

# The method of polarized traces for the 3D Helmholtz equation

Leonardo Zepeda-Núñez<sup>1</sup>, Adrien Scheuer<sup>2</sup>, Russell J. Hewett<sup>3</sup>, and Laurent Demanet<sup>4</sup>

## ABSTRACT

We have developed a fast solver for the 3D Helmholtz equation, in heterogeneous, constant density, acoustic media, in the high-frequency regime. The solver is based on the method of polarized traces, a layered domain-decomposition method, where the subdomains are connected via transmission conditions prescribed by the discrete Green's representation formula and artificial reflections are avoided by enforcing nonreflecting boundary conditions between layers. The method of polarized traces allows us to consider only unknowns at the layer interfaces, reducing the overall cost and memory footprint of the solver. We determine that polarizing the wavefields in this manner yields an efficient preconditioner for the reduced system, whose rate of convergence

is independent of the problem frequency. The resulting preconditioned system is solved iteratively using generalized minimum residual, where we never assemble the reduced system or preconditioner; rather, we implement them via solving the Helmholtz equation locally within the subdomains. The method is parallelized using Message Passing Interface and coupled with a distributed linear algebra library and pipelining to obtain an empirical on-line runtime  $\mathcal{O}(\max(1, R/L)N \log N)$ , where  $N = n^3$  is the total number of degrees of freedom,  $L$  is the number of subdomains, and  $R$  is the number of right-hand sides (RHS). This scaling is favorable for regimes in which the number of sources (distinct RHS) is large, for example, enabling large-scale implementations of frequency-domain full-waveform inversion.

## INTRODUCTION

Efficient modeling of time-harmonic wave scattering in heterogeneous acoustic or elastic media remains a difficult problem in numerical analysis, yet it has broad application in seismic inversion techniques (Chen, 1997; Pratt, 1999; Virieux and Operto, 2009). In the constant-density acoustic approximation, time-harmonic wave propagation is modeled by the Helmholtz equation

$$\Delta u(\mathbf{x}) + \omega^2 m(\mathbf{x})u(\mathbf{x}) = f_s(\mathbf{x}), \quad \text{in } \Omega, \quad (1)$$

with absorbing boundary conditions, and where  $\Omega$  is a 3D rectangular domain,  $\Delta$  is the 3D Laplacian,  $\mathbf{x} = (x, y, z)$ ,  $m = 1/c^2(\mathbf{x})$  is the

squared slowness for velocity  $c(\mathbf{x})$ ,  $u$  is the wavefield, and  $f_s$  are the sources, indexed by  $s = 1, \dots, R$ . The method presented in this paper describes a framework for solving frequency-domain wave-propagation problems, and as such, while we limit our discussion to this case, there is no essential obstruction to extending the result to the variable density or elastic cases. We present it in the context of simpler physics to avoid obfuscating the details of the method.

Throughout this paper, we assume that equation 1 is in the high-frequency regime, i.e., when  $\omega \sim n$ , where  $n$  is the number of unknowns in each dimension. Although there are other regimes which can be considered high-frequency, we use the mathematical definition of high-frequency, which is equivalent to refining the grid

Manuscript received by the Editor 22 February 2018; revised manuscript received 27 November 2018; published ahead of production 30 April 2019; published online 15 July 2019.

<sup>1</sup>Formerly Massachusetts Institute of Technology, Department of Mathematics and Earth Resources Laboratory, 77 Massachusetts Ave, Cambridge, Massachusetts 02139, USA; University of California Irvine, Department of Mathematics, 540 Rowland Hall, Irvine, California 92693, USA; currently Lawrence Berkeley National Laboratory, Computational Research Division, 1 Cyclotron Road, Berkeley, California 94720, USA. E-mail: lzepeda@math.mit.edu (corresponding author).

<sup>2</sup>Formerly Massachusetts Institute of Technology, Department of Mathematics and Earth Resources Laboratory, 77 Massachusetts Ave, Cambridge, Massachusetts 02139, USA; currently Université Catholique de Louvain, Institute of Information and Communication Technologies, Electronics and Applied Mathematics, 4, Avenue G. Lemaitre, Louvain-la-Neuve B-1348, Belgium. E-mail: adrien.scheuer@uclouvain.be.

<sup>3</sup>Formerly Total E&P Research & Technology USA, LLC, 1201 Louisiana St., Suite 1800, Houston, Texas 77002, USA; currently Virginia Tech, Department of Mathematics, 460 McBryde Hall, 255 Stanger St., Blacksburg, Virginia 24061, USA. E-mail: rhewett@vt.edu.

<sup>4</sup>Massachusetts Institute of Technology, Department of Mathematics and Earth Resources Laboratory, 77 Massachusetts Ave, Cambridge, Massachusetts 02139, USA. E-mail: laurent@math.mit.edu.

© 2019 Society of Exploration Geophysicists. All rights reserved.

interval to keep the number of points per wavelength constant. This notion of high-frequency differs greatly with the one traditionally used in geometric optics.

There are two primary difficulties in solving equation 1 in the high-frequency regime: building a stable discretization of the problem and solving the resulting linear system efficiently. It is difficult to develop a stable discretization because of pollution error (see Babuska et al., 1995). That is, if we scale  $\omega \sim n$ , then the solution of the discrete linear systems will eventually degrade with respect to the solution of the partial differential equation (PDE). Pollution error is typically masked by increasing the number of points per wavelength, thus oversampling the solution. There is extensive literature on this topic, so we refer the interested reader to Fang et al. (2017). In this paper, we focus on the second issue — solving the resulting system efficiently.

Solving the full-waveform inversion (FWI) (Tarantola, 1984) problem in the frequency domain is the driving force behind these developments. Using the time-harmonic formulation of the wave equation in an FWI context affords strategies that can avoid non-convexities (cycle skipping) in the inverse problem (Alessandrini et al., 2017). Additionally, given that there can be upward of  $10^6$  distinct sources in a high-frequency seismic survey, time-domain approaches can be computationally wasteful, trading off disk storage (checkpointing) for computation that generally cannot be reused. In an idealized implementation, frequency-domain methods for FWI can be very efficient because matrix factorizations can be reused and minimal data need to be checkpointed. Practically, solving the FWI problem in the frequency domain is extremely difficult because current solvers do not perform well at high frequencies in three dimensions and are not optimally parallelized.

Despite these challenges, and given the importance of solving equation 1 in geophysical contexts, there has been a renewed interest in developing efficient, parallel algorithms to solve the ill-conditioned linear system resulting from discretizing the Helmholtz equation. Recent progress toward an efficient solver, i.e., a solver with linear complexity, has generally focused on three strategies

- 1) Fast direct solvers, which couple multifrontal techniques with compressed linear algebra to obtain efficient direct solvers with small memory footprint. However, for high-frequency problems, these methods remain suboptimal.
- 2) Classical preconditioners, such as incomplete factorization preconditioners, multigrid-based preconditioners, and other iterative methods are relatively simple to implement but suffer from superlinear asymptotic complexity and require significant tuning to achieve effective runtimes.
- 3) Sweeping-like preconditioners, which are a relatively recent domain decomposition-based approach that has been shown to achieve linear or nearly linear asymptotic complexity.

We present a method that belongs to the third category. Sweeping preconditioners (e.g., Gander and Nataf, 2005; Engquist and Ying, 2011a, 2011b; Chen and Xiang, 2013a, 2013b; Stolk, 2013; Vion and Geuzaine, 2014; Liu and Ying, 2015; Zepeda-Núñez and Demanet, 2016) and their generalizations, i.e., domain-decomposition techniques coupled with high-quality transmission/absorption conditions, offer the right mix of ideas to attain linear or near-linear complexity in two dimensions and three dimensions, provided that the medium does not have large resonant cavities (Zepeda-Núñez and Demanet, 2016), which are generally not an issue in geophysical contexts. These methods rely on the sparsity of the linear system

to decompose the domain into layers, within which classical sparse-direct methods may be used to compute the interactions within the layer. Interactions across layers are computed by sequentially sweeping through the subdomains in an iterative fashion.

We will formally introduce the mathematics of polarized traces in what follows. However, the method has a physical interpretation, which helps to motivate our developments. Within each layer of the domain decomposition, the volumetric problem is reduced to a problem on the layer interfaces, where the values of the wavefield and its numerical normal derivative at those interfaces are the unknowns. This reduced problem is preconditioned by polarization — decomposing the wavefield into upgoing and downgoing components and propagating each component independently across the domain using a procedure akin to a Rayleigh integral (Berkhout, 1980) at each subdomain. As we will see from numerical experiments in heterogeneous media, when combined with Krylov solvers, the preconditioned system requires a handful of iterations to converge independently of the frequency.

Following Zepeda-Núñez and Demanet (2016), our parallel, scalable solver exploits

- local solvers, using efficient sparse direct solvers at each subdomain
- high-quality transmission conditions between subdomains, implemented via perfectly-matched layers (PML) (Bérenger, 1994; Johnson, 2010), and
- an efficient preconditioner based on polarizing conditions imposed via incomplete Green's integrals.

These concepts combine to yield a global-iterative method that converges in a small number of iterations. The method has two stages: an offline stage that can be precomputed independently of the right-hand sides (RHS) and an online stage that is computed for each RHS or by batch processing.

## Performance context

For current applications, empirical runtimes are a more practical measure of an algorithm's performance than asymptotic complexity. This requirement has led to a recent effort to reduce the runtimes of preconditioners with optimal asymptotic complexity by leveraging parallelism. For example, Poulson et al. (2013) introduce a new local solver, i.e., a solver for the subproblem defined on each layer, which carefully handles communication patterns between layers to obtain impressive timings. Although most sweeping algorithms require visiting each subdomain in sequential fashion, Stolk (2017) introduces a modified sweeping pattern, which changes the data dependencies during the sweeps to improve parallelism. Finally, Zepeda-Núñez and Demanet (2016) introduce the method of polarized traces, which reduces the solver's runtime by leveraging parallelism and fast summation methods. We follow from these developments and seek to improve performance of the method of polarized traces by finding concurrency outside of individual local solves.

To date, most studies focus on minimizing the parallel runtime or complexity of a single solve with a single RHS. However, in the scope of seismic inversion, where there can be tens of thousands of RHS, it is important to consider the overall runtime or complexity of solving for all RHS. In this context, we define linear runtime complexity to be  $\mathcal{O}(RN)$ , where  $N$  is the total number of degrees of freedom and  $R$  is the total number of RHS. For three dimensions,

we define  $N = n^3$  and recall that  $n$  is the number of degrees of freedom in a single dimension of a 3D volume.

In this paper, we present a solver for the 3D high-frequency Helmholtz equation with a sublinear, online, parallel runtime

$$\mathcal{O}(\alpha_{\text{pml}}^2 \max(1, R/L)N \log N), \quad (2)$$

where  $L \sim n$  is the number of subdomains in a layered domain decomposition and  $\alpha_{\text{pml}} \sim \log \omega$  is the number of points needed to implement a high-quality absorbing boundary condition between layers. The offline stage, which we do not detail, has complexity  $\mathcal{O}(\alpha_{\text{pml}}^3 N)$ . We achieve sublinear complexity in the online component by comprehensive parallelization of all aspects of the algorithm, including exploiting parallelism in local solves and by pipelining the RHS. Thus, as long as  $R \sim n$  at minimum, there is a no impact on the asymptotic complexity. In 3D studies, it is typical that the number of sources  $R \sim n^2$  because as frequency and resolution increase, the number of sources in the inline and crossline direction must also increase (Li et al., 2015).

We require that the thickness of each layer, in grid points, is held constant as the problem is scaled. Then, the number of layers must grow with frequency, so  $L \sim n$ . This hypothesis is critical for obtaining a quasilinear complexity algorithm and is related to the complexity of solving a quasi-2D problem using multifrontal methods (for further details, see Engquist and Ying, 2011b; Poulson et al., 2013). Each layer is further extended by  $\alpha_{\text{pml}}$  grid points for the PML. Poulson et al. (2013) document that  $\alpha_{\text{pml}}$  must grow with problem frequency,  $\alpha_{\text{pml}} \sim \log \omega$ , for the PML to remain effective, which is manifested by the number of Krylov solver iterations required for convergence scaling as  $\log \omega$ .

Pipelining, a significant source of the parallel performance of our implementation, for domain-decomposition methods has been previously considered (Stolk, 2017), albeit without complexity claims and without a fully tuned communication strategy between subdomains. Here, we present an empirical study of the effectiveness of pipelining, in the presence of multiple RHS (which scales in a frequency-dependent manner, similar to how it would during an acquisition for FWI), and we compare results with an asymptotic complexity model. We are able to exploit pipelining because the method of polarized traces only requires degrees of freedom at layer interfaces for the bulk of the computation, which reduces the memory footprint per RHS.

Finally, we solve the 3D problem in a high-performance computing (HPC) environment, under the assumption that the number of computing nodes in the HPC cluster is  $\mathcal{O}(n^3 \log(n)/M)$ , where  $M$  is the memory of a single compute node. Thus, following the assumption that there are  $L \sim n$  layers, we use  $\mathcal{O}(n^2 \log(n)/M)$  compute nodes for each layer. The restrictions on node growth come from the fact that computing nodes have finite memory, and thus more nodes are needed to solve larger problems. As the numerical examples will show, using more nodes per layer reduces the runtime per Krylov iteration by enhancing the parallelism of the local solves in each subdomain, provided that a carefully designed communication pattern is used to keep the communication overhead low.

## Related work

Modern linear algebra techniques, in particular nested-dissection methods (George, 1973) and multifrontal solvers (Duff and Reid, 1983; Xia et al., 2010), coupled with  $\mathcal{H}$ -matrices (Boerm et al.,

2006), have been applied to the Helmholtz problem, yielding, for example, the hierarchical Poincaré-Steklov solver (Gillman et al., 2015), solvers using hierarchical semiseparable (HSS) matrices (de Hoop et al., 2011; Wang et al., 2012, 2013), or block low-rank matrices (Amestoy et al., 2015, 2016). Although these methods take advantage of compressed linear algebra to gain more efficiency (e.g., Bebendorf, 2008), in the high-frequency regime, they still suffer from the same suboptimal asymptotic complexity as standard multifrontal methods (e.g., Demmel et al., 1999; Amestoy et al., 2001; Davis, 2004).

Multigrid methods (e.g., Brandt and Livshits, 1997; Aruliah and Ascher, 2002; Laird and Giles, 2002; Erlangga et al., 2006; Mulder, 2006; Sheikh et al., 2013), once thought to be inefficient (approaches stemming from the complex shifted Laplacian; Erlangga et al. [2006] can be advantageous if properly tuned. However, in general, they either require an expensive solver for the shifted problem or require a large number of iterations to reach convergence, depending on the scaling between the complex shift and the frequency [Gander et al., 2015]) for the Helmholtz problem, have been successfully applied to the Helmholtz problem by Calandra et al. (2013) and Stolk (2016). Although these algorithms do not result in a lower computational complexity, their empirical runtimes are impressive due to their highly parallelizable nature. Due to the possibility for efficient parallelization, there has been a renewed interest on multilevel preconditioners, such as the one in Hu and Zhang (2016).

Within the geophysical community, the analytic incomplete LU (AILU) method was explored by Plessix and Mulder (2003) and applied in the context of 3D seismic imaging, resulting in some large computations (Plessix, 2007). A variant of Kaczmarz preconditioners (Gordon and Gordon, 2010) has been studied and applied to time-harmonic wave equations by (Li et al., 2015). Another class of methods, called hybrid direct-iterative methods, has been explored by Sourbier et al. (2011). Although these solvers have, in general, relatively low-memory consumption, they tend to require many iterations to converge, thus hindering practical runtimes.

Domain-decomposition methods for solving PDEs date back to Schwarz (1870), in which the Laplace equation is solved iteratively (for a more recent treatise, see Lions, 1989). The application of domain decomposition to the Helmholtz problem was first proposed by Després (1990). Cessenat and Després (1998) further refine this approach with the development of the ultraweak variational formulation (UWVF) for the Helmholtz equation, in which the basis functions in each element, or subdomain, are solutions to the local homogeneous equation. The UWVF approach motivated a series of related methods, such as the partition of unity method of Babuska and Melenk (1997), the least-squares method of Monk and Wang (1999), the discontinuous enrichment method by Farhat et al. (2001), and Trefftz methods by Gittelsohn et al. (2009) and Moiola et al. (2011), among many others. A recent and thorough review of Trefftz and related methods was performed by Hiptmair et al. (2016).

The results of Lions (1989) and Després (1990) have inspired the development of various domain-decomposition algorithms, which are now classified as Schwarz algorithms (for a review on classical Schwarz methods, see Chan and Mathew, 1994; Toselli and Widlund, 2005; and for other applications of domain-decomposition methods for the Helmholtz equations, see de La Bourdonnaye et al., 1998; Ghanemi, 1998; McInnes et al., 1998; Collino et al., 2000; Magoules et al., 2000; Boubendir, 2007; Astaneh and Guddati, 2016). However, the convergence rate of such algorithms is strongly

dependent on the boundary conditions prescribed at the interfaces between subdomains. Gander et al. (2002) introduce an optimal, nonlocal boundary condition for domain interfaces, which is then approximated by an optimized Robin boundary condition. This last work led to the introduction of the framework of optimized Schwarz methods in Gander (2006) to describe optimized boundary conditions that provides high convergence. The design of better interface approximations was studied by Gander and Kwok (2011), Boubendir et al. (2012), Gander and Zhang (2013, 2014), and Gander and Xu (2016), among many others.

Engquist and Zhao (1998) introduce absorbing boundary conditions for domain-decomposition schemes for elliptic problems, and the first application of such techniques to the Helmholtz problem traces back to the AILU factorization (Gander and Nataf, 2000). The sweeping preconditioner, introduced in Engquist and Ying (2011a, 2011b), was the first algorithm to show that those ideas could yield algorithms with quasilinear complexity. There exist two variants of the sweeping preconditioner, which involved using either  $\mathcal{H}$ -matrices (Engquist and Ying, 2011a) or multifrontal solvers (Engquist and Ying, 2011b) to solve the local problem in each thin layer. These schemes are extended to different discretizations and physics by Tsuji et al. (2012, 2014) and Tsuji and Ying (2012). Since the introduction of the sweeping preconditioner, several related algorithms with similar claims have been proposed, such as the source transfer preconditioner (Chen and Xiang, 2013a, 2013b), the rapidly converging domain decomposition (Stolk, 2013) and its extensions (Stolk, 2017), the double sweep preconditioner (Vion and Geuzaine, 2014), and the method of polarized traces (Zepeda-Núñez and Demanet, 2016).

