A short note on a pipelined polarized-trace algorithm for 3D Helmholtz

Adrien Scheuer[1], Leonardo Zepeda-Núñez[1,3(*)], Russell J. Hewett[2] and Laurent Demanet[1]
[1]Dept. of Mathematics and Earth Resources Lab, Massachusetts Institute of Technology; [2]Total E&P Research & Technology USA; [3]Dept. of Mathematics, University of California, Irvine.

SUMMARY

We present a fast solver for the 3D high-frequency Helmholtz equation in heterogeneous, constant density, acoustic media. The solver is based on the method of polarized traces, coupled with distributed linear algebra libraries and pipelining to obtain a solver with online runtime $\mathcal{O}(\max(1,R/n)N\log N)$ where $N=n^3$ is the total number of degrees of freedom and R is the number of right-hand sides.

INTRODUCTION

Time-harmonic wave scattering in heterogeneous acoustic or elastic media is a hard problem in numerical analysis, and is ubiquitous within the algorithmic pipelines of inversion techniques as shown by Chen (1997); Pratt (1999); Virieux and Operto (2009).

Given its importance, there has been a renewed interest in developing efficient algorithms to solve the resulting ill-conditioned linear system. Recent progress has been made on mostly two fronts: fast direct methods and efficient preconditioners.

- Fast direct solvers, as the ones introduced by de Hoop et al. (2011); Gillman et al. (2014); Amestoy et al. (2015), couple multifrontal techniques (see George (1973); Duff and Reid (1983)) with compressed linear algebra (see Bebendorf (2008)), resulting in efficient direct solvers with small memory footprint, albeit with the same suboptimal asymptotic complexity as standard multifrontal methods (such as Amestoy et al. (2001); Davis (2004)) in the high-frequency regime.
- On the other had, recently introduced preconditioners can be subdivided into multigrid-based preconditioners, such as the ones proposed by Erlangga et al. (2006); Sheikh et al. (2013); Calandra et al. (2013), which are simple to implement but have super-linear complexity and need special tuning; and sweeping-like preconditioners such as the ones proposed by Gander and Nataf (2005); Engquist and Ying (2011a,b); Liu and Ying (2015); Chen and Xiang (2013); Stolk (2013); Zepeda-Núñez and Demanet (2016); Vion and Geuzaine (2014).

It has become clear that sweeping preconditioners and their generalizations, i.e., domain decomposition techniques coupled to high-quality transmission/absorption conditions, offer the right mix of ideas to attain linear or near-linear complexity in 2D and 3D., provided that the medium does not have large resonant cavities.

For practical applications, runtimes are often more important than asymptotic complexity. This requirement has led to a recent effort to reduce the runtimes of preconditioners with optimal asymptotic complexity via parallelization. We can for instance cite Poulson et al. (2013) where a new local solver carefully handles communication patterns to obtain impressive timings;Stolk (2015) in which the data dependencies during the sweeps are modified to improve the parallelism; and Zepeda-Núñez and Demanet (2016), which we explain in the sequel.

So far, the question had been to minimize the parallel runtime or complexity of a single solve with a single right-hand side. In the scope of seismic inversion however, a better question may be to minimize the overall runtime or complexity of several solves involving many (thousands) of right-hand sides (rhs). The important parameters are now N, the total number of degrees of freedom which we assume equals n^3; and R, the number of right-hand sides. Linear complexity would mean $\mathcal{O}(RN)$ operations.

In this note we present a solver for the 3D high-frequency Helmholtz equation with a *sublinear* runtime, given by

$$\mathcal{O}(\max(1,R/n)N\log N).$$

This scaling is possible by parallelization and pipelining of the right-hand sides. In other words, there is no essential complexity penalty as long as R stays on the order of n, the number of grid points per dimension. At its heart, the solver is based on the idea of polarized traces.

The method of polarized traces (Zepeda-Núñez and Demanet (2016)) is a layered domain decomposition method that uses 1) efficient direct solvers inside each subdomain, 2) high-quality transmission conditions between subdomains implemented via PML (c.f., Bérenger (1994)), and 3) an efficient preconditioner based on polarizing conditions imposed via incomplete Green's integrals. The result is an iterative method that converges in a typically very small number of iterations. The method has two stages: an offline stage, that is computed once for each system to invert; and an online stage, that is computed for each right-hand side or, in this case, for a batch of right-hand sides.

We point out that pipelining for this kind of problems is not new, as it is considered in Stolk (2015); however, the authors are unaware of any claims with respect to asymptotic *overall runtimes* within the context of inversion algorithms, in particular, with respect to full waveform inversion (c.f., Tarantola (1984)).

Assumptions for the complexity claim

We suppose that L, the number of layers in the domain decomposition, scales as $L\sim n$, i.e., each layer has a constant thickness in number of grid-points, denoted by q. Moreover, we assume that the number of computing nodes is $\mathcal{O}(N\log(N))$, with $\mathcal{O}(n^2\log(N))$ nodes for each layer, which is mainly due

to memory requirements. We make the implicit, yet realistic, assumption that the nodes have finite memory, thus to solve bigger problems, more nodes are needed. In theory, having more nodes would imply lower runtimes; however, as it will be explained in the sequel, the lack of asymptotic scalability of the linear solvers used in this work provides lower constants but, in general, no lower asymptotic runtimes.

METHOD

Consider a partition of a 3D rectangular domain Ω. Let $\mathbf{x} = (x,y,z)$ and consider the squared slowness $m(\mathbf{x}) = 1/c(\mathbf{x})^2$. The (constant-density, acoustic) Helmholtz equation is given by

$$\triangle u(\mathbf{x}) + \omega^2 m(\mathbf{x}) u(\mathbf{x}) = f_s(\mathbf{x}), \tag{1}$$

plus absorbing boundary conditions realized via PML as defined by Bérenger (1994); Johnson (2010); u is the wavefield and f_s are the right-hand sides, indexed by $s = 1,...,R$.

