre- and post-relaxations. That is, two for the first level, three for the second, and so on. Having large 3D problems in mind, we use five levels and approximate the coarsest grid solution. As in the work of Calandra et al.,[13] we apply 10 Jacobi-preconditioned GMRES iterations instead of a direct solve using a factorization. Because we use relatively elaborate and expensive cycles, we are able to solve the Helmholtz problem using much less cycles compared to using more standard cycles with $\alpha = 0.5$.

We use the same multigrid cycles for the ADR discretizations. For solving (11), we use (27) as a preconditioner treated by multigrid and choose $\beta = 0.25$. This seems to be the most effective option together with our elaborated cycles for test cases at rather large scales. For solving (10), we apply the two-stage scheme described in Algorithm 3. For the first stage, we apply, at most, five Krylov cycles or stop earlier if the relative residual drop with respect to (25) is 10^{-2}. Once the intermediate residual drop is reached, we switch to solving (26) using the same shifted Laplacian configuration mentioned before until convergence, to capture the reflections in the solution.

We show the total number of Krylov cycles required for convergence (#*it*), the solution time (t_{sol}), and the time required to compute the travel time τ by fast marching (t_{FM}). Our code is written in Julia language[61] and is available as part of the jInv software[62] (see https://github.com/JuliaInv/ForwardHelmholtz.jl). Our 2D tests were computed on a laptop machine using Windows 10 64bit OS, with Intel core-i7 2.8-GHz CPU with 32 GB of RAM. The 3D tests were computed on a workstation with Intel Xeon E5-2620 2GHz X2 (6 cores per socket, 2 Threads per core for a total of 24 cores) with 64 GB of RAM, running on Centos 7 Linux distribution. For the 3D results, we use single-precision computations, to save memory. The code for the solution phase is parallelized in the matrix-vector products, whereas the FM code is sequential as the algorithm is sequential. The parallelism can be efficiently exploited for FM if many linear systems need to be solved, and the travel times are calculated simultaneously.[62]

4.2.1 | 2D linear model

Our first test case is a linear velocity model (κ^2 is the inverse squared velocity) that does not include reflections. The model is given in Figure 8, and the results are summarized in Table 1. It is clear that the shifted Laplacian method requires the most iterations to solve the problem. Moreover, it requires more iterations as the problem gets larger and the frequency is higher. The solution of (10), denoted as "ADR central," requires about 20%–30% less iterations and time, due to the global approximation obtained in the first phase. Solving (11) is achieved with the least iterations and time, and more importantly, it is fairly mesh and frequency independent. Since this model does not contain reflections or caustics, then the "ADR upwind" option is the best one because it provides both accurate approximation and fast solution. It is expected to continue being so at even larger scales.

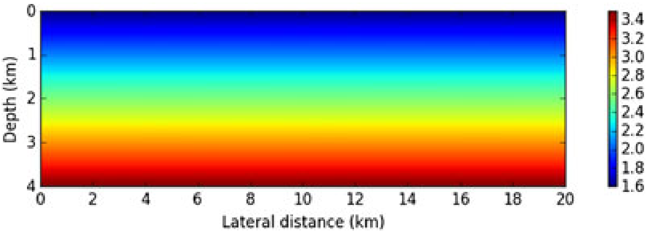

FIGURE 8 The linear velocity model. Units are in kilometers per second, and $\kappa^2 = \frac{1}{v^2}$

TABLE 1 Linear model: solution performance

	f(Hz)	Points per	*Standard*		*ADR central*		*ADR upwind*		*eikonal*
Grid size	$= \omega/2\pi$	**wavelength**	**#*it***	t_{sol}	**#*it***	t_{sol}	**#*it***	t_{sol}	t_{FM}
769 × 257	3.5	17.5	43	5.9 s	34	4.2 s	39	5.8 s	0.33 s
	5.5	11.2	67	9.1 s	54	6.5 s	33	4.7 s	0.37 s
1025 × 385	5.5	14.9	74	17.9 s	56	13.0 s	41	10.7 s	0.75 s
	7.5	10.9	102	24.8 s	74	17.0 s	40	10.4 s	0.82 s
1537 × 513	7.5	16.3	93	42.3 s	64	28.0 s	46	22.6 s	1.64 s
	11.0	11.2	151	68.2 s	93	42.9 s	46	22.5 s	1.65 s

Note. ADR = advection–diffusion–reaction.