## Organization

The remainder of this paper is organized as follows: We provide the numerical formulation of the Helmholtz equation, present the reduction to a surface integral equation (SIE), and introduce the method of polarized traces for solving the SIE. Next, we elaborate on the parallelization and communication patterns and examine the empirical complexities and runtimes. We demonstrate our claims empirically with several computational experiments. Finally, prior to concluding, we address some caveats of the method and discuss potential extensions to related problems.

## NUMERICAL FORMULATION

For this study, we discretize equation 1 using the standard second-order finite-difference method on a regular mesh of  $\Omega$ , with a grid of size  $n_x \times n_y \times n_z$  and a grid spacing of  $h$ . Absorbing boundary conditions are imposed on all sides of the domain via PMLs, as described by Bérenger (1994). Although we only address this case, sweeping methods can handle free-surface boundary conditions with a modest, nonasymptotic increase in the number of iterations (Poulson et al., 2013; Stolk, 2013).

We describe our PML implementation in detail because the quality and structure of the PML implementation strongly impact the convergence properties of the method. Following Johnson (2010), the PML is implemented via a complex coordinate stretching (We refer the interested reader to Chew and Weedon [1994] and Komatitsch and Martin [2007]). First, we define an extended domain  $\hat{\Omega}$  such that  $\Omega \subset \hat{\Omega}$  and we extend the Helmholtz operator from equation 1 to that domain as follows:

$$\mathcal{H} = \hat{\Delta} + m\omega^2 \text{in } \hat{\Omega}, \quad (3)$$

where  $m$  is an extension of the squared slowness to  $\hat{\Omega}$  and the extended Laplacian  $\hat{\Delta}$  is constructed by replacing the partial derivatives in the standard Laplacian  $\Delta = \partial_{xx} + \partial_{yy} + \partial_{zz}$  with coordinate-stretched partial derivatives defined on  $\hat{\Omega}$ :

$$\partial_x \rightarrow \beta_x(\mathbf{x})\partial_x, \partial_y \rightarrow \beta_y(\mathbf{x})\partial_y, \partial_z \rightarrow \beta_z(\mathbf{x})\partial_z. \quad (4)$$

The complex dilation function  $\beta_x(\mathbf{x})$  (and similarly  $\beta_y(\mathbf{x})$  and  $\beta_z(\mathbf{x})$ ) is defined as

$$\beta_x(\mathbf{x}) = \frac{1}{1 + i \frac{\sigma_x(\mathbf{x})}{\omega}}, \quad (5)$$

where the PML profile function  $\sigma_x(\mathbf{x})$  (and similarly  $\sigma_y(\mathbf{x})$  and  $\sigma_z(\mathbf{x})$ ) is

$$\sigma_x(\mathbf{x}) = \begin{cases} \frac{C}{\delta_{\text{pml}}} \left( \frac{x}{\delta_{\text{pml}}} \right)^2, & \text{if } x \in (-\delta_{\text{pml}}, 0), \\ 0, & \text{if } x \in [0, L_x], \\ \frac{C}{\delta_{\text{pml}}} \left( \frac{x-L_x}{\delta_{\text{pml}}} \right)^2, & \text{if } x \in (L_x, L_x + \delta_{\text{pml}}), \end{cases} \quad (6)$$

where  $L_x$  is the length of  $\Omega$  in the  $x$ -dimension and  $\delta_{\text{pml}}$  is the length of the extension. In general,  $\delta_{\text{pml}}$  grows slowly with the frequency, i.e.,  $\delta_{\text{pml}} \propto \mathcal{O}(\log \omega)$ , to obtain enough absorption as the frequency increases. The constant  $C$  is chosen to provide enough absorption. In practice,  $\delta_{\text{pml}}$  and  $C$  can be seen as parameters to be tuned for accuracy versus efficiency.

The extended Helmholtz operator provides the definition of the global continuous problem

$$\mathcal{H}u = f_s, \quad \text{in } \hat{\Omega}, \quad (7)$$

which is then discretized using finite differences to obtain the discrete global problem

$$\mathbf{H}u = \mathbf{f}_s. \quad (8)$$

## Local formulation

In the method of polarized traces,  $\Omega$  is decomposed into a set of  $L$  layers,  $\{\Omega^\ell\}_{\ell=1}^L$ . Without loss of generality and to match the physical intuition, we describe the decomposition in the  $z$ -dimension. In practice, however, to avoid excessive reflection in the direction of sweeps and to optimize the layer-communication volume, we construct the sweeps along a dimension orthogonal to the  $z$ -dimension. Each subdomain  $\Omega^\ell$  is extended to include an absorbing region, as above, yielding the extended subdomain  $\hat{\Omega}^\ell$ . For boundaries of  $\hat{\Omega}^\ell$  shared with  $\hat{\Omega}$ , the absorbing layer is considered to be inherited from the global problem. For the intralayer boundaries of  $\Omega^\ell$ , i.e., those due to the partitioning of  $\Omega$ , the extension to the additional absorbing layers in  $\hat{\Omega}^\ell$  are necessary to prevent reflections at layer interfaces, which are detrimental to convergence.

Then, the local Helmholtz problem is

$$\mathcal{H}^\ell v^\ell := \Delta^\ell v^\ell(\mathbf{x}) + m^\ell \omega^2 v^\ell(\mathbf{x}) = f_s^\ell(\mathbf{x}) \quad \text{in } \hat{\Omega}^\ell, \quad (9)$$

where  $m^\ell$  and  $f^\ell$  are the local restrictions of the model parameters and source functions to  $\hat{\Omega}^\ell$ , generated by extending  $m\chi_{\Omega^\ell}$  and  $f\chi_{\Omega^\ell}$  to  $\hat{\Omega}^\ell$ , where  $\chi_{\Omega^\ell}$  is the characteristic function of  $\Omega^\ell$ . The local Laplacian  $\Delta^\ell$  is defined using the same coordinate stretching approach as above, except on  $\hat{\Omega}^\ell$ . As before,  $\delta_{\text{pml}}$ , and thus  $\alpha_{\text{pml}}$ , must scale as  $\log \omega$  to obtain the convergence rate claimed in this paper.

We discretize the local problem in equation 9 resulting in the discrete local Helmholtz system

$$\mathbf{H}^\ell \mathbf{v}^\ell = \mathbf{f}_s^\ell. \quad (10)$$

For the finite-difference implementation in this paper, we assume a structured, equispaced Cartesian mesh with mesh points  $\mathbf{x}_{i,j,k} = (x_i, y_j, z_k) = (ih, jh, kh)$ . Assuming the same ordering as Zepeda-Núñez and Demanet (2016), we write the global solution in terms of the depth index

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_z}), \quad (11)$$

where  $\mathbf{u}_k$  is a plane sampled at constant depth  $z_k$  (In MATLAB notation,  $\mathbf{u}_k = (u_{:,k})$ ).

Let  $\mathbf{u}^\ell$  be the local restriction of  $\mathbf{u}$  to  $\Omega^\ell$ ; i.e.,  $\mathbf{u}^\ell = \chi_{\Omega^\ell} \mathbf{u}$ . Following the above notation,  $\mathbf{u}_k^\ell$  is the local solution trace in the plane at local depth  $z_k^\ell$ . For notational convenience, we renumber the local depth indices so that  $\mathbf{u}_1^\ell$  and  $\mathbf{u}_{n^\ell}^\ell$  are the top and bottom planes of the bulk domain. Points due to the PML are not considered (with this renumbering, the local depth index  $z_k^l$  maps to the global depth index  $z_{n^l+k}$ , where  $n^l = \sum_{j=1}^{l-1} n^j$ ). Finally, we concatenate all the vectors containing the degrees of freedom at the interfaces, resulting in

$$\underline{\mathbf{u}} = (\mathbf{u}_{n^1}^1, \mathbf{u}_1^2, \mathbf{u}_{n^2}^2, \dots, \mathbf{u}_1^{L-1}, \mathbf{u}_{n^{L-1}}^{L-1}, \mathbf{u}_1^L)^t, \quad (12)$$

which is the vector of interface traces for all  $L$  layers. This is illustrated in the fourth panel of Figure 1, where  $\mathbf{u}_{n^1}^1$  are the degrees of freedom above the light-blue dots at the bottom of the top subdomain,  $\mathbf{u}_1^2$  are the degrees of freedom below the light-blue dots at the top of the middle subdomain, and so on.

To map the solution vectors at fixed depth planes back to the discretized whole volume of  $\Omega^\ell$ , we define the Dirac delta at a fixed depth

$$(\delta(z - z_p) \mathbf{v}_q)_{i,j,k} = \begin{cases} 0 & \text{if } k \neq p, \\ \frac{(\mathbf{v}_q)_{i,j}}{h^3} & \text{if } k = p. \end{cases} \quad (13)$$

This definition of the numerical Dirac delta is specific to a classical finite-difference discretization. If the discretization changes, it is still possible to define a numerical Dirac delta using the approach developed in Zepeda-Núñez and Demanet (2018).

The offline computations on local subdomains are detailed in Algorithm 1.

### Reduction to surface integral equation

The global solution is related to the local layer solutions by coupling the subdomains using the Green's representation formula (GRF) within each layer. The resulting SIE, posed at the interface between layers, effectively reduces the problem from the global domain  $\Omega$  to the interfaces between layers. The resulting SIE has the form

$$\underline{\mathbf{M}} \underline{\mathbf{u}} = \underline{\mathbf{f}}, \quad (14)$$

where  $\underline{\mathbf{M}}$  is formed by interface-to-interface Green's functions,  $\underline{\mathbf{u}}$  is defined in equation 12, and  $\underline{\mathbf{f}}$  is the RHS, formed as in line 8 of Algorithm 2, where a high-level sketch of the algorithm to solve the 3D high-frequency Helmholtz equation is given.

The matrix  $\underline{\mathbf{M}}$  is a block banded matrix (Figure 2) of size  $2(L-1)n^2 \times 2(L-1)n^2$ . Theorem 1 of Zepeda-Núñez and Demanet (2016) reveals that the solution of equation 14 is exactly the restriction of the solution of equation 8 to the interfaces between layers.

Consequently, if the traces of the exact solution are known, then it is possible to apply the GRF to locally reconstruct exactly the global solution within each layer. Equivalently, the reconstruction can be performed by modifying the local source with a measure supported on the layer interfaces and solving the local system with the local solver (lines 11–12 of Algorithm 2).

To efficiently solve the 3D problem, it is critical that the matrix  $\underline{\mathbf{M}}$  is never explicitly formed. Instead, a matrix-free approach (Zepeda-

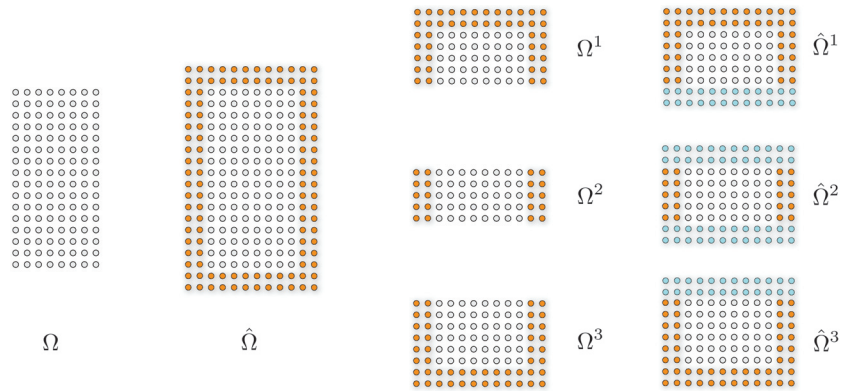


Figure 1. Sketch in 2D of the partition of the domain in layers. The domain  $\Omega$  is extended to  $\hat{\Omega}$  by adding the PML nodes (orange). After decomposition into subdomains  $\Omega^\ell$ , the internal boundaries are padded with extra PML nodes (light blue) resulting in the subdomains  $\hat{\Omega}^\ell$ .

#### Algorithm 1. Offline computation.

- 1: **function**  $[L^\ell, U^\ell] = \text{FACTORIZATION}(m, \omega)$
- 2: **for**  $\ell = 1:L$  **do**
- 3:      $\mathbf{H}^\ell = \Delta^\ell + m^\ell \omega^2$       $\triangleright$  Build the local system
- 4:      $\mathbf{L}^\ell \mathbf{U}^\ell = \mathbf{H}^\ell$       $\triangleright$  Compute the LU factorization
- 5: **end for**
- 6: **end function**

Núñez and Demanet, 2018) is used to apply the blocks of  $\mathbf{M}$  via applications of the local solver, using equivalent sources supported at the interfaces between layers, as shown in Algorithm 3. Moreover, as seen in Algorithm 3, the application of  $\mathbf{M}$  is easily implemented in parallel, with a small communication overhead. The only nonembarrassingly parallel stage of Algorithm 2 is the solution of equation 14, which is inherently sequential.

Given that  $\mathbf{M}$  is never explicitly formed, an iterative method is the natural choice for solving equation 14. In practice, the condition number of  $\mathbf{M}$  is very large and it has a wide spectrum in the complex plane, which implies that a large number of iterations is required to achieve convergence. To alleviate this problem, we apply the method of polarized traces, as an efficient preconditioner for equation 14, which we describe below.

### Preconditioning with polarized traces

Reducing the Helmholtz problem to an SIE allows us to efficiently parallelize most of the computation required to solve equation 8. The only remaining sequential bottleneck is the solution of

#### Algorithm 2. Online computation using the SIE reduction.

```

1: function  $\mathbf{u} = \text{HELMHOLTZ SOLVER } \mathbf{f}$ 
2:   for  $\ell = 1:L$ 
3:      $\mathbf{f}^\ell = \mathbf{f}|_{\Omega^\ell}$  ▷ partition the source
4:   end for
5:   for  $\ell = 1:L$ 
6:      $\mathbf{v}^\ell = (\mathbf{H}^\ell)^{-1}\mathbf{f}^\ell$  ▷ solve local problems
7:   end for
8:    $\underline{\mathbf{f}} = (\mathbf{v}_n^1, \mathbf{v}_1^2, \mathbf{v}_n^2, \dots, \mathbf{v}_1^L)^t$  ▷ form RHS
9:    $\mathbf{M}\underline{\mathbf{u}} = \underline{\mathbf{f}}$  ▷ solve equation 14
10:  for  $\ell = 1:L$ 
11:     $\mathbf{g}^\ell = \mathbf{f}^\ell + \delta(z_1 - z)\mathbf{u}_{n^{\ell-1}}^{\ell-1} - \delta(z_0 - z)\mathbf{u}_1^\ell$   

         $- \delta(z_{n^{\ell+1}} - z)\mathbf{u}_{n^\ell}^\ell + \delta(z_{n^\ell} - z)\mathbf{u}_1^{\ell+1}$ 
12:     $\mathbf{u}^\ell = (\mathbf{H}^\ell)^{-1}\mathbf{g}^\ell$  ▷ inner solve
13:  end for
14:   $\mathbf{u} = (\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^{L-1}, \mathbf{u}^L)^t$  ▷ concatenate
15: end function

```

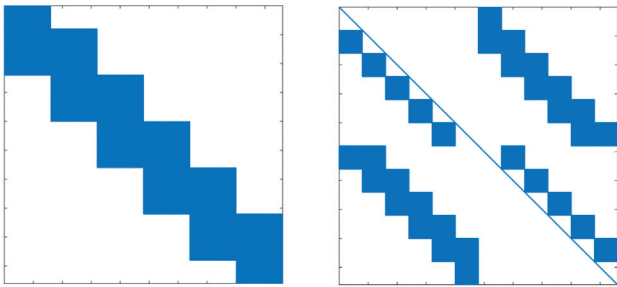


Figure 2. Sparsity pattern of the SIE matrix in equation 14 (left) and the polarized SIE matrix in equation 18 (right).

equation 14. Given the size and the distributed nature of  $\mathbf{M}$ , iterative methods such as generalized minimum residual (GMRES) (Saad and Schultz, 1986) or Bi-CGSTAB (van der Vorst, 1992), are the logical approach for solving equation 14. However, numerical experiments indicate that the condition number of  $\mathbf{M}$  scales as  $\mathcal{O}(h^{-2})$ , or as  $\mathcal{O}(\omega^2)$  in the high-frequency regime (Zepeda-Núñez and Demanet, 2016). The number of iterations required for Krylov methods to converge is usually proportional to the condition number of the system, yielding poor scalability for solving the SIE at high frequencies. To alleviate this problem, we use the method of polarized traces to convert the SIE to an equivalent problem, which is easily preconditioned. This preconditioned system only requires  $\mathcal{O}(\log \omega)$  GMRES iterations (this scaling is empirically deduced, under the assumption that no large resonant cavities are present in the media), i.e., it is essentially independent of the frequency. Here, we provide a high-level review of the method of polarized traces and its implementation and we direct the reader to Zepeda-Núñez and Demanet (2016) for a detailed exposition.

To precondition the SIE (equation 14) with the method of polarized traces, the solution at the interfaces is decomposed in upgoing and downgoing components, such that

$$\underline{\mathbf{u}} = \underline{\mathbf{u}}^\uparrow + \underline{\mathbf{u}}^\downarrow, \quad (15)$$

which defines the polarized wavefield

$$\underline{\mathbf{u}} = \begin{pmatrix} \underline{\mathbf{u}}^\downarrow \\ \underline{\mathbf{u}}^\uparrow \end{pmatrix}. \quad (16)$$

By introducing the polarized wavefield, we have deliberately doubled the unknowns and produced an underdetermined system. To close the system, we impose annihilation, or polarizing, conditions (see section 3 of Zepeda-Núñez and Demanet, 2016) that are encoded in matrix form as

#### Algorithm 3. Application of the boundary integral matrix $\mathbf{M}$ .

```

1: function  $\underline{\mathbf{u}} = \text{BOUNDARY INTEGRAL } (\mathbf{v})$ 
2:    $\tilde{\mathbf{f}}^1 = -\delta(z_{n^1+1} - z)\mathbf{v}_{n^1}^1 + \delta(z_{n^1} - z)\mathbf{v}_1^2$ 
3:    $\mathbf{w}^1 = (\mathbf{H}^1)^{-1}\tilde{\mathbf{f}}^1$ 
4:    $\mathbf{u}_{n^1}^\ell = \mathbf{w}_{n^1}^\ell - \mathbf{v}_{n^1}^\ell$ 
5:   for  $\ell = 2:L-1$ 
6:      $\tilde{\mathbf{f}}^\ell = \delta(z_1 - z)\mathbf{v}_{n^{\ell-1}}^{\ell-1} - \delta(z_0 - z)\mathbf{v}_1^\ell$   

         $- \delta(z_{n^{\ell+1}} - z)\mathbf{v}_{n^\ell}^\ell + \delta(z_{n^\ell} - z)\mathbf{v}_1^{\ell+1}$ 
7:      $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1}\tilde{\mathbf{f}}^\ell$  ▷ inner solve
8:      $\mathbf{u}_1^\ell = \mathbf{w}_1^\ell - \mathbf{v}_1^\ell; \mathbf{u}_{n^\ell}^\ell = \mathbf{w}_{n^\ell}^\ell - \mathbf{v}_{n^\ell}^\ell$ 
9:   end for
10:   $\tilde{\mathbf{f}}^L = \delta(z_1 - z)\mathbf{v}_{n^{L-1}}^{L-1} - \delta(z_0 - z)\mathbf{v}_1^L$ 
11:   $\mathbf{w}^L = (\mathbf{H}^L)^{-1}\tilde{\mathbf{f}}^L$ 
12:   $\mathbf{u}_1^L = \mathbf{w}_1^L - \mathbf{v}_1^L$ 
13: end function

```

$$\underline{\mathbf{A}}^\dagger \underline{\mathbf{u}}^\dagger = 0, \quad \text{and} \quad \underline{\mathbf{A}}^\dagger \underline{\mathbf{u}}^\dagger = 0. \quad (17)$$

Requiring that the solution satisfies both equation 14 and the annihilation conditions yields another equivalent formulation

$$\underline{\mathbf{M}} \underline{\mathbf{u}} = \underline{\mathbf{f}}_s, \quad (18)$$

where

$$\underline{\mathbf{M}} = \begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \\ \underline{\mathbf{A}}^\dagger & \underline{\mathbf{A}}^\dagger \end{bmatrix}, \quad \text{and} \quad \underline{\mathbf{f}}_s = \begin{pmatrix} \underline{\mathbf{f}}_s \\ 0 \end{pmatrix}. \quad (19)$$

Following a series of algebraic operations and permutations (for full details, see Zepeda-Núñez and Demanet, 2016), we obtain an equivalent formulation of the polarized SIE matrix in equation 18

$$\underline{\mathbf{M}} = \begin{bmatrix} \underline{\mathbf{D}}^\dagger & \underline{\mathbf{U}} \\ \underline{\mathbf{L}} & \underline{\mathbf{D}}^\dagger \end{bmatrix}. \quad (20)$$

There exists straightforward, parallel algorithms for applying the subblocks of the block matrices  $\underline{\mathbf{D}}^\dagger$ ,  $\underline{\mathbf{D}}^\dagger$ ,  $\underline{\mathbf{L}}$ , and  $\underline{\mathbf{U}}$ . By construction  $\underline{\mathbf{D}}^\dagger$  and  $\underline{\mathbf{D}}^\dagger$  can be easily inverted using block forward and backward substitution because they are block triangular with identity blocks on their diagonals. The blocks that appear in the sparsity pattern of  $\underline{\mathbf{M}}$  (Figure 2) are a direct manifestation of interactions between the layer interfaces.

Although the resulting block linear system can be solved using standard matrix-splitting iterations, such as block Jacobi iteration or block Gauss-Seidel iteration (Saad, 2003), it is natural to continue to use GMRES to solve the system due to the parallelizable nature of applying the constituent blocks of  $\underline{\mathbf{M}}$ . Particularly, the structure of  $\underline{\mathbf{M}}$  is convenient for using a single iteration of Gauss-Seidel as a preconditioner

$$\underline{\mathbf{P}} \underline{\mathbf{M}} \underline{\mathbf{u}} = \underline{\mathbf{P}} \underline{\mathbf{f}}_s, \quad (21)$$

where the preconditioning matrix is

$$\underline{\mathbf{P}} = \begin{pmatrix} \underline{\mathbf{D}}^\dagger & \mathbf{O} \\ \underline{\mathbf{L}} & \underline{\mathbf{D}}^\dagger \end{pmatrix}^{-1}. \quad (22)$$

In the subsequent sections, we will elaborate on the physical and numerical meanings of the constituent blocks of  $\underline{\mathbf{M}}$  and  $\underline{\mathbf{P}}$ .

### Polarization

The main novelty of the method of polarized traces is due to the polarization conditions, which are encoded in the matrices  $\underline{\mathbf{A}}^\dagger$  and  $\underline{\mathbf{A}}^\dagger$ . The polarizing conditions provide a streamlined way to define an iterative solver using standard matrix splitting techniques, and thus an efficient preconditioner for Krylov methods, such as GMRES.

The polarization conditions are constructed by projecting the solution on two orthogonal sets, physically given by waves traveling upward and downward. Similar constructs are well-known to the geophysics community because methods that decompose wavefields into distinct downgoing and upgoing components are the backbone of several imaging techniques (see Zhang, 2006). Commonly, the decomposition is obtained using discretizations of pseudodifferential operators, which can be interpreted as separating the

wavefield into a set of wave atoms traveling in different directions, which are then propagated accordingly. Methods for decomposing and locally extrapolating directionally decomposed wavefields are well-documented (Wu, 1994; Collino and Joly, 1995; Ristow and Ruehl, 1997; de Hoop et al., 2000).

In our case, we rewrite the decomposition condition as an integral relation between the Neumann and Dirichlet data of the wavefield, which ultimately leads to the annihilation conditions in equation 17. The pair composed of the Neumann and Dirichlet traces should lie within the null space of an integral operator defined on an interface, which allows the decomposition of the total wavefield into the upgoing and downgoing components, with each having a clear physical interpretation. In particular, an upgoing wavefield is a wavefield generated by a source located beneath the interface and it satisfies a radiation condition at positive infinity and a downgoing wavefield is a wavefield generated by a source located above the interface and it satisfies a radiation condition at negative infinity. As detailed in Zepeda-Núñez and Demanet (2016), defining the decomposition in this manner allows us to extrapolate each component in a stable manner using an incomplete Green's integral.

The extrapolation of upgoing components is performed algorithmically by the inversion of  $\underline{\mathbf{D}}^\dagger$  and, in the same fashion, the extrapolation of downgoing components is performed by the inversion of  $\underline{\mathbf{D}}^\dagger$ . Moreover, the application of the operator  $\underline{\mathbf{L}}$  isolates the upgoing reflections due to downgoing waves interacting with the material in each subdomain and similarly for the operator  $\underline{\mathbf{U}}$ .

The application of the preconditioner to a decomposed wavefield

$$\underline{\mathbf{P}} \begin{pmatrix} \underline{\mathbf{v}}^\dagger \\ \underline{\mathbf{v}}^\dagger \end{pmatrix} = \begin{pmatrix} (\underline{\mathbf{D}}^\dagger)^{-1} \underline{\mathbf{v}}^\dagger \\ (\underline{\mathbf{D}}^\dagger)^{-1} \underline{\mathbf{r}} \end{pmatrix}, \quad (23)$$

for  $\underline{\mathbf{r}} = (\underline{\mathbf{v}}^\dagger - \underline{\mathbf{L}}(\underline{\mathbf{D}}^\dagger)^{-1} \underline{\mathbf{v}}^\dagger)$ , can be physically interpreted as follows:

- $(\underline{\mathbf{D}}^\dagger)^{-1} \underline{\mathbf{v}}^\dagger$ : extrapolate the downgoing components by propagating them downward.
- $\underline{\mathbf{r}} = (\underline{\mathbf{v}}^\dagger - \underline{\mathbf{L}}(\underline{\mathbf{D}}^\dagger)^{-1} \underline{\mathbf{v}}^\dagger)$ : compute the local reflection of the extrapolated field and add them to the upgoing components.
- $(\underline{\mathbf{D}}^\dagger)^{-1} \underline{\mathbf{r}}$ : extrapolate the upgoing components by propagating them upward.

### Algorithms

As with the application of  $\underline{\mathbf{M}}$  in Algorithm 3, we construct matrix-free methods for solving  $(\underline{\mathbf{D}}^\dagger)^{-1}$  and  $(\underline{\mathbf{D}}^\dagger)^{-1}$  (Algorithms 4 and 5), where local solves are applied in an inherently sequential fashion. To complete the preconditioner, a matrix-free (and embarrassingly parallel) algorithm for applying  $\underline{\mathbf{L}}$  is given in Algorithm 6. Similar algorithms for applying  $\underline{\mathbf{U}}$ ,  $\underline{\mathbf{D}}^\dagger$ , and  $\underline{\mathbf{D}}^\dagger$ , as well as the complete matrix-free algorithm for applying  $\underline{\mathbf{M}}$ , are provided in Appendix A.

In solving the systems for  $(\underline{\mathbf{D}}^\dagger)^{-1}$  and  $(\underline{\mathbf{D}}^\dagger)^{-1}$ , each application of the local solver is local to each layer, which means that some communication is required to transfer solution updates from one layer to the next. The sequential nature of the method for solving these systems implies that only one set of processors, those assigned to the current layer, is working at any given stage of the algorithm. This is illustrated in Figure 3, where each block represents a local solve and the execution path moves from left to right. As explained in Zepeda-Núñez and Demanet (2016), it is possible to apply  $\underline{\mathbf{D}}^\dagger$

and  $\underline{\mathbf{L}}$  simultaneously, thus decreasing the number of local solves per layer.

#### Physical intuition

We deliberately present the preconditioner in a purely algebraic fashion because it is instructive for implementing the method. However, there is a physical interpretation of the steps in the preconditioner, which we describe below.

As alluded to previously, the application of the preconditioner, and in particular the block back-substitution in Algorithms A-2 and A-3, can be seen as a sequence of depth extrapolation steps. Indeed, lines 4 and 5 in Algorithm 4 are the discrete counterpart of the incomplete Green's integral defined by

$$u^\downarrow(\mathbf{x}) = \int_{\Gamma_{\ell-1,\ell}} (G^\ell(\mathbf{x}, \mathbf{y}) \partial_z u^\downarrow(\mathbf{y}) - \partial_z G^\ell(\mathbf{x}, \mathbf{y}) u^\downarrow(\mathbf{y})) dS_{\mathbf{y}}, \quad (24)$$

which is equivalent to the Rayleigh integral used to extrapolate a wavefield measured in the surface toward the interior of the earth by Berkhout (1980). Likewise, lines 4 and 5 of Algorithm 5 are the discrete counterpart to an upgoing discrete Green's integral.

#### Algorithm 4. Downward sweep, application of $(\underline{\mathbf{D}}^\downarrow)^{-1}$ .

---

```

1: function  $\underline{\mathbf{u}}^\downarrow = \text{DOWNWARD SWEEP } (\underline{\mathbf{v}}^\downarrow)$ 
2:    $\mathbf{u}_{n^1}^{\downarrow,1} = -\mathbf{v}_{n^1}^{\downarrow,1}$ 
3:    $\mathbf{u}_{n^{l+1}}^{\downarrow,1} = -\mathbf{v}_{n^{l+1}}^{\downarrow,1}$ 
4:   for  $\ell = 2:L-1$ 
5:      $\tilde{\mathbf{f}}^\ell = \delta(z_1 - z) \mathbf{u}_{n^{\ell-1}}^{\downarrow,\ell-1} - \delta(z_0 - z) \mathbf{u}_{n^{\ell-1+1}}^{\downarrow,\ell-1}$ 
6:      $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1} \tilde{\mathbf{f}}^\ell$ 
7:      $\mathbf{u}_{n^\ell}^{\downarrow,\ell} = \mathbf{w}_{n^\ell}^\ell - \mathbf{v}_{n^\ell}^{\downarrow,\ell}$ 
8:      $\mathbf{u}_{n^{\ell+1}}^{\downarrow,\ell} = \mathbf{w}_{n^{\ell+1}}^\ell - \mathbf{v}_{n^{\ell+1}}^{\downarrow,\ell}$ 
9:   end for
10:   $\underline{\mathbf{u}}^\downarrow = (\mathbf{u}_{n^1}^{\downarrow,1}, \mathbf{u}_{n^{l+1}}^{\downarrow,1}, \mathbf{u}_{n^2}^{\downarrow,2}, \dots, \mathbf{u}_{n^{L-1}}^{\downarrow,L-1}, \mathbf{u}_{n^{L-1+1}}^{\downarrow,L-1})^t$ 
11: end function

```

---

#### Algorithm 5. Upward sweep, application of $(\underline{\mathbf{D}}^\uparrow)^{-1}$ .

---

```

1: function  $\underline{\mathbf{u}}^\uparrow = \text{UPWARD SWEEP } (\underline{\mathbf{v}}^\uparrow)$ 
2:    $\mathbf{u}_0^{\uparrow,L} = -\mathbf{v}_0^{\uparrow,L}$ 
3:    $\mathbf{u}_1^{\uparrow,L} = -\mathbf{v}_1^{\uparrow,L}$ 
4:   for  $\ell = L-1:2$ 
5:      $\tilde{\mathbf{f}}^\ell = -\delta(z_{n^{\ell+1}} - z) \mathbf{u}_0^{\uparrow,\ell+1} + \delta(z_{n^\ell} - z) \mathbf{u}_1^{\uparrow,\ell+1}$ 
6:      $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1} \tilde{\mathbf{f}}^\ell$ 
7:      $\mathbf{u}_1^{\uparrow,\ell} = \mathbf{w}_1^\ell - \mathbf{v}_1^{\uparrow,\ell}$ 
8:      $\mathbf{u}_0^{\uparrow,\ell} = \mathbf{w}_0^\ell - \mathbf{v}_0^{\uparrow,\ell}$ 
9:   end for
10:   $\underline{\mathbf{u}}^\uparrow = (\mathbf{u}_0^{\uparrow,2}, \mathbf{u}_1^{\uparrow,2}, \mathbf{u}_0^{\uparrow,3}, \dots, \mathbf{u}_0^{\uparrow,L}, \mathbf{u}_1^{\uparrow,L})^t$ 
11: end function

```

---

The quality of the extrapolation depends directly on the quality of the approximation of the local Green's function  $G^\ell$  with respect to the global Green's function. In the reductive case, if the local Green's function is precisely the global Green's function, the method will converge in two iterations (see Gander, 2006). However, this is equivalent to solving the global problem, which is prohibitively expensive. Instead, we compute a local approximation of the Green's function, such that the GRF is valid within the layer only, not globally. As expected with domain-decomposition methods, incorrect local approximations introduce numerical artifacts, which are typically due to truncating the domain in a manner that is inconsistent with the underlying physics. In the method of polarized traces, these issues are mitigated with judicious use of high-order absorbing boundary conditions in the form of PMLs. As a physical consequence, the local Green's function can only see local features within a particular layer. Far-field interactions, reflections induced by material changes in the other layers, will not be observed by the local Green's function and must be handled iteratively, by sequentially sweeping through the domains.

An important consequence of the Green's integral representation is that it completely eliminates the difficulties that most domain-decomposition methods have with seamlessly connecting subdomains together. Rather than assigning data-dependent boundary conditions, the coupling is performed using potentials defined on the physical interfaces, and the absorbing boundary conditions in an extended domain effectively dampen spurious reflections. The transmission conditions given by the discrete GRF are algebraically exact; thus, there is no need for tuning parameters.

## PARALLELIZATION STRATEGIES

The computational effort needed to solve industrial scale 3D problems requires aggressive parallelization and optimization of the algorithm. To obtain a scalable implementation, the algorithm and code must be designed to balance the use and occupancy of three key resources: the computing cores, memory, and communication network. In this section, we describe our parallel implementation of

#### Algorithm 6. Upward reflections, application of $\underline{\mathbf{L}}$ .

---

```

1: function  $\underline{\mathbf{u}}^\uparrow = \text{UPWARD REFLECTIONS } (\underline{\mathbf{v}}^\uparrow)$ 
2:   for  $\ell = 2:L-1$ 
3:      $\mathbf{f}^\ell = \delta(z_1 - z) \mathbf{v}_0^{\uparrow,\ell} - \delta(z_0 - z) \mathbf{v}_1^{\uparrow,\ell}$ 
        $- \delta(z_{n^{\ell+1}} - z) \mathbf{v}_0^{\uparrow,\ell+1} + \delta(z_{n^\ell} - z) \mathbf{v}_1^{\uparrow,\ell+1}$ 
4:      $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1} \mathbf{f}^\ell$ 
5:      $\mathbf{u}_1^{\uparrow,\ell} = \mathbf{w}_1^\ell - \mathbf{v}_1^{\uparrow,\ell}$ 
6:      $\mathbf{u}_0^{\uparrow,\ell} = \mathbf{w}_0^\ell$ 
7:   end for
8:    $\mathbf{f}^L = \delta(z_1 - z) \mathbf{v}_0^{\uparrow,L} - \delta(z_0 - z) \mathbf{v}_1^{\uparrow,L}$ 
9:    $\mathbf{w}^L = (\mathbf{H}^L)^{-1} \mathbf{f}^L$ 
10:   $\mathbf{u}_1^{\uparrow,L} = \mathbf{w}_1^L - \mathbf{v}_1^{\uparrow,L}$ 
11:   $\mathbf{u}_0^{\uparrow,L} = \mathbf{w}_0^L$ 
12:   $\underline{\mathbf{u}}^\uparrow = (\mathbf{u}_0^{\uparrow,2}, \mathbf{u}_1^{\uparrow,2}, \mathbf{u}_0^{\uparrow,3}, \dots, \mathbf{u}_0^{\uparrow,L}, \mathbf{u}_1^{\uparrow,L})^t$ 
13: end function

```

---



the method of polarized traces, with a focus on maximizing the use of these resources.