4.2.2 | The 2D Marmousi 2 model

Our second test case is the P-wave velocity of the Marmousi 2 model,[63] which is given in Figure 9. This model is mostly piecewise constant and includes many reflectors. Table 2 summarizes the results for this test case. The performance of the shifted Laplacian approach is quite similar to the previous test case. The method is quite robust to the heterogeneity of the model and more sensitive to the frequency. In the "ADR central" section, we see less advantage compared to the previous smooth test case. That is because the "global" reflection-less approximation obtained from solving (25) is less effective. We note again that (10) can be solved using the shifted Laplacian method alone with the same efficiency as (9). The third option "ADR upwind" is again achieved in less iterations and time and is again more mesh independent than the other options.

4.2.3 | The 3D linear model

In this experiment, we consider the 3D version of the linear model in Figure 8, which is smooth and does not introduce reflections. Table 3 summarizes the results for this test case. Because of memory limitations, the resolution and frequencies are much lower here than in 2D, and therefore, the iteration counts are much lower than in 2D. In terms of iteration counts, we again see that the standard discretization is taking the most iterations, whereas the other ADR systems are solved in less iterations (with upwind ADR taking the least). The time-per-iteration in the ADR central column is lower than in the standard column, probably because of less work that is done on coarser grids. That is, in Algorithm 2, some cycles do not include a second recursive call because a threshold is achieved in FGMRES (we use 0.1 as a threshold for the coarse grid as suggested in the work of Notay and Vassilevski[64]). In light of the 2D results, we do not expect this to be a significant advantage of the ADR discretization over the standard discretization in larger problems. The cost of the FM preprocessing is higher now compared to the 2D case, because in addition to the problem size, the algorithm is more

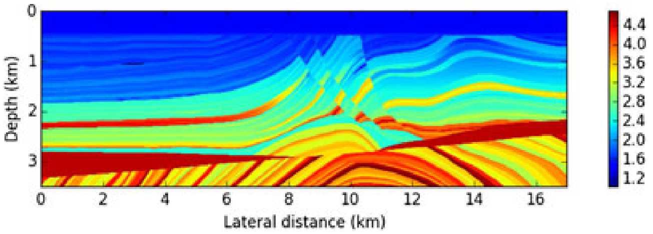

FIGURE 9 The Marmousi 2 P-wave velocity model. Units are in kilometers per second, and $\kappa^2 = \frac{1}{v^2}$

TABLE 2 Marmousi 2 model: solution performance

	f(Hz)	Points per	*Standard*		*ADR central*		*ADR upwind*		*eikonal*
Grid size	$= \omega/2\pi$	wavelength	*#it*	t_{sol}	*#it*	t_{sol}	*#it*	t_{sol}	t_{FM}
769 × 257	3.0	17.3	42	5.7 s	35	4.6 s	41	6.1 s	0.46 s
	4.5	11.5	64	8.8 s	50	7.0 s	42	6.2 s	0.47 s
1025 × 385	4.5	15.4	61	16.5 s	52	13.1 s	55	14.6 s	0.93 s
	6.5	10.8	115	28.1 s	93	22.4 s	52	13.9 s	0.84 s
1537 × 513	6.5	16.0	95	42.8 s	79	36.1 s	63	31.4 s	1.8 s
	9.0	11.5	164	76.8 s	124	58.1 s	54	26.7 s	1.8 s

Note. ADR = advection–diffusion–reaction.