In the previous section, we formally introduced a matrix-free approach for preconditioning the SIE system on layer interfaces. However, this approach still relies on local solves that are implemented using a direct solver. It is possible to use specially designed iterative local solvers by nesting the method of polarized traces within each layer (Zepeda-Núñez and Demanet, 2018) or a recursive version of the sweeping factorization (Liu and Ying, 2015). However, such an approach would require a complicated code with a very carefully implemented communication pattern. For simplicity, and to broaden the portability of the framework, we use a hybrid approach, in which the local solves use off-the-shelf numerical linear algebra libraries and the polarization is matrix-free. We will address the parallelism on two fronts: intra- and interlayer parallelism.

### Pipelining the many RHS

First, we address parallelism due to the layer decomposition. Primarily, the parallelism across layers is due to the SIE and the preconditioner used to help solve it. There are five trivially parallel (by layer) applications of the local solver: four due to  $\underline{M}$  and one due to the appearance of  $\underline{L}$  in the preconditioner. However, in the preconditioner application, there are sequential bottlenecks due to the applications of  $(\underline{D}^\downarrow)^{-1}$  and  $(\underline{D}^\uparrow)^{-1}$  via block back-substitution. Despite the trivial parallel nature of the other local solver applications, applying the preconditioner using Algorithms 4 and 5 permits work to be done on only one layer at a time, thus forcing most of the computer to remain idle. This is illustrated in the top half of Figure 3, where each blue box represents a local solve and algorithm execution moves from left to right. Supposing that each local solve costs  $\gamma(n)$  time, then following Figure 3, each GMRES iteration can be performed in  $5\gamma(n) + 2L\gamma(n)$ , ignoring communication costs.

To alleviate the sequential bottleneck, we leverage the fact that seismic problems have thousands of RHS and introduce pipelining. Pipelining allows us to process multiple right sides simultaneously, each at different levels of progress through the sweeps, which helps to balance the computational load on the layers, reducing the idle time and increasing the computational efficiency. The pipelining principle is demonstrated in the bottom half of Figure 3, where the boxes represent a local solve and the blue, green, and orange colors indicate distinct RHS. Pipelining allows the layers to perform work for different RHS simultaneously. Indeed, as long as there are at least  $2L$  RHS, the pipeline can be completely full and all available compute resources will be occupied. For the pipelined algorithm, again disregarding communication costs, the runtime of a GMRES iteration is  $5R\gamma(n) + 2(L + R)\gamma(n)$ . Recalling that  $L \sim n$ ,  $R \sim n$ , and  $\gamma(n) = \mathcal{O}(a_{\text{pm}}^2 n^2 \log n)$ , the cost ratio for solving  $R$  RHS compared with one RHS is constant:

$$\frac{5R\gamma(n) + 2(L + R)\gamma(n)}{5\gamma(n) + 2L\gamma(n)} = \mathcal{O}(1). \quad (25)$$

One of the advantages of the method of polarized traces is that the memory requirement to store the intermediate representation of the solution is lower than other methods because it requires solutions for the degrees of freedom involved in the SIE only. Thus, for each RHS, only  $N/q$  data need to be stored, where  $q$  is the thickness of the interface. This reduction in storage, when combined with the

reduced storage due to the relatively small number of GMRES iterations required for convergence, yields a smaller memory footprint for the outer GMRES iteration than methods requiring to update the full volume. It is possible to further reduce the memory footprint by using Bi-CGSTAB instead of GMRES, keeping the computational cost almost constant (Zepeda-Núñez and Demanet, 2018).

### Parallel multifrontal local solves

To obtain good parallel performance, we use a high-performance distributed linear algebra library to solve the local problems within each layer. Due to the sparsity pattern of the linear system at each layer, a typical recipe for the local solves is to reorder the degrees of freedom to increase stability and reduce numerical fill-in, perform a multifrontal factorization, and solve the resulting factorized system with forward- and backward-substitution, often called triangular solves. There exist many techniques to parallelize the multifrontal factorization and the triangular solves (for a recent and extensive review of different techniques for solving sparse systems, see Davis et al., 2016). A popular approach is to use supernodal elimination trees (Ashcraft et al., 1987) defined through nested dissection, which results in highly scalable factorizations (Gupta et al., 1997), albeit with less efficient triangular solves Joshi et al. (1997). To avoid the poor scalability of dense triangular solves due to this approach, Raghavan (1998) introduces a scheme called selective inversion, which is applied by Poulson et al. (2013), specifically for the Helmholtz problem. Although very efficient, using the techniques mentioned before requires lengthy and complex code and we choose to use off-the-shelf libraries, which can be effortlessly changed if necessary.

For the results in this paper, we use STRUMPACK (Rouet et al., 2016) to perform the local solves. STRUMPACK is a state-of-the-art distributed sparse linear solver library that relies on supernodal factorizations (it uses a ULV factorization when the compression is turned on; Chandrasekaran et al., 2006; Xia, 2013), a 2D block-cyclic distribution of the matrix, and a static mapping technique to assign tasks to Message Passing Interface (MPI) processes based on proportional mapping (Pothen and Sun, 1993) to achieve good

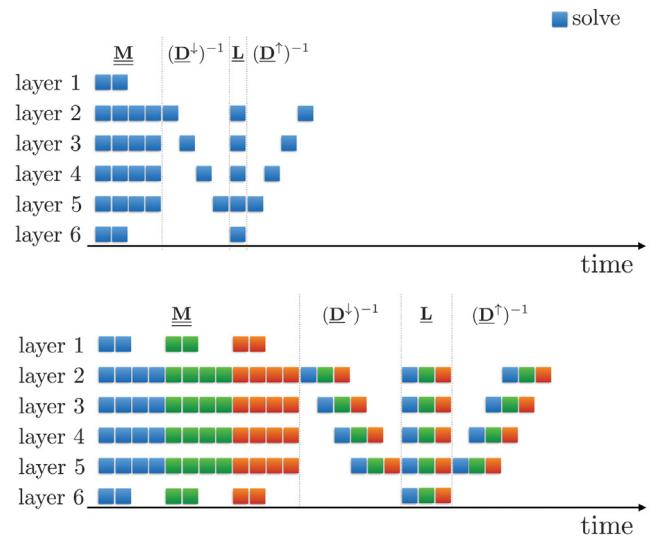


Figure 3. Sketch of the load of each node in the GMRES iteration.

parallel performance. The implementation is competitive with other distributed linear algebra solvers with liberal licenses, such as SuperLU-DIST (Li and Demmel, 2003) and MUMPS (Amestoy et al., 2001), while providing the user more freedom to arrange the distribution of the matrix and RHS, within a distributed memory environment, in a manner that is optimal for specific applications.

A strong advantage of STRUMPACK is that the factorization and solve processes can be accelerated using compressed linear algebra, in particular, HSS compression with nested bases, using randomized sampling techniques. In general, using compressed formulations reduces the memory footprint and in some cases results in faster algorithms. For the high-frequency regime, it is known that solvers based on compression techniques do not provide a lower asymptotic complexity, due to the fact that the ranks of the off-diagonal blocks are frequency dependent (Engquist and Zhao, 2018). However, these solvers still tend to provide smaller memory footprints for the cases that we consider in this paper, albeit, with much bigger runtimes. Therefore, we deliberately do not consider the performance of the adaptive compression in this paper and leave such treatment for future work.

The STRUMPACK solver addresses parallelism within layers in two ways: classical distribution of tasks with MPI and synchronous processing within each task with OpenMP, which we exploit for the results presented in our numerical results. STRUMPACK's hybrid parallelism model allows us to maximize the use of computational resources. We have designed the distribution of MPI tasks to exploit highly asynchronous communication patterns, thus reducing the communication time substantially.

Finally, in most of the experiments shown in the sequel, we process at most one RHS per layer. It is possible to solve more than one RHS per layer and take advantage of BLAS3 routines. Moreover, a

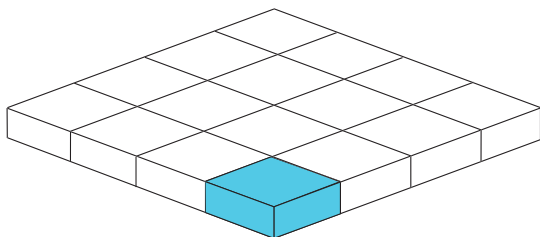


Figure 4. Sketch of the decomposition of the degrees of freedom of the slabs in cubes.

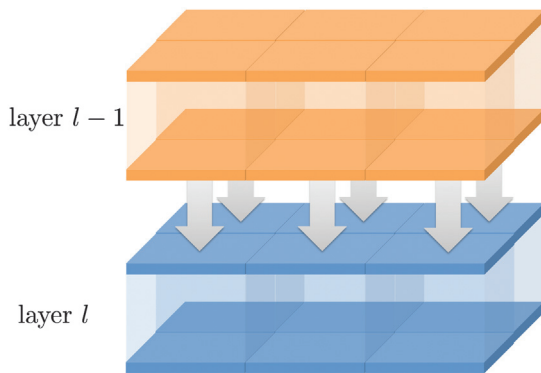


Figure 5. Sketch of the asynchronous communication between slabs.

quick inspection to the algorithms within the preconditioner shows that the RHS are sparse. Indeed, the sources are supported on the interfaces, and the solutions are only needed at the boundaries. In principle, it is possible to take advantage of the sparsity of the solution and the RHS to reduce the constants by removing some branches from the elimination tree. These techniques would certainly reduce the constants, but they would have little impact on the asymptotic scaling; hence, they were not explored in this study. Support for these features is present in current revisions of the high-performance linear algebra package, MUMPS (Amestoy et al., 2001); however, at the time of development of our solver code, the communication pattern and the data locality at each node was not flexible enough to obtain the scalings reported in this paper.

### Communication pattern

As is common in massively parallel applications, there is a bottleneck due to the communication between parallel tasks, which is strongly dependent on the distribution of the tasks on the cluster. For this implementation, we assume a very simple topology for the distribution of unknowns. We distribute the parallel tasks following the layer structure. For each slab of unknowns, we assign  $\mathcal{O}(n^2)$  tasks, and using MPI directives enforce that the tasks are contiguous within physical computing nodes. Each slab is divided into  $\mathcal{O}(n^2)$  cubes, as illustrated in Figure 4, and the parallel tasks associated with that slab are divided evenly and contiguously amongst the cubes. Each cube contains a contiguous block of the solution, as shown in Figure 4, and the associated entries of the local matrix.

Within the slab, the topology is designed such that each cube only communicates with its neighboring cubes in the same slab, generally inside the local solver. Across slabs, cubes only communicate with the cube in the same position on the adjacent slabs immediately above and below it, as illustrated in Figure 5. Under this particular topology, we can distinguish two main communications bottlenecks:

- 1) the communication between parallel tasks within the distributed linear algebra solver, and
- 2) the communication of the boundary data between slabs, during the application of the preconditioner.

As a consequence of using third-party distributed linear algebra solvers, we have little control over the communication pattern, particularly because STRUMPACK uses MC64 (Duff and Koster, 2001) for enhancing stability and ParMetis (Karypis and Kumar, 1998) to optimally reorder the matrix to reduce fill-in during the factorization. To have the desired distribution of the degrees of freedom among the cubes, we reorder the matrix with a Z-ordering (Asano et al., 1997), so that smallest division corresponds exactly to the degrees of freedom within a cube. Then, the matrix is then assembled and passed to the linear solver in a distributed fashion.

Communication between slabs is a product of the application of the preconditioner in equation 22, in which back- and forward-substitution are used to apply  $(\mathbf{D}^\dagger)^{-1}$  and  $(\mathbf{D}^\dagger)^{-1}$ . Algorithms A-2 and A-3 require a local solve in each slab, followed by communication of the trace information to the next slab, in which another local solve is performed. This operation is repeated until all slabs are visited within the sweep.

By dividing the slabs into cubes, the communication of the trace information between slabs is very efficient. Given that each cube communicates with the cube directly above and below, it is possible

to perform asynchronous point-to-point communication between the cubes of two adjacent slabs, as shown in Figure 5. This allows the communication to be performed in nearly constant time, up to saturation of the network. Moreover, the trace information is already distributed for distributed assembly of the RHS within the subsequent slab. As stated before, it is possible to use topologies better suited for the multifrontal solver, such as the one due to Poulson et al. (2013). However, such an implementation requires a very precise understanding of the reordering mechanism within the solver, which we do not generally have for black-box solvers.

## COMPLEXITY OF POLARIZED TRACES

The runtime complexity of the polarized traces preconditioner is driven by the costs of computation and communication. Achieving optimal performance requires delicately balancing the parallel distribution of the problem depending on the characteristics of the target HPC system. In this section, we develop models for computation and communication costs, which guide problem parameter selection in HPC environments.

### Computation

As before, each layer has  $\mathcal{O}((n_z + \alpha_{\text{pml}}) \times n^2)$  grid points; i.e., they are  $n_z$  grid points thick with  $\alpha_{\text{pml}}$  additional points due to the PML. We have that  $n_z = \mathcal{O}(1)$  because  $L \sim n$ , which implies that we are solving a quasi-2D problem. The additional cost is due only to the points used to implement the absorbing boundary conditions. When applying 2D nested dissection to the quasi-2D problem, we have  $\mathcal{O}(\alpha_{\text{pml}} n)$  degrees of freedom in the biggest front, thus leading to a complexity of  $\mathcal{O}(\alpha_{\text{pml}}^3 n^3)$  for the factorization of the systems local to each layer and  $\mathcal{O}(\alpha_{\text{pml}}^2 n^2 \log n)$  for the application of the triangular solver (Duff and Reid, 1983). Sequentially, the complexity of Algorithm 1 is  $\mathcal{O}(\alpha_{\text{pml}}^3 N^{4/3})$ , but given that the loop in line 2 of Algorithm 1 is embarrassingly parallelizable, Algorithm 1 can be performed in  $\mathcal{O}(\alpha_{\text{pml}}^3 N)$  time (as will be seen in the numerical experiments, this scaling can be further reduced due to the parallelism at the level of the multifrontal solver). Due to the sequential nature of Algorithms 4 and 5, applying the preconditioner requires  $2L$  local solves per iteration, applied sequentially. Consequently, the total complexity for the application of the preconditioner is  $\mathcal{O}(\alpha_{\text{pml}}^2 N \log N)$ . For  $\alpha_{\text{pml}} \sim \log n$ , at most  $\mathcal{O}(\log n)$  iterations are empirically needed to converge; thus, the complexity of the solver is linear (up to polylogarithmic factors), provided that  $L \sim n$  and that the number of iterations for convergence grows slowly.

It is possible to relax the restriction that  $L \sim n$ , instead allowing  $L \sim n^b$ , where  $b < 1$ . However, in this regime, maintaining the overall linear complexity requires that we exchange the multifrontal solver for an iterative solver (Liu and Ying, 2015; Zepeda-Núñez and Demanet, 2018). The main disadvantage of this approach is that it reduces the possible parallelism due to using multifrontal solvers, which makes an efficient implementation of the pipelining difficult and makes the communication patterns more complicated.

### Pipelining

We introduced pipelining of  $R$  RHS to alleviate the sequential nature of applying the preconditioner. To understand the runtime impact of pipelining multiple RHS, we consider its impact on the complexity of applying  $\mathbf{M}$  and applying the preconditioner.

Given that the runtime cost of each local solve is  $\mathcal{O}(\alpha_{\text{pml}}^2 n^2 \log n)$  and recalling that applying  $\mathbf{M}$  is embarrassingly parallel, the cost of applying  $\mathbf{M}$  to  $R$  RHS is  $\mathcal{O}(\alpha_{\text{pml}}^2 R n^2 \log n)$ . As long as  $R \sim L \sim n$ , applying the preconditioner costs  $\mathcal{O}(L \alpha_{\text{pml}}^2 n^2 \log n)$ . However, when  $R \gtrsim L$ , the additional RHS are treated sequentially, resulting in a cost of  $\mathcal{O}(\alpha_{\text{pml}}^2 R n^2 \log n)$ . Using the fact that  $L \sim n$  and that  $N = n^3$ , we obtain the advertised runtime of  $\mathcal{O}(\alpha_{\text{pml}}^2 \max(1, R/L) N \log N)$ .

### Communication

We treat the distributed linear algebra solver as a black box; therefore, we do not analyze the costs of communication within the local solve. Thus, we only consider the cost of communication due to the global solve, that is, the costs of communicating between subdomains across layers. We assume that each subdomain has fast access to its corresponding patch of the  $R$  wavefields and the  $R$  sources. Moreover, we assume that each subdomain assembles and stores its portion of the global solution (approaches stemming from the complex shifted Laplacian; Erlangga et al., 2006) can be advantageous if properly tuned. However, in general, they either require an expensive solver for the shifted problem or require a large number of iterations to reach convergence, depending on the scaling between the complex shift and the frequency (Gander et al., 2015).

The offline stage of the algorithm, assembly of the local matrices, and the local factorizations, are embarrassingly parallel under the assumptions described above. The online part has three stages:

- 1) the preparation of the RHS (lines 2–8 in Algorithm 2),
- 2) the solve of the SIE (lines 9 in Algorithm 2), and
- 3) the assembly of the global solution (lines 10–14 in Algorithm 2).

Of these, the first and third stages require no communication under the above assumptions.

Solving the SIE has two main phases: the application of  $\mathbf{M}$  and the application of the preconditioner. Applying  $\mathbf{M}$ , as shown in Algorithm 3, is an embarrassingly parallel operation with zero communication. On the other hand, applying the preconditioner, which is fully sequential, requires communication of  $\mathcal{O}(n^2)$  unknowns from one layer to the next. In our implementation, these unknowns are distributed evenly between  $\mathcal{O}(n^2)$  MPI tasks. Using a point-to-point communication strategy, each of the MPI tasks assigned to a layer only communicates with one corresponding MPI task in the adjacent (above and below) layers, as illustrated in Figure 5. Thus, by exploiting asynchronous communication, the communication between layers can be performed in  $\mathcal{O}(1)$  time up to saturation of the bandwidth, which is asymptotically negligible with respect to solving the local linear systems. This communication must be performed  $\mathcal{O}(L)$  times during each sweeping operation in Algorithms 4 and 5. Consequently, the total communication cost of applying the preconditioner is  $\mathcal{O}(n)$ , up to the saturation of the bandwidth.