TABLE 3 3D linear model: solution performance

	f(Hz)	Points per	*Standard*		*ADR central*		*ADR upwind*		*eikonal*
Grid size	$= \omega/2\pi$	wavelength	*#it*	t_{sol}	*#it*	t_{sol}	*#it*	t_{sol}	t_{FM}
257 × 257 × 65	1.5	13.6	18	115 s	18	87 s	17	107 s	16.5 s
	2.0	10.2	22	140 s	23	110 s	18	116 s	17 s
385 × 385 × 97	2.0	15.3	23	212 s	21	160 s	19	170 s	60.5 s
	3.0	10.2	33	302 s	32	242 s	20	180 s	62 s
513 × 513 × 129	3.0	13.7	35	534 s	30	401 s	21	311 s	165 s
	4.0	10.2	41	620 s	36	490 s	24	352 s	159 s

Note. ADR = advection–diffusion–reaction.

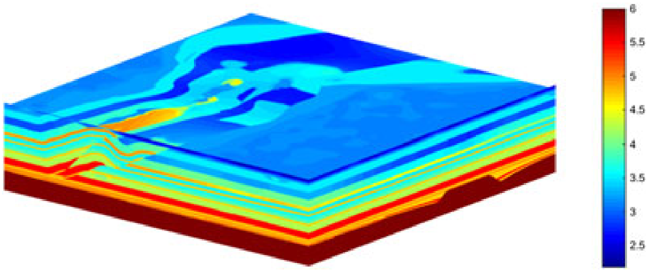

FIGURE 10 The SEG overthrust velocity model. Units are in kilometers per second, and $\kappa^2 = \frac{1}{v^2}$. The model corresponds to a domain of 20 × 20 × 4.65 km

TABLE 4 3D overthrust model: solution performance

Grid size	*f(Hz)* $= \omega/2\pi$	Points per wavelength	*Standard* #it	t_{sol}	*ADR central* #it	t_{sol}	*ADR upwind* #it	t_{sol}	*eikonal* t_{FM}
257 × 257 × 65	2.0	14.4	14	89 s	13	62 s	19	118 s	16.5 s
	3.0	9.6	20	127 s	19	91 s	23	146 s	17.2 s
385 × 385 × 97	3.0	14.4	20	185 s	18	137 s	26	236 s	67 s
	4.0	10.8	26	241 s	24	180 s	25	226 s	64 s
513 × 513 × 129	4.0	14.4	27	409 s	22	294 s	34	505 s	170 s
	6.0	9.6	44	674 s	37	504 s	32	474 s	177 s

Note. ADR = advection–diffusion–reaction.

complicated in this case. Still, the solution time of FM is not very significant, considering that the multigrid solution timings are obtained using a highly parallelized code (16 cores), whereas FM is completely serial. This can be exploited if many systems are required to be solved. As the problem gets larger and the multigrid solution takes more and more iterations, the FM cost will become less and less significant.

> *Remark.* Although the cost of each cycle in our solver is of linear complexity, the timings in Table 3 scale slightly better than linearly. That is a result of better performance in the parallelism of our code—as the problem grows, the efficiency of the matrix-vector multiplications is better. This behavior is obviously independent of the method.

4.2.4 | The 3D overthrust model

The last model that we present is the 3D SEG overthrust model,[65] which, similarly to the Marmousi 2 model, includes many reflecting layers. The model appears in Figure 10, and Table 4 summarizes the performance results. At the smaller sizes, the upwind ADR discretization requires the most effort to solve, but it becomes more efficient compared with the other discretizations when the problem gets larger. The shifted Laplacian approach used for the standard and central ADR discretizations yields comparable counts for both. Again, there is a slight edge to ADR central due to Stage 1 in Algorithm 3. We expect that at larger scales, we will see results that are similar to the results in the 2D Marmousi 2 model.

5 | CONCLUSIONS AND FUTURE WORK

In this paper, we present a new approach for discretizing and solving the Helmholtz equation with a point source. We reformulate the problem based on the Rytov decomposition of the solution, yielding an eikonal equation for the phase and a complex-valued ADR equation for the amplitude. We choose the phase based on the travel time of the wave and compute it based on the factored eikonal equation using the fast marching algorithm. The factored version yields a more accurate treatment of the point source for the phase and a relatively smooth solution for the amplitude. The ADR equation is discretized using second-order upwind and central difference discretizations, and the solution of the system is achieved using multigrid. Our approach has two main advantages. First, the majority of the solution of the Helmholtz equation is represented by smooth functions, and hence, the reformulated problems are more suitable for multigrid computations. Secondly, the obtained solution is not a first-arrival anzatz only—it includes all the information of the wave propagation including reflections and inferences.