## NUMERICAL EXPERIMENTS

In this section, we present the results of several numerical experiments used to verify the complexity described above. In particular, we demonstrate the performance of the 3D preconditioner in various heterogeneous media for a single source and then illustrate the impact of pipelining on parallel performance. Our polarized traces implementation is written in C and compiled with the 2015 Intel compiler

suite. The current implementation uses the IEEE double-precision floating point. To perform the local solves, we use STRUMPACK v1.1.0 with Intel MKL support for fast linear algebra operations. The preconditioner is parallelized with MPI, and STRUMPACK is parallelized with MPI and OpenMP. The experiments were performed on Total's "Laure" SGI ICE-X cluster, where each computing node contains dual eight-core Intel Sandy Bridge processors, 64GB of RAM and are connected with an Infiniband interconnect.

### Homogeneous media

First, we demonstrate the effectiveness of the preconditioner by solving the Helmholtz problem in homogeneous media. For this model, and all non-SEAM synthetic models, the domain and the wavespeed are rescaled and nondimensionalized, so that the length of the domain and the slowest wavespeed are one. With no reflectors in the medium, the convergence of the algorithm is only dependent on the frequency and the quality of the absorbing boundary condition at the layer interfaces. In this experiment, as well as the subsequent experiments,  $\alpha_{\text{pml}}$  has sufficient points to minimize artificial reverberations while simultaneously keeping the number of iterations low. For higher frequencies, to preserve the low iteration count, we would need to scale  $\alpha_{\text{pml}}$  as  $\mathcal{O}(\log n)$ . In practice, we add one point each time we double the frequency starting from four points at lowest frequency considered.

In this experiment, we test four sizes,  $n = 50, 100, 200,$  and  $400$ , which corresponds to frequencies of 8, 16, 32, and 64 Hz. The source frequency is scaled with the problem size to stay in the high-frequency regime, and sources are assumed to be point sources. The number of layers is also scaled with the problem,  $L = 5, 10, 20,$  and  $40$ . Due to memory limitations on the computing node, in some cases, the nodes were saturated before all cores could be assigned to an MPI task. For these cases, we allow the remaining cores to be used

for multithreaded processing with OpenMP. The outer GMRES iteration terminates when the residual of the preconditioned system is reduced to  $10^{-7}$ , which is excessive in a production, single-precision environment; however, it shows the favorable behavior of the solver under more challenging conditions. Lower tolerances can produce misleading results when frequencies are not high enough, only revealing a preasymptotic behavior. For each configuration, we report the wall-clock times for initialization, matrix assembly, matrix factorization, and total online time for  $R = 1$  and  $R = L$  with pipelining. For each experiment,  $R$  is large enough to fill the pipeline and demonstrate that we are in the asymptotic regime. Additionally, we track the number of GMRES iterations required to achieve the desired convergence.

The solution at 64 Hz is provided in Figure 6. The full results of the experiment are given in Table 1 and the observed runtimes, compared with the theoretical runtimes for  $R = 1$  and  $R = L$  pipelined RHS are shown in Figures 7 and 8, respectively. Only when  $R \geq L$  does the theoretical scalability break the linear threshold; however, in both cases, the method of polarized traces scales better than the theoretical scaling, which we attribute to optimizations and parallelism in the local solver. Of note in Table 1, the number of GMRES iterations grows very slowly with the frequency, even when we do not scale  $\alpha_{\text{pml}}$  optimally. In the experiments in which hybrid parallelism (MPI-OpenMP) is used, the runtimes are reduced almost linearly for medium-sized problems; however, the improvements fade as the size of the problem increases. This behavior is due to the linear solver, whereas for large problems, the memory access time in the triangular solves becomes dominant, reducing the parallelism.

### Smooth heterogeneous media

Using the same configurations as above, we solve the Helmholtz problem in the smoothed random media shown in Figure 9 to

**Table 1. Runtime (in seconds) for one and several RHS for the solution of the Helmholtz equation using a homogeneous model.**

$N$	$50^3$	$100^3$	$100^3$	$200^3$	$200^3$	$400^3$	$400^3$	$400^3$
$L$	5	10	10	20	20	40	40	40
MPI tasks	5	10	10	80	80	640	640	640
OpenMP threads per task	1	1	2	1	2	1	2	3
Total cores	5	10	20	80	160	640	1280	1920
Total nodes	1	1	2	5	10	80	80	128
				Single RHS				
# GMRES iterations	4	4	4	5	5	6	6	6
Initialization (s)	0.2	1.0	0.9	6.9	4.4	18.9	18.9	18.4
Factorization (s)	4.1	41.1	21.9	153.2	78.3	320.5	200.1	148.6
Online (s)	4.0	39.2	22.6	182.0	109.7	696.6	401.4	315.5
Average GMRES (s)	0.9	8.4	4.8	32.0	19.2	103.5	59.3	46.6
				Pipelined RHS				
$R$ (number of RHS)	5	10	10	20	20	40	40	40
Online (s)	15.8	189.4	106.2	1255.5	668.5	3994.2	2654.4	1878.1
Average GMRES (s)	3.4	40.6	22.7	223.8	118.6	599.9	401.0	283.0
Online per RHS (s)	3.2	18.9	10.6	62.8	33.4	99.9	66.4	47.0
Average GMRES per RHS (s)	0.7	4.1	2.3	11.2	5.9	15.0	10.0	7.1

demonstrate the effectiveness of the solver in heterogeneous media. This test demonstrates that the method is particularly robust to media, where the rays can bend and develop caustics. The solution for the configuration equivalent to that of Figure 6 is given in Figure 10. In Figure 10, it is clear that the features of the model are comparable to the wavelength used, thus the solution presents interference, caustics, nonspherical wavefronts. Table 2 contains the complete experimental results, where we observe that the variation in the media has little real effect on the runtime or convergence properties. In this case, even if the rays bend, the preconditioner does a remarkable job at tracking the rays in the correct direction and propagating them accordingly. The wall times are shown in Figures 7 and 8.

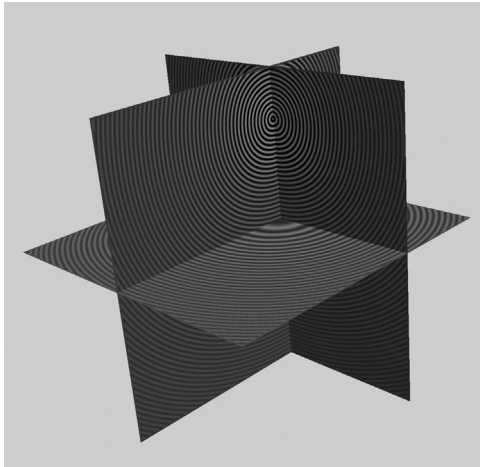


Figure 6. Solution of the Helmholtz equation, at 64 Hz, in constant wavespeed 1 (unitless).

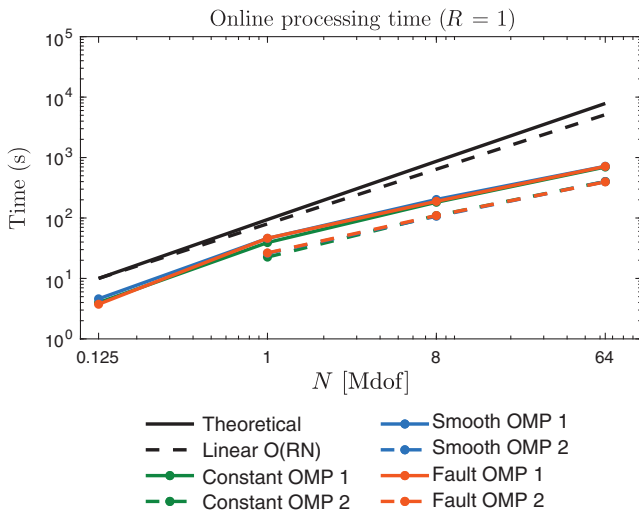


Figure 7. Observed runtime as a function of  $N$  for homogeneous media (green), smooth heterogeneous media (blue), and the “fault” model (orange), with pure MPI (solid) and hybrid MPI-OpenMP (dashed) for  $R = 1$  RHS. For comparison, theoretical scaling of polarized traces algorithm is given (solid black) as well as linear scaling (dashed black).

### Fault model

In general, iterative methods are very sensitive to discontinuous media. At a high frequency, interaction with short-wavelength structures, such as discontinuities, increases the number of reflections. Each additional reflection requires additional iterations to convergence, hindering the efficiency of iterative methods.

Using the same configuration as for the homogeneous model, with the discontinuous velocity given in Figure 11, we demonstrate that the method of polarized traces deteriorates only marginally as a

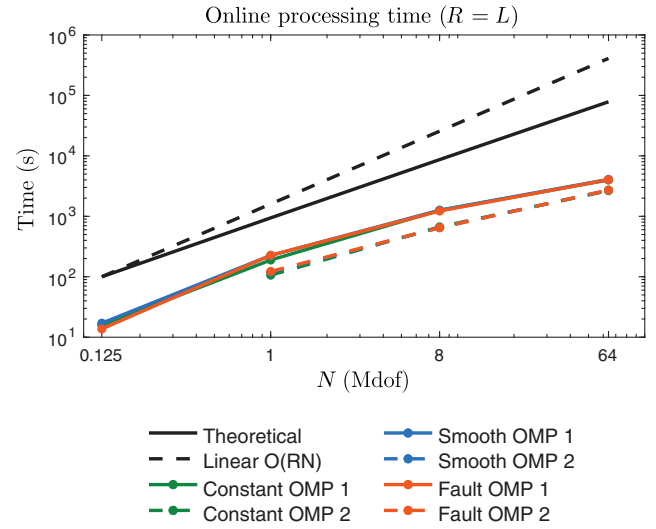


Figure 8. Observed runtime as a function of  $N$  for homogeneous media (green), smooth heterogeneous media (blue), and the fault model (orange), with pure MPI (solid) and hybrid MPI-OpenMP (dashed) for  $R = L$  RHS. For comparison, theoretical scaling of polarized traces algorithm is given (solid black) as well as linear scaling (dashed black).

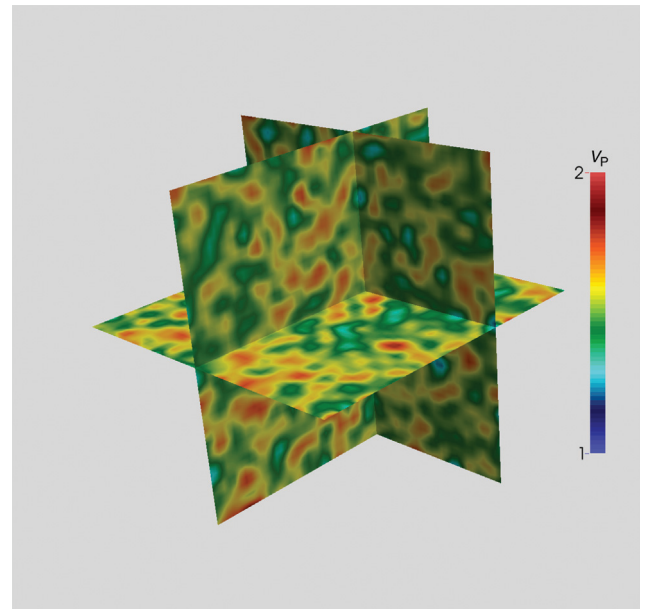


Figure 9. Randomly generated smooth heterogeneous medium.

function of the frequency and number of subdomains. A solution at 64 Hz is given in Figure 12, and the runtime scalability is again given in Figures 7 and 8. As shown in Table 3, we observe the same behavior as in the previous cases and that the strong reflection is handed efficiently by the transmission and polarizing conditions.

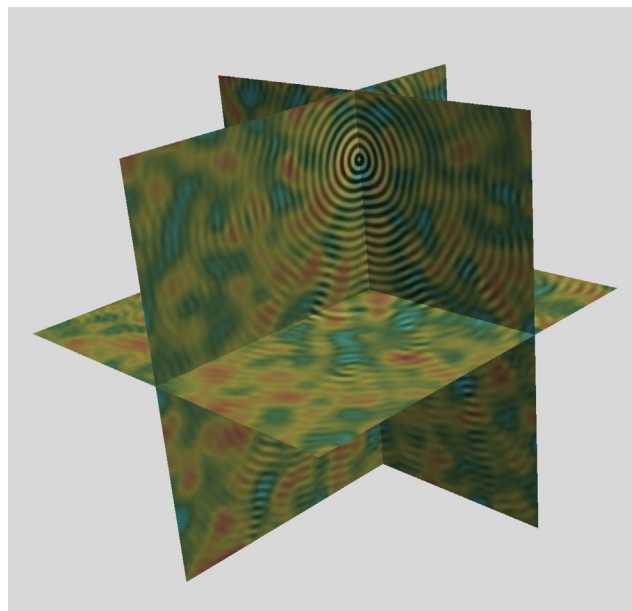


Figure 10. Real part of the solution to the Helmholtz equation using the random smooth medium.

## SEAM model

Beyond mere sensitivity to discontinuities of the medium, iterative solvers are highly sensitive to the roughness and heterogeneity of the velocity model, due to the complexity of interactions, reflections, and drastic changes of direction of waves in the presence of high gradients in the wavespeed. However, the performance method of polarized traces degrades only marginally for highly hetero-

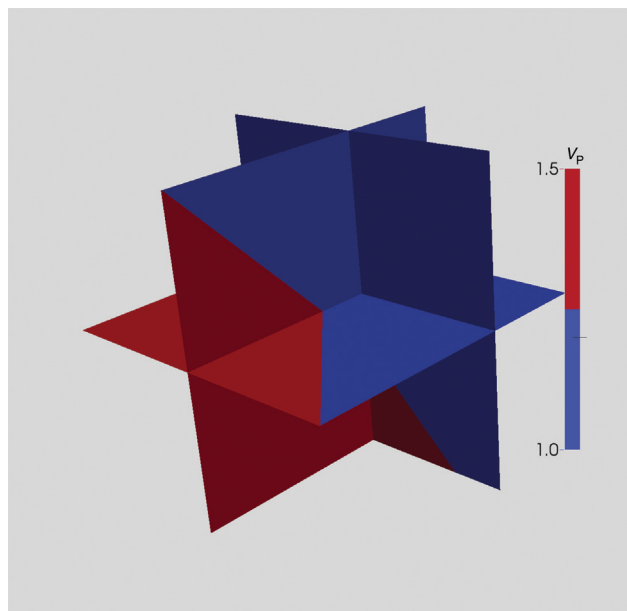


Figure 11. Simple fault model.

**Table 2. Runtime (in seconds) for one and several RHS for the solution of the Helmholtz equation for the smooth model in Figure 9.**

$N$	$50^3$	$100^3$	$100^3$	$200^3$	$200^3$	$400^3$	$400^3$	$400^3$
$L$	5	10	10	20	20	40	40	40
MPI tasks	5	10	10	80	80	640	640	640
OpenMP threads per task	1	1	2	1	2	1	2	3
Total cores	5	10	20	80	160	640	1280	1920
Total nodes	1	1	2	5	10	80	80	128
			Single RHS					
# GMRES iterations	5	5	5	5	5	6	6	6
Initialization (s)	0.2	1.1	1.0	7.3	4.6	21.3	21.2	20.8
Factorization (s)	3.8	41.1	21.8	156.0	79.4	323.7	204.5	151.5
Online (s)	4.6	45.9	26.1	202.2	106.9	717.0	400.1	314.5
Average GMRES (s)	0.8	8.1	4.6	35.5	18.7	106.4	59.2	46.5
			Pipelined RHS					
$R$ (number of RHS)	5	10	10	20	20	40	40	40
Online (s)	17.1	225.1	118.8	1260.9	650.2	4085.0	2714.8	1872.1
Average GMRES (s)	3.0	39.8	20.9	223.6	115.6	613.3	409.2	281.9
Online per RHS (s)	3.4	22.5	11.9	63.0	32.5	102.1	67.9	46.8
Average GMRES per RHS (s)	0.6	4.0	2.1	11.2	5.8	15.3	10.2	7.0

geneous media, except in the presence resonant cavities, where the increase in number of reflections leads to an increase in iterations required for convergence (for a detailed study of the limitations of the method of polarized traces, see [Zepeda-Núñez and Demanet, 2018](#)). We demonstrate this desirable performance on the SEAM Phase I velocity model ([Fehler and Keliher, 2011](#)), shown in Figure 13, which contains rough heterogeneities such as a complex salt body and several stratified regions. In this experiment, we test three

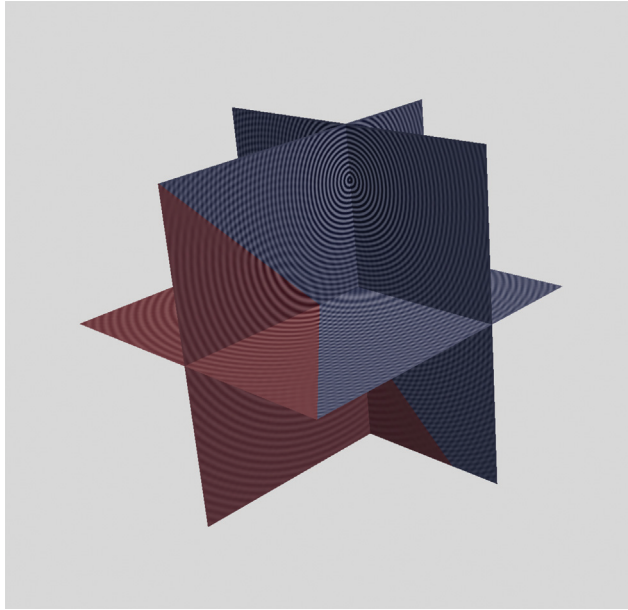


Figure 12. Real part of the solution to the Helmholtz equation using the simple fault model.

problem sizes,  $N = 0.65$ ,  $5.16$ , and  $41.2M$  degrees of freedom, which use  $L = 12$ ,  $24$ , and  $48$  layers, respectively. The remainder of the experimental setup is unchanged. An example solution, plotted over the velocity model, is given in Figure 14.

As seen in the data in Table 4 and plotted in Figures 15 and 16, the runtimes are sublinear with respect to the total number of unknowns. Interestingly, we also observe a sublinear runtime in the offline stages of the algorithm, which we attribute to the parallelism

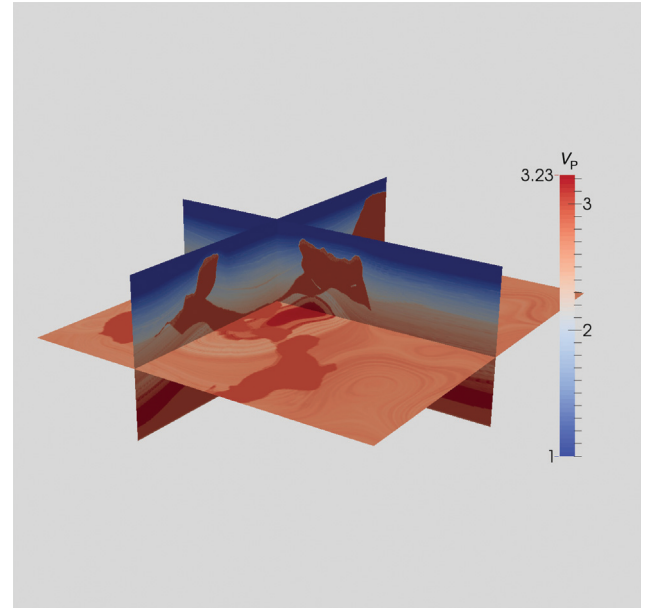


Figure 13. Plot of the SEAM model.

**Table 3. Runtime (in seconds) for one and several RHS for the solution of the Helmholtz equation for the fault model.**

$N$	$50^3$	$100^3$	$100^3$	$200^3$	$200^3$	$400^3$	$400^3$	$400^3$
$L$	5	10	10	20	20	40	40	40
MPI tasks	5	10	10	80	80	640	640	640
OpenMP threads per task	1	1	2	1	2	1	2	3
Total cores	5	10	20	80	160	640	1280	1920
Total nodes	1	1	2	5	10	80	80	128
Single RHS								
# GMRES iterations	4	5	5	5	5	6	6	6
Initialization (s)	0.4	1.1	1.0	7.3	4.7	20.4	20.3	21.0
Factorization (s)	3.8	40.4	22.1	152.2	79.9	317.6	199.5	152.5
Online (s)	3.7	46.2	26.2	188.5	109.8	713.2	395.8	315.6
Average GMRES (s)	0.8	8.1	4.6	33.0	19.2	106.2	58.7	46.5
Pipelined RHS								
$R$ (number of RHS)	5	10	10	20	20	40	40	40
Online (s)	13.7	226.7	122.4	1222.7	647.1	4031.6	2710.6	1838.9
Average GMRES (s)	2.9	40.1	21.6	216.5	114.7	605.0	409.9	276.3
Online per RHS (s)	2.7	22.7	12.2	61.1	32.4	100.8	67.7	46.0
Average GMRES per RHS (s)	0.6	4.0	2.2	10.8	5.7	15.1	10.2	6.9

in the multifrontal factorization. This is expected because the factorization is more computationally intensive than memory intensive. In the more memory-intensive, and thus less parallel, solve phase, we still see an improvement over the theoretical curve, but the improvement is less pronounced.

Finally, for the largest test case, we demonstrate the impact of pipelining by comparing the scalability of our method with the theo-

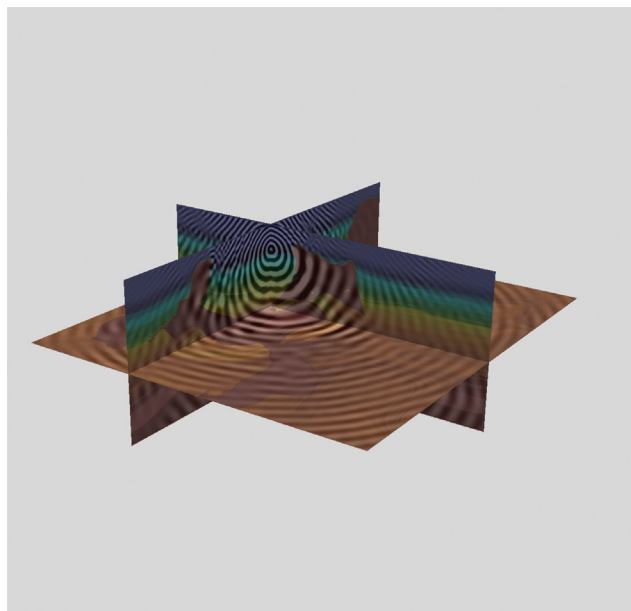


Figure 14. Real part of the solution of the Helmholtz equation using the SEAM model.

retical scalability, as a function of  $R$ . As shown in Figure 17, experimental results indicate that we obtain the expected scalability. A slight divergence from the theoretical curve is expected once the pipeline is fully saturated because the theoretical curve does not take into account the cost of filling and flushing the pipeline.

Finally, Figure 17 depicts the behavior of the pipelining as we increase the number of RHS. As expected, as we add more and more RHS to be solved simultaneously, the runtime per RHS decreases, until the pipeline is full, when the average runtime remains almost constant.

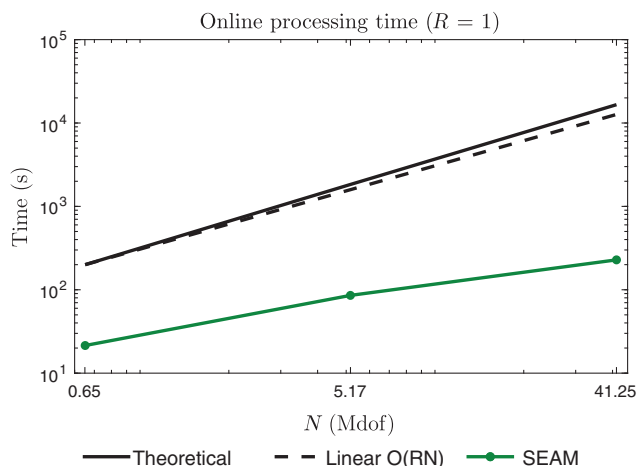


Figure 15. Observed runtime as function of  $N$  for the SEAM model for  $R = 1$  RHS. For comparison, we show the theoretical scaling of the polarized trace algorithm (solid black), as well as a linear scaling (dashed black).

**Table 4. Runtime (in seconds) for one and several RHS for the solution of the Helmholtz equation for the SEAM model. The application of the polarized system defined in equation 18 is achieved by applying each block as shown in Algorithm A-1.**

$N$	$6.51 \cdot 10^5$	$5.16 \cdot 10^6$	$4.12 \cdot 10^7$	$4.12 \cdot 10^7$
$L$	12	24	48	48
MPI tasks	12	48	384	384
OpenMP threads per task	1	2	2	3
Total cores	12	96	768	1152
Total nodes	1	6	77	77
	Single RHS			
# GMRES iterations	4	5	6	6
Initialization (s)	0.6	2.3	10.4	10.7
Factorization (s)	15.2	46.5	111.4	97.9
Online (s)	21.4	85.6	269.8	228.4
Average GMRES (s)	4.6	14.9	40.0	33.7
	Pipelined RHS			
$R$ (number of RHS)	12	24	48	48
Online (s)	106.3	474.8	1527.1	1415.4
Average GMRES (s)	22.8	83.9	229.4	212.9
Online per RHS (s)	8.8	19.8	31.8	29.5
Average GMRES per RHS (s)	1.9	3.5	4.8	4.4



## DISCUSSION

## Accuracy of the global solution

Although the focus of this paper is on solving equation 3, for completeness, we provide an overview of challenges associated with the discretization.

From the Shannon-Nyquist sampling theorem, an oscillatory function at frequency  $\omega$  requires  $\mathcal{O}(\omega^d)$  degrees of freedom to be accurately represented, without aliasing. For example, to accurately represent the solution of equation 1, only  $\mathcal{O}(\omega^3)$  degrees of freedom are required. At this limit, the accuracy is still limited by the error in the discretization of the differential operator. Even if the medium is very smooth, standard methods based on finite differences and finite elements are subject to pollution error: The ratio between the error of the numerical approximation and the best approximation cannot be bounded independently of  $\omega$  (Babuska et al., 1995; Ihlenburg and Babuska, 1995; Babuska and Banerjee, 2012).

The direct consequence of pollution error is that the approximation error, i.e., the error between the analytical and the numerical solution to the linear system, increases with the frequency, even if  $n \sim \omega$ . Thus, to obtain a bounded approximation error independent of the frequency, one must oversample the wavefield, relative to the Shannon-Nyquist criterion —  $n$  must grow faster than  $\omega$ . Unfortunately, oversampling provides discretizations with a suboptimal number of degrees of freedom with respect to the frequency.

To alleviate pollution error, several new approaches have been proposed, which can be broadly classified into two groups:

- 1) methods using standard polynomial bases with modified variational formulations (Goldstein, 1986; Melenk and Sauter, 2011; Melenk et al., 2013; Moiola and Spence, 2014; Graham et al., 2015)
- 2) methods based on well-known variational formulations but using a nonstandard basis, such as plane waves (Hiptmair et al., 2016; Perugia et al., 2016), polynomials modulated by plane waves (Betcke and Phillips, 2012; Nguyen et al., 2015), or other specially adapted functions.

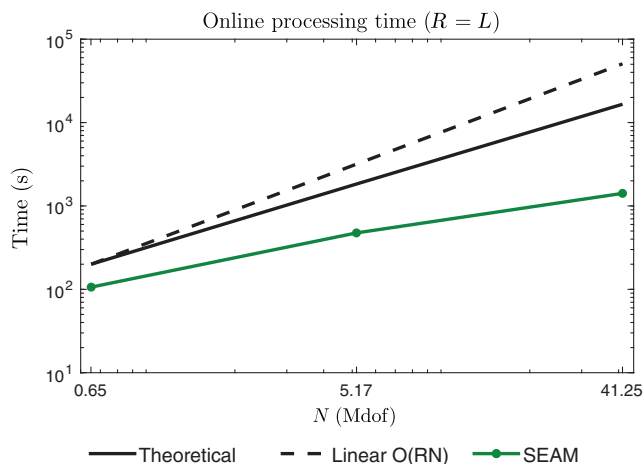


Figure 16. Observed runtime as function of  $N$  for the SEAM model for  $R = L$  RHS. For comparison, we show the theoretical scaling of the polarized traces algorithm (solid black), as well as a linear scaling (dashed black).

Even though the methods mentioned above have been successful in reducing pollution error, the resulting linear systems cannot, in general, be solved in quasilinear time or better because the matrices either have a high degree of interconnectivity or are extremely ill-conditioned. However, under some hypothesis, it is possible to use asymptotic information or a reformulation as an integral equation to bypass the pollution error.

In that regard, the method of polarized traces has been extended to take advantage of these special cases to solve equation 1 without pollution error and with quasilinear complexity for smooth media (Fang et al., 2017) and for media with compactly supported heterogeneities (Zepeda-Núñez and Zhao, 2016). For highly heterogeneous media, the accuracy of finite elements has not been extensively studied; however, the method of polarized traces has been coupled with efficient hybridizable discontinuous Galerkin discretizations for highly heterogeneous media (Taus et al., 2016).

In this paper, we assume that waves will propagate in very general and highly heterogeneous media; thus, we do not have a theoretical framework to assess the accuracy. Instead, we use numerical experiments to check the accuracy of the solution. We compare our solution with one generated by solving the PDE on a grid with twice the number of points per wavelength, and we obtain a higher order accurate solution via Richardson extrapolation. In this regime, we observed a relative error of less than 10% between our solution and the numerical baseline.

## Extensions of the method

## Other discretizations

Although the method of polarized traces is not restricted to the second-order finite differences used in this study, applying higher order finite-difference schemes makes the numerical implementation slightly more complicated and expensive. The increased size of higher order stencils requires thicker (in terms of grid points) coupling layers in the interface regions, which increases the memory footprint. The resulting computation is slightly more complicated but is definitely still feasible.

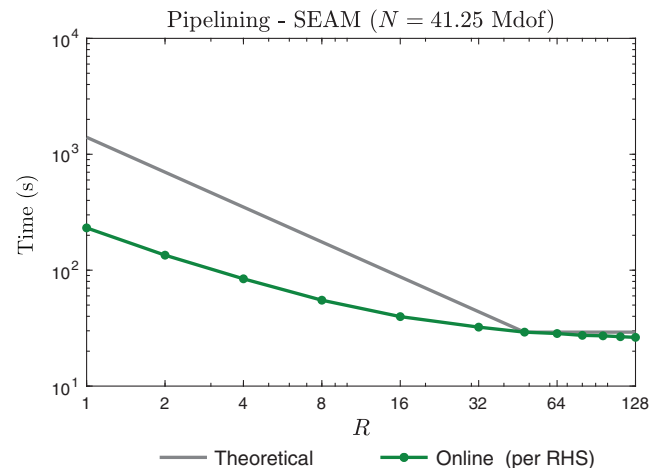


Figure 17. The impact of scalability on pipelining for the SEAM model.  $N$  is held constant and  $R$  is increased. The observed runtime as function of  $R$  is in green, and the dashed black line illustrates the theoretical scalability.

Also, although we present our method in the context of finite differences, the framework is also valid in the context of finite-element (Taus et al., 2016; Fang et al., 2017) and integral-equation (Zepeda-Núñez and Zhao, 2016) formulations. Additionally, the method can be easily extended to finite elements with unstructured meshes using, for example, the PML proposed in Bermúdez et al. (2007), with the caveat that the partition of the medium in slabs may become slightly more cumbersome to work with.

### Attenuation

In this study, we have not explicitly addressed the Helmholtz equation in the presence of interior attenuation. Attenuation can be interpreted as a complex shift in frequency, which provides additional dampening throughout the attenuated region, which generally improves the performance of iterative and hybrid solvers. For a sweeping method preconditioned with polarized traces, one can expect a reduction in the number of Krylov solver iterations required, due to the reduction in reflections caused by the attenuation. However, in heavily attenuated cases, for similar reasons, the problem may be easier to solve with other methods, such as multigrid methods. For an extensive review of attenuated Helmholtz equations, we refer to Gander et al. (2015).

### Nonacoustic physics

Our method relies only on the sparsity of the matrix and an efficient implementation of an absorbing boundary condition for an efficient implementation. The efficiency and scalability are not due our choice of isotropic acoustic physics for this study. The sparsity pattern in the discrete time-harmonic wave equation is a by-product of the discretization, and it can be manipulated to extend the method to other regimes, e.g., the time-harmonic elastic case. Such an extension can be obtained using, e.g., a finite-difference discretization of the time-harmonic elastic wave equation, and after reordering the degrees of freedom by clustering unknowns associated to grid points nearby in space, it is possible to obtain a block-banded system. The mathematical and computational machinery developed in this paper, and related results (e.g., Zepeda-Núñez and Demanet, 2018), are then applicable. For example, Zepeda-Núñez and Zhao (2016) extend the method of polarized traces to solve an unphysical discretization coming from a minimization problem, in which the main challenge was building the nonreflecting boundary conditions.

## CONCLUSION

We have presented a new and efficient solver for the 3D high-frequency Helmholtz equation in heterogeneous media. The solver achieves sublinear runtimes by leveraging the pipelining and effective communication patterns surrounding a well-parallelized black-box local solver. The method presented in this paper broadens the applicability of parallel direct methods by embedding them in a domain-decomposition method whose rate of convergence is independent of the frequency, subject to the constraint that there are no subwavelength structures oriented perpendicularly to the sweeping direction and that there are no large resonant structures in the medium. Although the results obtained in our study show excellent performance in relatively simple media and in the complicated SEAM model, we would expect the performance to degrade in these regimes

in which there will be more reflections. That said, in some cases, the performance degradation can be mitigated by decomposing the domain in different dimensions.

Our implementation of the proposed solver shows that it is possible to scale the problem size and the number of RHS, while preserving a compute time that scales sublinearly with respect to the total number of degrees of freedom. A next step would be to increase the number of RHS, processing them in batches. In our treatment, we restricted our usage to a single RHS per layer, which is overly conservative. By using BLAS3 operations supporting multiple RHS, one would expect only a slight increase of the overall runtime with no degradation to the asymptotic performance.

A scalable solver for the time-harmonic wave equation is critical to make feasible solving the subsurface inversion problem in the frequency domain. By exploiting pipelining, the approach we have presented makes optimal use of HPC systems and optimal reuse of numerical calculations. Specifically, by paying an upfront factorization cost that is always proportional to the cost of the current best-available direct or hybrid solvers on the subdomains, we can optimally saturate HPC clusters by processing multiple RHS simultaneously. By holding the number of subdomains proportional to the number of RHS, this approach easily scales to the number of RHS necessary to solve FWI problems. With parallelism over subdomains, within the local subdomain solves, and across multiple problems (RHS), the method presented here makes maximal use of parallelism.

## ACKNOWLEDGMENTS

This work was sponsored by Total SA. L. Demanet is also sponsored by AFOSR grant no. FA9550-17-1-0316, ONR grant no. N00014-16-1-2122, and NSF grant no. DMS-1255203. The authors are grateful to M. Fehler and Total SA for their permission to use the 3D SEAM model.

## APPENDIX A ALGORITHM

To apply the blocks in a matrix-free fashion, we use Algorithms A-2 and 6 in line 4 of Algorithm A-1, and we use Algorithms A-3 and A-4 in line 5 of Algorithm A-1. All algorithms in this section are embarrassingly parallel at the level of the layers as depicted in Figure 3.