Our accuracy results show that for models that do not introduce caustics and reflections, our approach yields more accurate solutions than standard Helmholtz discretization and, in particular, does not introduce a phase error. When reflections and caustics are observed, the accuracy of the ADR discretizations is comparable to the standard one. Our performance results show that the standard and central ADR discretizations are both hard to solve. The ADR system using upwind discretization is solved more efficiently but is less accurate than the ADR discretization with central difference in the presence of reflections.

Our approach is intriguing for geophysical applications, especially FWI where many solutions of the Helmholtz equations are required for a point source. There, the model is adapted iteratively, starting from a smooth model like in Figure 8 until a detailed model like in Figure 9 is estimated in the final stages. Smooth models are encountered frequently, and hence, our approach can be beneficial computationally. More importantly, our ADR discretization does not introduce dispersion errors for the main wave, which may be very hard to overcome in the inversion process in FWI.

Our future research aims to explore the advantages of the new discretizations in the context of FWI and to further improve the numerical solver for the Helmholtz equation at larger scales and a higher wave number.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement 623212—MC Multiscale Inversion.

ORCID

Eran Treister http://orcid.org/0000-0002-5351-0966

REFERENCES

1. Pratt RG. Seismic waveform inversion in the frequency domain, part 1: theory, and verification in a physical scale model. Geophysics. 1999;64(3):888–901.
2. Epanomeritakis I, Akçelik V, Ghattas O, Bielak J. A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion. Inverse Probl. 2008;24(3).
3. Virieux J, Operto S. An overview of full-waveform inversion in exploration geophysics. Geophysics. 2009;74(6):WCC1–WCC26.
4. Krebs JR, Anderson JE, Hinkley D, et al. Fast full-wavefield seismic inversion using encoded sources. Geophysics. 2009;74(6):WCC177–WCC188.
5. Biondi B, Almomin A. Simultaneous inversion of full data bandwidth by tomographic full-waveform inversion. Geophysics. 2014;79(3):WA129–WA140.
6. Métivier L, Brossier R, Virieux J, Operto S. Full waveform inversion and the truncated Newton method. SIAM J Sci Comput. 2013;35(2):B401–B437.
7. Métivier L, Brossier R, Operto S, Virieux J. Full waveform inversion and the truncated Newton method. SIAM Rev. 2017;59(1):153–195.
8. Treister E, Haber E. Full waveform inversion guided by travel time tomography. SIAM J Sci Comput. 2017;39(5):S587–S609.
9. Singer I, Turkel E. A perfectly matched layer for the Helmholtz equation in a semi-infinite strip. J Comput Phys. 2004;201(2):439–465.
10. Engquist B, Majda A. Radiation boundary conditions for acoustic and elastic wave calculations. Commun Pure Appl Math. 1979;32(3):313–357.
11. Liao Q, McMechan GA. Multifrequency viscoacoustic modeling and inversion. Geophysics. 1996;61(5):1371–1378.
12. Erlangga YA, Oosterlee CW, Vuik C. Novel multigrid based preconditioner for heterogeneous Helmholtz problems. SIAM J Sci Comput. 2006;27(4):1471–1492.
13. Calandra H, Gratton S, Pinel X, Vasseur X. An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media. Numer Linear Algebra Appl. 2013;20(4):663–688.
14. Poulson J, Engquist B, Li S, Ying L. Parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations. SIAM J Sci Comput. 2013;35(3):C194–C212.
15. Haber E, Maclachlan S. A fast method for the solution of the Helmholtz equation. J Comput Phys. 2011;230(12):4403–4418.
16. Oosterlee CW, Vuik C, Mulder WA, Plessix R-E. Shifted-Laplacian preconditioners for heterogeneous Helmholtz problems. In: Advanced computational methods in science and engineering. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2010; 21–46.
17. Airaksinen T, Heikkola E, Pennanen A, Toivanen J. An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation. J Comput Phys. 2007;226(1):1196–1210.
18. Erlangga YA, Nabben R. On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian. Electron Trans Numer Anal. 2008;31(403-424):403–424.
19. Cools S, Reps B, Vanroose W. A new level-dependent coarse grid correction scheme for indefinite Helmholtz problems. Numer Linear Algebra Appl. 2014;21(4):513–533.