### Algorithm A-1. Application of $\mathbf{M}$ .

---

```

1: function  $\mathbf{u}$  = APPLICATION POLARIZED ( $\mathbf{y}$ )
2:    $(\mathbf{v}^\downarrow, \mathbf{v}^\uparrow) = \mathbf{y}$ 
3:    $\mathbf{u}^\downarrow = \mathbf{D}^\downarrow \mathbf{v}^\downarrow + \mathbf{U} \mathbf{v}^\uparrow$ 
4:    $\mathbf{u}^\uparrow = \mathbf{D}^\uparrow \mathbf{v}^\uparrow + \mathbf{L} \mathbf{v}^\downarrow$ 
5:    $\mathbf{u} = (\mathbf{u}^\downarrow, \mathbf{u}^\uparrow)$ 
6: end function

```

---

**Algorithm A-2. Application of  $\underline{\mathbf{D}}^\downarrow$ .**


---

```

1: Function  $\underline{\mathbf{u}}^\downarrow = \text{DOWNWARD SWEEP } (\underline{\mathbf{v}}^\downarrow)$ 
2:  $\mathbf{u}_n^{\downarrow,1} = -\mathbf{v}_n^{\downarrow,1}$ 
3:  $\mathbf{u}_{n+1}^{\downarrow,1} = -\mathbf{v}_{n+1}^{\downarrow,1}$ 
4: for  $\ell = 2:L-1$ 
5:  $\tilde{\mathbf{f}}^\ell = \delta(z_1 - z)\mathbf{v}_{n^{\ell-1}}^{\downarrow,\ell-1} - \delta(z_0 - z)\mathbf{v}_{n^{\ell-1}+1}^{\downarrow,\ell-1}$ 
6:  $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1}\tilde{\mathbf{f}}^\ell$ 
7:  $\mathbf{u}_n^{\downarrow,\ell} = \mathbf{w}_n^\ell - \mathbf{v}_n^{\downarrow,\ell}$ 
8:  $\mathbf{u}_{n+1}^{\downarrow,\ell} = \mathbf{w}_{n+1}^\ell - \mathbf{v}_{n+1}^{\downarrow,\ell}$ 
9: end for
10:  $\underline{\mathbf{u}}^\downarrow = (\mathbf{u}_n^{\downarrow,1}, \mathbf{u}_{n+1}^{\downarrow,1}, \mathbf{u}_n^{\downarrow,2}, \dots, \mathbf{u}_{n^{L-1}}^{\downarrow,L-1}, \mathbf{u}_{n^{L-1}+1}^{\downarrow,L-1})^t$ 
11: end function

```

---

**Algorithm A-3. Upward sweep, application of  $(\underline{\mathbf{D}}^\uparrow)^{-1}$ .**


---

```

1: function  $\underline{\mathbf{u}}^\uparrow = \text{UPWARD SWEEP } (\underline{\mathbf{v}}^\uparrow)$ 
2:  $\mathbf{u}_0^{\uparrow,L} = -\mathbf{v}_0^{\uparrow,L}$ 
3:  $\mathbf{u}_1^{\uparrow,L} = -\mathbf{v}_1^{\uparrow,L}$ 
4: for  $\ell = L-1:2$ 
5:  $\tilde{\mathbf{f}}^\ell = -\delta(z_{n^{\ell+1}} - z)\mathbf{v}_0^{\uparrow,\ell+1} + \delta(z_{n^\ell} - z)\mathbf{v}_1^{\uparrow,\ell+1}$ 
6:  $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1}\tilde{\mathbf{f}}^\ell$ 
7:  $\mathbf{u}_1^{\uparrow,\ell} = \mathbf{w}_1^\ell - \mathbf{v}_1^{\uparrow,\ell}$ 
8:  $\mathbf{u}_0^{\uparrow,\ell} = \mathbf{w}_0^\ell - \mathbf{v}_0^{\uparrow,\ell}$ 
9: end for
10:  $\underline{\mathbf{u}}^\uparrow = (\mathbf{u}_0^{\uparrow,2}, \mathbf{u}_1^{\uparrow,2}, \mathbf{u}_0^{\uparrow,3}, \dots, \mathbf{u}_0^{\uparrow,L}, \mathbf{u}_1^{\uparrow,L})^t$ 
11: end function

```

---

**Algorithm A-4. Downward reflections, application of  $\underline{\mathbf{U}}$ .**


---

```

1: function  $\underline{\mathbf{u}}^\uparrow = \text{UPWARD REFLECTIONS } (\underline{\mathbf{v}}^\downarrow)$ 
2: for  $\ell = 2:L-1$ 
3:  $\mathbf{f}^\ell = \delta(z_1 - z)\mathbf{v}_0^{\downarrow,\ell} - \delta(z_0 - z)\mathbf{v}_1^{\downarrow,\ell}$ 
    $- \delta(z_{n^{\ell+1}} - z)\mathbf{v}_1^{\downarrow,\ell+1} + \delta(z_{n^\ell} - z)\mathbf{v}_0^{\downarrow,\ell+1}$ 
4:  $\mathbf{w}^\ell = (\mathbf{H}^\ell)^{-1}\mathbf{f}^\ell$ 
5:  $\mathbf{u}_1^{\uparrow,\ell} = \mathbf{w}_1^\ell - \mathbf{v}_1^{\downarrow,\ell}$ 
6:  $\mathbf{u}_0^{\uparrow,\ell} = \mathbf{w}_0^\ell$ 
7: end for
8:  $\mathbf{f}^L = \delta(z_1 - z)\mathbf{v}_0^{\downarrow,L} - \delta(z_0 - z)\mathbf{v}_1^{\downarrow,L}$ 
9:  $\mathbf{w}^L = (\mathbf{H}^L)^{-1}\mathbf{f}^L$ 
10:  $\mathbf{u}_1^{\uparrow,L} = \mathbf{w}_1^L - \mathbf{v}_1^{\downarrow,L}$ 
11:  $\mathbf{u}_0^{\uparrow,L} = \mathbf{w}_0^L$ 
12:  $\underline{\mathbf{u}}^\uparrow = (\mathbf{u}_0^{\uparrow,2}, \mathbf{u}_1^{\uparrow,2}, \mathbf{u}_0^{\uparrow,3}, \dots, \mathbf{u}_0^{\uparrow,L}, \mathbf{u}_1^{\uparrow,L})^t$ 
13: end function

```

---

**DATA AND MATERIALS AVAILABILITY**

Data associated with this research are available and can be obtained by contacting the corresponding author.

**REFERENCES**

- Alessandrini, G., M. V. de Hoop, F. Faucher, R. Gaburro, and E. Sincich, 2017, Inverse problem for the Helmholtz equation with cauchy data: Reconstruction with conditional well-posedness driven iterative regularization: ArXiv e-prints.
- Amestoy, P., C. Ashcraft, O. Boiteau, A. Buttari, J.-Y. L'Excellent, and C. Weisbecker, 2015, Improving multifrontal methods by means of block low-rank representations: SIAM Journal on Scientific Computing, **37**, A1451–A1474, doi: [10.1137/120903476](https://doi.org/10.1137/120903476).
- Amestoy, P., R. Brossier, A. Buttari, J.-Y. L'Excellent, T. Mary, L. Métivier, A. Miniussi, and S. Operto, 2016, Fast 3D frequency-domain full-waveform inversion with a parallel block low-rank multifrontal direct solver: Application to OBC data from the North Sea: Geophysics, **81**, no. 6, R363–R383, doi: [10.1190/geo2016-0052.1](https://doi.org/10.1190/geo2016-0052.1).
- Amestoy, P. R., I. S. Duff, J.-Y. L'Excellent, and J. Koster, 2001, A fully asynchronous multifrontal solver using distributed dynamic scheduling: SIAM Journal on Matrix Analysis and Applications, **23**, 15–41, doi: [10.1137/S0895479899358194](https://doi.org/10.1137/S0895479899358194).
- Aruliah, D., and U. Ascher, 2002, Multigrid preconditioning for Krylov methods for time-harmonic Maxwell's equations in three dimensions: SIAM Journal on Scientific Computing, **24**, 702–718, doi: [10.1137/S1064827501387358](https://doi.org/10.1137/S1064827501387358).
- Asano, T., D. Ranjan, T. Roos, E. Welzl, and P. Widmayer, 1997, Space-filling curves and their use in the design of geometric data structures: Theoretical Computer Science, **181**, 3–15, doi: [10.1016/S0304-3975\(96\)00259-9](https://doi.org/10.1016/S0304-3975(96)00259-9).
- Ashcraft, C., R. Grimes, J. Lewis, B. Peyton, H. Simon, and P. Bjørstad, 1987, Progress in sparse matrix methods for large linear systems on vector supercomputers: The International Journal of Supercomputing Applications, **1**, 10–30, doi: [10.1177/109434208700100403](https://doi.org/10.1177/109434208700100403).
- Astaneh, A. V., and M. N. Guddati, 2016, A two-level domain decomposition method with accurate interface conditions for the Helmholtz problem: International Journal for Numerical Methods in Engineering, **107**, 74–90, doi: [10.1002/nme.v107.1](https://doi.org/10.1002/nme.v107.1).
- Babuska, I., and U. Banerjee, 2012, Stable generalized finite element method (SGFEM): Computer Methods in Applied Mechanics and Engineering, **201-204**, 91–111, doi: [10.1016/j.cma.2011.09.012](https://doi.org/10.1016/j.cma.2011.09.012).
- Babuska, I., F. Ihlenburg, E. T. Paik, and S. A. Sauter, 1995, A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution: Computer Methods in Applied Mechanics and Engineering, **128**, 325–359, doi: [10.1016/0045-7825\(95\)00890-X](https://doi.org/10.1016/0045-7825(95)00890-X).
- Babuska, I., and J. M. Melenk, 1997, The partition of unity method: International Journal for Numerical Methods in Engineering, **40**, 727–758, doi: [10.1002/\(SICI\)1097-0207\(19970228\)40:4<727::AID-NME86>3.0.CO;2-N](https://doi.org/10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N).
- Bebendorf, M., 2008, Hierarchical matrices: A means to efficiently solve elliptic boundary value problems: Springer-Verlag, Lecture Notes in Computational Science and Engineering.
- Bérenger, J.-P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: Journal of Computational Physics, **114**, 185–200, doi: [10.1006/jcph.1994.1159](https://doi.org/10.1006/jcph.1994.1159).
- Berkhout, A. J., 1980, Seismic migration: Imaging of acoustic energy by wave field extrapolation: Elsevier.
- Bermúdez, A., L. Hervella-Nieto, A. Prieto, and R. Rodríguez, 2007, An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems: Journal of Computational Physics, **223**, 469–488, doi: [10.1016/j.jcp.2006.09.018](https://doi.org/10.1016/j.jcp.2006.09.018).
- Betcke, T., and J. Phillips, 2012, Approximation by dominant wave directions in plane wave methods: Technical report, UCL.
- Boerm, S., L. Grasedyck, and W. Hackbusch, 2006, Hierarchical matrices: Max-Planck-Institute, Lecture Notes.
- Boubendir, Y., 2007, An analysis of the BEM-FEM non-overlapping domain decomposition method for a scattering problem: Journal of Computational and Applied Mathematics, **204**, 282–291, doi: [10.1016/j.cam.2006.02.044](https://doi.org/10.1016/j.cam.2006.02.044).
- Boubendir, Y., X. Antoine, and C. Geuzaine, 2012, A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation: Journal of Computational Physics, **231**, 262–280, doi: [10.1016/j.jcp.2011.08.007](https://doi.org/10.1016/j.jcp.2011.08.007).
- Brandt, A., and I. Livshits, 1997, Wave-ray multigrid method for standing wave equations: Electronic Transactions on Numerical Analysis, **6**, 162–181.
- Calandra, H., S. Gratton, X. Pinel, and X. Vasseur, 2013, An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media: Numerical Linear Algebra with Applications, **20**, 663–688, doi: [10.1002/nla.1860](https://doi.org/10.1002/nla.1860).

- Cessenat, O., and B. Després, 1998, Application of an ultra weak variational formulation of elliptic PDES to the two-dimensional Helmholtz problem: *SIAM Journal on Numerical Analysis*, **35**, 255–299, doi: [10.1137/S0036142995285873](https://doi.org/10.1137/S0036142995285873).
- Chan, T. F., and T. P. Mathew, 1994, Domain decomposition algorithms: *Acta Numerica*, **3**, 61–143, doi: [10.1017/S0962492900002427](https://doi.org/10.1017/S0962492900002427).
- Chandrasekaran, S., M. Gu, and T. Pals, 2006, A fast ULV decomposition solver for hierarchically semiseparable representations: *SIAM Journal on Matrix Analysis and Applications*, **28**, 603–622, doi: [10.1137/S0895479803436652](https://doi.org/10.1137/S0895479803436652).
- Chen, Y., 1997, Inverse scattering via Heisenberg's uncertainty principle: *Inverse Problems*, **13**, 253–282, doi: [10.1088/0266-5611/13/2/005](https://doi.org/10.1088/0266-5611/13/2/005).
- Chen, Z., and X. Xiang, 2013a, A source transfer domain decomposition method for Helmholtz equations in unbounded domain: *SIAM Journal on Numerical Analysis*, **51**, 2331–2356, doi: [10.1137/130917144](https://doi.org/10.1137/130917144).
- Chen, Z., and X. Xiang, 2013b, A source transfer domain decomposition method for Helmholtz equations in unbounded domain. Part 2: Extensions: *Numerical Mathematics: Theory, Methods and Applications*, **6**, 538–555.
- Chew, W. C., and W. H. Weedon, 1994, A 3-D perfectly matched medium from modified Maxwell's equations with stretched coordinates: *Micro-wave and Optical Technology Letters*, **7**, 599–604, doi: [10.1002/mop.4650071304](https://doi.org/10.1002/mop.4650071304).
- Collino, F., S. Ghanemi, and P. Joly, 2000, Domain decomposition method for harmonic wave propagation: A general presentation: *Computer Methods in Applied Mechanics and Engineering*, **184**, 171–211, doi: [10.1016/S0045-7825\(99\)00228-5](https://doi.org/10.1016/S0045-7825(99)00228-5).
- Collino, F., and P. Joly, 1995, Splitting of operators, alternate directions, and paraxial approximations for the three-dimensional wave equation: *SIAM Journal on Scientific Computing*, **16**, 1019–1048, doi: [10.1137/0916059](https://doi.org/10.1137/0916059).
- Davis, T. A., 2004, Algorithm 832: UMFPACK v4.3: An unsymmetric-pattern multifrontal method: *ACM Transactions on Mathematical Software*, **30**, 196–199, doi: [10.1145/992200](https://doi.org/10.1145/992200).
- Davis, T. A., S. Rajamanickam, and W. M. Sid-Lakhdar, 2016, A survey of direct methods for sparse linear systems: *Acta Numerica*, **25**, 383–566, doi: [10.1017/S0962492916000076](https://doi.org/10.1017/S0962492916000076).
- de Hoop, M., J. H. Le Rousseau, and R.-S. Wu, 2000, Generalization of the phase-screen approximation for the scattering of acoustic waves: *Wave Motion*, **31**, 43–70, doi: [10.1016/S0165-2125\(99\)00026-8](https://doi.org/10.1016/S0165-2125(99)00026-8).
- de Hoop, M. V., S. Wang, and J. Xia, 2011, On 3D modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver: *Geophysical Prospecting*, **59**, 857–873, doi: [10.1111/j.1365-2478.2011.00982.x](https://doi.org/10.1111/j.1365-2478.2011.00982.x).
- de La Bourdonnaye, A., C. Farhat, A. Macedo, F. Magoules, and F.-X. Roux, 1998, A nonoverlapping domain decomposition method for the exterior Helmholtz problem: *Contemporary Mathematics*, **218**, 42–66, doi: [10.1090/conm/218](https://doi.org/10.1090/conm/218).
- Demmel, J. W., S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, 1999, A supermodal approach to sparse partial pivoting: *SIAM Journal Matrix Analysis and Applications*, **20**, 720–755, doi: [10.1137/S0895479895291765](https://doi.org/10.1137/S0895479895291765).
- Després, B., 1990, Décomposition de domaine et problème de Helmholtz: *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, **311**, 313–316.
- Duff, I. S., and J. Koster, 2001, On algorithms for permuting large entries to the diagonal of a sparse matrix: *SIAM Journal on Matrix Analysis and Applications*, **22**, 973–996, doi: [10.1137/S0895479899358443](https://doi.org/10.1137/S0895479899358443).
- Duff, I. S., and J. K. Reid, 1983, The multifrontal solution of indefinite sparse symmetric linear: *ACM Transactions on Mathematical Software*, **9**, 325–302, doi: [10.1145/356044.356047](https://doi.org/10.1145/356044.356047).
- Engquist, B., and L. Ying, 2011a, Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation: *Communications on Pure and Applied Mathematics*, **64**, 697–735, doi: [10.1002/cpa.v64.5](https://doi.org/10.1002/cpa.v64.5).
- Engquist, B., and L. Ying, 2011b, Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers: *Multiscale Modeling and Simulation*, **9**, 686–710, doi: [10.1137/100804644](https://doi.org/10.1137/100804644).
- Engquist, B., and H.-K. Zhao, 1998, Absorbing boundary conditions for domain decomposition: *Applied Numerical Mathematics*, **27**, 341–365, doi: [10.1016/S0168-9274\(98\)00019-1](https://doi.org/10.1016/S0168-9274(98)00019-1).
- Engquist, B., and H.-K. Zhao, 2018, Approximate separability of the Green's function of the Helmholtz equation in the high frequency limit: *Computations on Pure and Applied Mathematics*, **71**, 2220–2274, doi: [10.1002/cpa.v71.11](https://doi.org/10.1002/cpa.v71.11).
- Erlangga, Y. A., C. W. Oosterlee, and C. Vuik, 2006, A novel multigrid based preconditioner for heterogeneous Helmholtz problems: *SIAM Journal on Scientific Computing*, **27**, 1471–1492, doi: [10.1137/040615195](https://doi.org/10.1137/040615195).
- Fang, J., J. Qian, L. Zepeda-Núñez, and H. Zhao, 2017, Learning dominant wave directions for plane wave methods for high-frequency Helmholtz equations: *Research in the Mathematical Sciences*, **4**, 9, doi: [10.1186/s40687-017-0098-9](https://doi.org/10.1186/s40687-017-0098-9).
- Farhat, C., I. Harari, and L. P. Franca, 2001, The discontinuous enrichment method: *Computer Methods in Applied Mechanics and Engineering*, **190**, 6455–6479, doi: [10.1016/S0045-7825\(01\)00232-8](https://doi.org/10.1016/S0045-7825(01)00232-8).
- Fehler, M., and P. Keliher, 2011, Seam phase 1: Challenges of subsalt imaging in Tertiary basins, with emphasis on deepwater Gulf of Mexico: SEG.
- Gander, M., I. Graham, and E. Spence, 2015, Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: What is the largest shift for which wavenumber-independent convergence is guaranteed?: *Numerische Mathematik*, **131**, 1–48.
- Gander, M., and F. Nataf, 2000, AILU for Helmholtz problems: A new preconditioner based on an analytic factorization: *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, **331**, 261–266.
- Gander, M. J., 2006, Optimized Schwarz methods: *SIAM Journal on Numerical Analysis*, **44**, 699–731, doi: [10.1137/S0036142903425409](https://doi.org/10.1137/S0036142903425409).
- Gander, M. J., and F. Kwok, 2011, Optimal interface conditions for an arbitrary decomposition into subdomains, in Y. Huang, R. Kornhuber, O. Widlund, and J. Xu, eds., *Domain decomposition methods in science and engineering XIX*: Springer, Lecture Notes in Computational Science and Engineering, 101–108.
- Gander, M. J., F. Magoules, and F. Nataf, 2002, Optimized Schwarz methods without overlap for the Helmholtz equation: *SIAM Journal on Scientific Computing*, **24**, 38–60, doi: [10.1137/S1064827501387012](https://doi.org/10.1137/S1064827501387012).
- Gander, M. J., and F. Nataf, 2005, An incomplete LU preconditioner for problems in acoustics: *Journal of Computational Acoustics*, **13**, 455–476, doi: [10.1142/S0218396X05002803](https://doi.org/10.1142/S0218396X05002803).
- Gander, M. J., and Y. Xu, 2016, Optimized Schwarz method with two-sided transmission conditions in an unsymmetric domain decomposition, in T. Dickopf, J. M. Gander, L. Halpern, R. Krause, and F. L. Pavarino, eds., *Domain decomposition methods in science and engineering XXII*: Springer International Publishing, 631–639.
- Gander, M. J., and H. Zhang, 2013, Domain decomposition methods for the Helmholtz equation: A numerical investigation, in R. Bank, M. Holst, O. Widlund, and J. Xu, eds., *Domain decomposition methods in science and engineering XX*: Springer, Lecture Notes in Computational Science and Engineering, 215–222.
- Gander, M. J., and H. Zhang, 2014, Optimized Schwarz methods with overlap for the Helmholtz equation, in J. Erhel, J. M. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, eds., *Domain decomposition methods in science and engineering XXI*: Springer International Publishing, 207–215.
- George, A., 1973, Nested dissection of a regular finite element mesh: *SIAM Journal on Numerical Analysis*, **10**, 345–363, doi: [10.1137/0710032](https://doi.org/10.1137/0710032).
- Ghanemi, S., 1998, A domain decomposition method for Helmholtz scattering problems: *Ninth International Conference on Domain Decomposition Methods*, 105–112.
- Gillman, A., A. H. Barnett, and P.-G. Martinsson, 2015, A spectrally accurate direct solution technique for frequency-domain scattering problems with variable media: *BIT Numerical Mathematics*, **55**, 141–170, doi: [10.1007/s10543-014-0499-8](https://doi.org/10.1007/s10543-014-0499-8).
- Gittelsohn, C. J., R. Hiptmair, and I. Perugia, 2009, Plane wave discontinuous Galerkin methods: Analysis of the h-version: *ESAIM: Mathematical Modeling and Numerical Analysis*, **43**, 297–331, doi: [10.1051/m2an/2009002](https://doi.org/10.1051/m2an/2009002).
- Goldstein, C. I., 1986, The weak element method applied to Helmholtz type equations: *Applied Numerical Mathematics*, **2**, 409–426, doi: [10.1016/0168-9274\(86\)90043-7](https://doi.org/10.1016/0168-9274(86)90043-7).
- Gordon, D., and R. Gordon, 2010, Carp-cg: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated {PDEs}: *Parallel Computing*, **36**, 495–515, doi: [10.1016/j.parco.2010.05.004](https://doi.org/10.1016/j.parco.2010.05.004).
- Graham, I., M. Löhndorf, J. Melenk, and E. Spence, 2015, When is the error in the h-BEM for solving the Helmholtz equation bounded independently of k?: *BIT Numerical Mathematics*, **55**, 171–214, doi: [10.1007/s10543-014-0501-5](https://doi.org/10.1007/s10543-014-0501-5).
- Gupta, A., G. Karypis, and V. Kumar, 1997, Highly scalable parallel algorithms for sparse matrix factorization: *IEEE Transactions on Parallel and Distributed Systems*, **8**, 502–520, doi: [10.1109/71.598277](https://doi.org/10.1109/71.598277).
- Hiptmair, R., A. Moiola, and I. Perugia, 2016, A survey of Trefftz methods for the Helmholtz equation, in G.-R. Barenechea, F. Brezzi, A. Cangiani, and E.-H. Georgoulis, eds., *Building bridges: Connections and challenges in modern approaches to numerical partial differential equations*: Springer, 237–278.
- Hu, Q., and H. Zhang, 2016, Substructuring preconditioners for the systems arising from plane wave discretization of Helmholtz equations: *SIAM Journal on Scientific Computing*, **38**, A2232–A2261, doi: [10.1137/151003040](https://doi.org/10.1137/151003040).
- Ihlenburg, F., and I. Babuska, 1995, Finite element solution of the Helmholtz equation with high wave number. Part 1: The h-version of the FEM: *Computers and Mathematics with Applications*, **30**, 9–37, doi: [10.1016/0898-1221\(95\)00144-N](https://doi.org/10.1016/0898-1221(95)00144-N).
- Johnson, S., 2010, Notes on perfectly matched layers (PMLs).
- Joshi, M. V., A. Gupta, G. Karypis, and V. Kumar, 1997, A high performance two dimensional scalable parallel algorithm for solving sparse triangular systems: *Proceedings of the Fourth International Conference on High-Performance Computing*, 137–143.

- Karypis, G., and V. Kumar, 1998, A parallel algorithm for multilevel graph partitioning and sparse matrix ordering: *Journal of Parallel and Distributed Computing*, **48**, 71–95, doi: [10.1006/jpdc.1997.1403](https://doi.org/10.1006/jpdc.1997.1403).
- Komatitsch, D., and R. Martin, 2007, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation: *Geophysics*, **72**, no. 5, SM155–SM167, doi: [10.1190/1.2757586](https://doi.org/10.1190/1.2757586).
- Laird, A., and M. Giles, 2002, Preconditioned iterative solution of the 2D Helmholtz equation: Technical Report NA 02-12, Computing Lab, Oxford University.
- Li, X. S., and J. W. Demmel, 2003, SuperLU DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems: *ACM Transactions on Mathematical Software*, **29**, 110–140, doi: [10.1145/779359](https://doi.org/10.1145/779359).
- Li, Y., L. Métivier, R. Brossier, B. Han, and J. Virieux, 2015, 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver: *Geophysics*, **80**, no. 3, T101–T118, doi: [10.1190/geo2014-0480.1](https://doi.org/10.1190/geo2014-0480.1).
- Lions, P.-L., 1989, On the Schwarz alternating method II, in T. Chan, R. Glowinski, J. Periaux, and O. Widlund, eds., *Domain decomposition methods*: SIAM, Lecture Notes in Computational Science and Engineering, 47–70.
- Liu, F., and L. Ying, 2015, Recursive sweeping preconditioner for the 3D Helmholtz equation: ArXiv e-prints.
- Magoules, F., K. Meerbergen, and J.-P. Coyette, 2000, Application of a domain decomposition with Lagrange multipliers to acoustic problems arising from the automotive industry: *Journal of Computational Acoustics*, **08**, 503–521, doi: [10.1142/S0218396X00000297](https://doi.org/10.1142/S0218396X00000297).
- McInnes, L. C., R. F. Susan-Resiga, D. E. Keyes, and H. M. Atassi, 1998, Additive Schwarz methods with nonreflecting boundary conditions for the parallel computation of Helmholtz problems: *Contemporary Mathematics*, **218**, 325–333, doi: [10.1090/conm/218](https://doi.org/10.1090/conm/218).
- Melenk, J. M., A. Parsania, and S. Sauter, 2013, General DG-methods for highly indefinite Helmholtz problems: *Journal of Scientific Computing*, **57**, 536–581, doi: [10.1007/s10915-013-9726-8](https://doi.org/10.1007/s10915-013-9726-8).
- Melenk, J. M., and S. Sauter, 2011, Wavenumber explicit convergence analysis for Galerkin discretizations of the Helmholtz equation: *SIAM Journal on Numerical Analysis*, **49**, 1210–1243, doi: [10.1137/090776202](https://doi.org/10.1137/090776202).
- Moiola, A., R. Hiptmair, and I. Perugia, 2011, Plane wave approximation of homogeneous Helmholtz solutions: *Zeitschrift für angewandte Mathematik und Physik*, **62**, 809–837, doi: [10.1007/s00033-011-0147-y](https://doi.org/10.1007/s00033-011-0147-y).
- Moiola, A., and E. Spence, 2014, Is the Helmholtz equation really sign-indefinite?: *SIAM Review*, **56**, 274–312, doi: [10.1137/120901301](https://doi.org/10.1137/120901301).
- Monk, P., and D.-Q. Wang, 1999, A least-squares method for the Helmholtz equation: *Computer Methods in Applied Mechanics and Engineering*, **175**, 121–136, doi: [10.1016/S0045-7825\(98\)00326-0](https://doi.org/10.1016/S0045-7825(98)00326-0).
- Mulder, W., 2006, A multigrid solver for 3D electromagnetic diffusion: *Geophysical Prospecting*, **54**, 633–649, doi: [10.1111/j.1365-2478.2006.00558.x](https://doi.org/10.1111/j.1365-2478.2006.00558.x).
- Nguyen, N. C., J. Peraire, F. Reitich, and B. Cockburn, 2015, A phase-based hybridizable discontinuous Galerkin method for the numerical solution of the Helmholtz equation: *Journal of Computational Physics*, **290**, 318–335, doi: [10.1016/j.jcp.2015.02.002](https://doi.org/10.1016/j.jcp.2015.02.002).
- Perugia, I., P. Pietra, and A. Russo, 2016, A plane wave virtual element method for the Helmholtz problem: *ESAIM: Mathematical Modeling and Numerical Analysis*, **50**, 783–808, doi: [10.1051/m2an/2015066](https://doi.org/10.1051/m2an/2015066).
- Plessix, R.-E., 2007, A Helmholtz iterative solver for 3D seismic-imaging problems: *Geophysics*, **72**, no. 5, SM185–SM194, doi: [10.1190/1.2738849](https://doi.org/10.1190/1.2738849).
- Plessix, R.-E., and W. A. Mulder, 2003, Separation-of-variables as a preconditioner for an iterative Helmholtz solver: *Applied Numerical Mathematics*, **44**, 385–400, doi: [10.1016/S0168-9274\(02\)00165-4](https://doi.org/10.1016/S0168-9274(02)00165-4).
- Pothen, A., and C. Sun, 1993, A mapping algorithm for parallel sparse Cholesky factorization: *SIAM Journal on Scientific Computing*, **14**, 1253–1257, doi: [10.1137/0914074](https://doi.org/10.1137/0914074).
- Poulson, J., B. Engquist, S. Li, and L. Ying, 2013, A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations: *SIAM Journal on Scientific Computing*, **35**, C194–C212, doi: [10.1137/120871985](https://doi.org/10.1137/120871985).
- Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain. Part I: Theory and verification in a physical scale model: *Geophysics*, **64**, 888–901, doi: [10.1190/1.1444597](https://doi.org/10.1190/1.1444597).
- Raghavan, P., 1998, Efficient parallel sparse triangular solution using selective inversion: *Parallel Processing Letters*, **8**, 29–40, doi: [10.1142/S0129626498000067](https://doi.org/10.1142/S0129626498000067).
- Ristow, D., and T. Ruehl, 1997, 3-D implicit finite-difference migration by multiway splitting: *Geophysics*, **62**, 554–567, doi: [10.1190/1.1444165](https://doi.org/10.1190/1.1444165).
- Rouet, F.-H., X. S. Li, P. Ghysels, and A. Napov, 2016, A distributed-memory package for dense hierarchically semi-separable matrix computations using randomization: *ACM Transactions on Mathematical Software*, **42**, 1–35, doi: [10.1145/2956571](https://doi.org/10.1145/2956571).
- Saad, Y., 2003, *Iterative methods for sparse linear systems*, 2nd ed.: SIAM.
- Saad, Y., and M. H. Schultz, 1986, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems: *SIAM Journal on Scientific and Statistical Computing*, **7**, 856–869, doi: [10.1137/0907058](https://doi.org/10.1137/0907058).
- Schwarz, H. A., 1870, Über einen grenzübergang durch alternierendes verfahren: *Viertel-jahresschrift der Naturforschenden Gesellschaft in Zurich*, **15**, 272–286.
- Sheikh, A. H., D. Lahaye, and C. Vuik, 2013, On the convergence of shifted Laplace preconditioner combined with multilevel deflation: *Numerical Linear Algebra with Applications*, **20**, 645–662, doi: [10.1002/nla.1882](https://doi.org/10.1002/nla.1882).
- Soubrier, F., A. Haiddar, L. Giraud, H. Ben-Hadj-Ali, S. Operto, and J. Virieux, 2011, Three-dimensional parallel frequency-domain visco-acoustic wave modeling based on a hybrid direct/iterative solver: *Geophysical Prospecting*, **59**, 834–856, doi: [10.1111/j.1365-2478.2011.00966.x](https://doi.org/10.1111/j.1365-2478.2011.00966.x).
- Stolk, C., 2013, A rapidly converging domain decomposition method for the Helmholtz equation: *Journal of Computational Physics*, **241**, 240–252, doi: [10.1016/j.jcp.2013.01.039](https://doi.org/10.1016/j.jcp.2013.01.039).
- Stolk, C. C., 2016, A dispersion minimizing scheme for the 3-D Helmholtz equation based on ray theory: *Journal of Computational Physics*, **314**, 618–646, doi: [10.1016/j.jcp.2016.03.023](https://doi.org/10.1016/j.jcp.2016.03.023).
- Stolk, C. C., 2017, An improved sweeping domain decomposition preconditioner for the Helmholtz equation: *Advances in Computational Mathematics*, **43**, 45–76, doi: [10.1007/s10444-016-9475-y](https://doi.org/10.1007/s10444-016-9475-y).
- Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: *Geophysics*, **49**, 1259–1266, doi: [10.1190/1.1441754](https://doi.org/10.1190/1.1441754).
- Taus, M., L. Demanet, and L. Zepeda-Núñez, 2016, A short note on a fast and high-order hybridizable discontinuous Galerkin solver for the 2D high-frequency Helmholtz equation: 86th Annual International Meeting, SEG, Expanded Abstracts, 3835–3840, doi: [10.1190/segam2016-13848017.1](https://doi.org/10.1190/segam2016-13848017.1).
- Toselli, A., and O. B. Widlund, 2005, *Domain decomposition methods — Algorithms and theory*: Springer.
- Tsuji, P., B. Engquist, and L. Ying, 2012, A sweeping preconditioner for time-harmonic Maxwell’s equations with finite elements: *Journal of Computational Physics*, **231**, 3770–3783, doi: [10.1016/j.jcp.2012.01.025](https://doi.org/10.1016/j.jcp.2012.01.025).
- Tsuji, P., J. Poulson, B. Engquist, and L. Ying, 2014, Sweeping preconditioners for elastic wave propagation with spectral element methods: *ESAIM: Mathematical Modeling and Numerical Analysis*, **48**, 433–447, doi: [10.1051/m2an/2013114](https://doi.org/10.1051/m2an/2013114).
- Tsuji, P., and L. Ying, 2012, A sweeping preconditioner for Yee’s finite difference approximation of time-harmonic Maxwell’s equations: *Frontiers of Mathematics in China*, **7**, 347–363, doi: [10.1007/s11464-012-0191-8](https://doi.org/10.1007/s11464-012-0191-8).
- van der Vorst, H. A., 1992, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems: *SIAM Journal on Scientific and Statistical Computing*, **13**, 631–644, doi: [10.1137/0913035](https://doi.org/10.1137/0913035).
- Vion, A., and C. Geuzaine, 2014, Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem: *Journal of Computational Physics*, **266**, 171–190, doi: [10.1016/j.jcp.2014.02.015](https://doi.org/10.1016/j.jcp.2014.02.015).
- Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: *Geophysics*, **74**, no. 6, WCC1–WCC26, doi: [10.1190/1.3238367](https://doi.org/10.1190/1.3238367).
- Wang, S., M. V. de Hoop, J. Xia, and X. S. Li, 2012, Massively parallel structured multifrontal solver for time-harmonic elastic waves in 3-D anisotropic media: *Geophysical Journal International*, **191**, 346–366, doi: [10.1111/j.1365-246X.2012.05634.x](https://doi.org/10.1111/j.1365-246X.2012.05634.x).
- Wang, S., X. S. Li, J. Xia, Y. Situ, and M. V. de Hoop, 2013, Efficient scalable algorithms for solving dense linear systems with hierarchically semiseparable structures: *SIAM Journal on Scientific Computing*, **35**, C519–C544, doi: [10.1137/110848062](https://doi.org/10.1137/110848062).
- Wu, R.-S., 1994, Wide-angle elastic wave one-way propagation in heterogeneous media and an elastic wave complex-screen method: *Journal of Geophysical Research: Solid Earth*, **99**, 751–766, doi: [10.1029/93JB02518](https://doi.org/10.1029/93JB02518).
- Xia, J., 2013, Randomized sparse direct solvers: *SIAM Journal on Matrix Analysis and Applications*, **34**, 197–227, doi: [10.1137/12087116X](https://doi.org/10.1137/12087116X).
- Xia, J., S. Chandrasekaran, M. Gu, and X. S. Li, 2010, Superfast multifrontal method for large structured linear systems of equations: *SIAM Journal on Matrix Analysis and Applications*, **31**, 1382–1411, doi: [10.1137/09074543X](https://doi.org/10.1137/09074543X).
- Zepeda-Núñez, L., and L. Demanet, 2016, The method of polarized traces for the 2D Helmholtz equation: *Journal of Computational Physics*, **308**, 347–388, doi: [10.1016/j.jcp.2015.11.040](https://doi.org/10.1016/j.jcp.2015.11.040).
- Zepeda-Núñez, L., and L. Demanet, 2018, Nested domain decomposition with polarized traces for the 2D Helmholtz equation: *SIAM Journal on Scientific Computing*, **40**, B942–B981, doi: [10.1137/15M104582X](https://doi.org/10.1137/15M104582X).
- Zepeda-Núñez, L., and H. Zhao, 2016, Fast alternating bidirectional preconditioner for the 2D high-frequency Lippmann-Schwinger equation: *SIAM Journal on Scientific Computing*, **38**, B866–B888, doi: [10.1137/16M1064660](https://doi.org/10.1137/16M1064660).
- Zhang, Y., 2006, The theory of true amplitude one-way wave equation migrations: *Chinese Journal of Geophysics*, **49**, 1267–1289, doi: [10.1002/cjg2.v49.5](https://doi.org/10.1002/cjg2.v49.5).