20. Tsuji P, Tuminaro R. Augmented AMG-shifted Laplacian preconditioners for indefinite Helmholtz problems. Numer Linear Algebra Appl. 2015;22(6):1077–1101.

21. Reps B, Weinzierl T. Complex additive geometric multilevel solvers for Helmholtz equations on spacetrees. ACM Trans Math Softw (TOMS). 2017;44(1). Article ID 2.

22. Ganesh M, Morgenstern C. An efficient multigrid algorithm for heterogeneous acoustic media sign-indefinite high-order FEM models. Numer Linear Algebra Appl. 2017;24(3).

23. Saad Y. Flexible inner-outer preconditioned GMRES algorithm. SIAM J Sci Comput. 1993;14(2):461–469.

24. Van der Vorst HA. Bi-CGSTAB: a fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comput. 1992;13(2):631–644.

25. Engquist B, Ying L. Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation. Commun Pure Appl Math. 2011;64(5):697–735.

26. van Leeuwen T, Gordon D, Gordon R, Herrmann FJ. Preconditioning the Helmholtz equation via row-projections. Paper presented at: 74th EAGE Conference & Exhibition; 2012; Copenhagen, Denmark. Vancouver, Canada: Seismic Laboratory for Imaging and Modeling; 2012.

27. Gordon D, Gordon R. Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers. J Comput Appl Math. 2013;237(1):182–196.

28. Brandt A, Livshits I. Wave-ray multigrid method for standing wave equations. Electron Trans Numer Anal. 1997;6:162–181.

29. Olson LN, Schroder JB. Smoothed aggregation for Helmholtz problems. Numer Linear Algebra Appl. 2010;17(2-3):361–386.

30. Livshits I. A scalable multigrid method for solving indefinite Helmholtz equations with constant wave numbers. Numer Linear Algebra Appl. 2014;21(2):177–193.

31. Leung S, Qian J, Burridge R. Eulerian Gaussian beams for high-frequency wave propagation. Geophysics. 2007;72(5):SM61–SM76.

32. Luo S, Qian J. Factored singularities and high-order Lax–Friedrichs sweeping schemes for point-source traveltimes and amplitudes. J Comput Phys. 2011;230(12):4742–4755.

33. Luo S, Qian J, Zhao H. Higher-order schemes for 3D first-arrival traveltimes and amplitudes. Geophysics. 2012;77(2):T47–T56.

34. Luo S, Qian J, Burridge R. Fast Huygens sweeping methods for Helmholtz equations in inhomogeneous media in the high frequency regime. J Comput Phys. 2014;270:378–401.

35. Crandall MG, Lions P-L. Viscosity solutions of Hamilton-Jacobi equations. Trans Am Math Soc. 1983;277(1):1–42.

36. Rouy E, Tourin A. A viscosity solutions approach to shape-from-shading. SIAM J Numer Anal. 1992;29(3):867–884.

37. Tsitsiklis JN. Efficient algorithms for globally optimal trajectories. IEEE Trans Autom Control. 1995;40(9):1528–1538.

38. Sethian JA. A fast marching level set method for monotonically advancing fronts. Proc Natl Acad Sci USA. 1996;93(4):1591–1595.

39. Sethian JA. Fast marching methods. SIAM Rev. 1999;41(2):199–235.

40. Tsai Y-HR, Cheng L-T, Osher S, Zhao H-K. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. SIAM J Numer Anal. 2003;41(2):673–694.

41. Zhao H. A fast sweeping method for eikonal equations. Math Comput. 2005;74(250):603–627.

42. Zhang Y-T, Zhao H-K, Qian J. High order fast sweeping methods for static Hamilton-Jacobi equations. J Sci Comput. 2006;29(1):25–56.

43. Kao CY, Osher S, Qian J. Lax–Friedrichs sweeping scheme for static Hamilton-Jacobi equations. J Comput Phys. 2004;196(1):367–391.

44. Qian J, Zhang Y-T, Zhao H-K. A fast sweeping method for static convex Hamilton-Jacobi equations. J Sci Comput. 2007;31(1):237–271.

45. Fomel S, Luo S, Zhao H. Fast sweeping method for the factored eikonal equation. J Comput Phys. 2009;228(17):6440–6455.

46. Pica A. Fast and accurate finite-difference solutions of the 3D eikonal equation parametrized in celerity. Paper presented at: 1997 SEG Annual Meeting; 1997; Dallas, TX. Tulsa, OK: Society of Exploration Geophysicists; 1997. p. 1774–1777.

47. Treister E, Haber E. A fast marching algorithm for the factored eikonal equation. J Comput Phys. 2016;324:210–225.

48. Luo S, Qian J. Fast sweeping methods for factored anisotropic eikonal equations: multiplicative and additive factors. J Sci Comput. 2012;52(2):360–382.

49. Luo S, Qian J, Burridge R. High-order factorization based high-order hybrid fast sweeping methods for point-source eikonal equations. SIAM J Numer Anal. 2014;52(1):23–44. https://doi.org/10.1137/120901696

50. Noble M, Gesret A, Belayouni N. Accurate 3-D finite difference computation of traveltimes in strongly heterogeneous media. Geophys J Int. 2014;199(3):1572–1585.

51. Singer I, Turkel E. High-order finite difference methods for the Helmholtz equation. Comput Methods Appl Mech Eng. 1998;163(1-4):343–358.

52. Singer I, Turkel E. Sixth-order accurate finite difference schemes for the Helmholtz equation. J Comput Acoust. 2006;14(03):339–351.

53. Operto S, Virieux J, Amestoy P, L'Excellent J-Y, Giraud L, Ali HBH. 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: a feasibility study. Geophysics. 2007;72(5):SM195–SM211.

54. Turkel E, Gordon D, Gordon R, Tsynkov S. Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number. J Comput Phys. 2013;232(1):272–287.

55. Bayliss A, Goldstein CI, Turkel E. On accuracy conditions for the numerical computation of waves. J Comput Phys. 1985;59(3):396–404.

56. Briggs WL, Henson VE, McCormick SF. A multigrid tutorial. 2nd ed. Philadelphia, PA: SIAM; 2000.

57. Trottenberg U, Oosterlee C, Schüller A. Multigrid. Cambridge, MA: Academic Press; 2001.

58. Yavneh I. Why multigrid methods are so efficient. Comput Sci Eng. 2006;8(6):12–22.

59. Knibbe H, Oosterlee CW, Vuik C. GPU implementation of a Helmholtz Krylov solver preconditioned by a shifted Laplace multigrid method. J Comput Appl Math. 2011;236(3):281–293.

60. Elman HC, Ernst OG, O'leary DP. Multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. SIAM J Sci Comput. 2001;23(4):1291–1315.

61. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev. 2017;59(1):65–98. https://doi.org.10.1137/141000671. http://julialang.org/publications/julia-fresh-approach-BEKS.pdf

62. Ruthotto L, Treister E, Haber E. jInv – A flexible Julia package for PDE parameter estimation. SIAM J Sci Comput. 2017;39(5):S702–S722.

63. Martin GS, Wiley R, Marfurt KJ. Marmousi2: an elastic upgrade for Marmousi. Lead Edge. 2006;25(2):156–166.

64. Notay Y, Vassilevski PS. Recursive Krylov-based multigrid cycles. Numer Linear Algebra Appl. 2008;15(5):473–487.

65. Aminzadeh F, Jean B, Kunz T. 3-D salt and overthrust models. Tulsa, OK: Society of Exploration Geophysicists; 1997.

How to cite this article: Treister E, Haber E. A multigrid solver to the Helmholtz equation with a point source based on travel time and amplitude. *Numer Linear Algebra Appl*. 2019;26:e2206. https://doi.org/10.1002/nla.2206