Eq. 1 is discretized using standard second and fourth order finite differences using a regular mesh, with a grid of size $n_x \times n_y \times n_z$ and a grid spacing h. The resulting discretized system is given by

$$\mathbf{H}\mathbf{u} = \mathbf{f}_s. \tag{2}$$

The method of polarized traces utilizes a layered domain decomposition $\{\Omega^\ell\}_{\ell=1}^L$, consisting of L layers of size $n_x \times n_y \times q$ plus the points used for the PML. We denote m^ℓ the restriction of the model parameters to Ω^ℓ, given by $m^\ell = m\chi_{\Omega^\ell}$, where χ_{Ω^ℓ} is the characteristic function of Ω^ℓ. We define the local Helmholtz problems by

$$\triangle v^\ell(\mathbf{x}) + m^\ell \omega^2 v^\ell(\mathbf{x}) = f_s^\ell(\mathbf{x}) = f_s \chi_{\Omega^\ell}(\mathbf{x}) \tag{3}$$

plus absorbing boundary conditions at the interfaces between subdomains. The local problem is discretized as $\mathbf{H}^\ell \mathbf{v}^\ell = \mathbf{f}_s^\ell$.

Given that the mesh is structured, we can define $\mathbf{x}_{i,j,k} = (x_i, y_j, z_k) = (ph, qh, rh)$. We assume the same ordering as in Zepeda-Núñez and Demanet (2016), i.e.

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{n_z}), \tag{4}$$

and we denote (using MATLAB notation)

$$\mathbf{u}_j = (u_{:,:,j}), \tag{5}$$

for the entries of $\mathbf{u}$ sampled at constant depth z_j. We write $\mathbf{u}^\ell$ for the wavefield defined locally at the ℓ-th layer, i.e., $\mathbf{u}^\ell = \chi_{\Omega^\ell}\mathbf{u}$, and $\mathbf{u}_k^\ell$ for the values at the local depth* z_k of $\mathbf{u}^\ell$. In particular, $\mathbf{u}_1^\ell$ and $\mathbf{u}_{n^\ell}^\ell$ are the top and bottom planes† of $\mathbf{u}^\ell$. We then gather the interface traces in the vector

$$\underline{\mathbf{u}} = \left(\mathbf{u}_{n^1}^1, \mathbf{u}_1^2, \mathbf{u}_{n^2}^2, ..., \mathbf{u}_1^{L-1}, \mathbf{u}_{n^{L-1}}^{L-1}, \mathbf{u}_1^L\right)^t. \tag{6}$$

Furthermore, we define the Dirac delta at a fixed depth. This operator takes a vector $\mathbf{v}_q$ at fixed depth, and gives back a vector defined in the volume that is given by

$$(\delta(z-z_p)\mathbf{v}_q)_{i,j,k} = \begin{cases} 0, & \text{if } k \neq p, \\ \frac{(\mathbf{v}_q)_{i,j}}{h^3}, & \text{if } k = p. \end{cases} \tag{7}$$

*We hope that there is little risk of confusion in overloading z_j (local indexing) for $z_{n_c^\ell + j}$ (global indexing), where $n_c^\ell = \sum_{j=1}^{\ell-1} n^j$ is the cumulative number of points in depth.

†We do not considered the PML points here.

Reduction to a surface integral equation

To obtain the global solution from the local systems, we couple the subdomains using the Green's representation formula within each layer, which results in a surface integral equation (SIE) posed at the interfaces between layers, of the form

$$\underline{\mathbf{M}}\,\underline{\mathbf{u}} = \underline{\mathbf{f}}_s, \tag{8}$$

where $\underline{\mathbf{M}}$ is formed by interface-to-interface Green's functions, $\underline{\mathbf{u}}$ is defined in Eq. 6 (in particular it is a vector of size $\mathcal{O}(Ln^2)$), and $\underline{\mathbf{f}}$ is specified in lines 2-6 of Alg. 1.

The matrix $\underline{\mathbf{M}}$ is a block banded matrix (see Fig. 1 (left) for its sparsity pattern) of size $2(L-1)n^2 \times 2(L-1)n^2$. It can be shown (Thm. 1 of Zepeda-Núñez and Demanet (2016)) that the solution of the system in Eq. 8 is exactly the restriction of the solution of Eq. 2 to the interfaces between layers.

Following Zepeda-Núñez and Demanet (2016), if the traces of the exact discrete solution are known, then it is possible to apply the Green's representation formula to reconstruct the global solution within each layer in an exact manner. The reconstruction can be performed in an equivalent manner by modifying the local right-hand sides with a measure supported on the interfaces between layers and then use a local solve. The procedure for the reconstruction is depicted in lines 11-12 of Alg. 1.

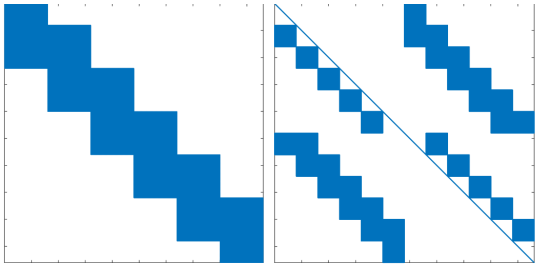

Figure 1: Sparsity pattern of the SIE matrix in Eq. 8 (left), and the polarized SIE matrix in Eq. 12 (right) .

A high-level description of the algorithm to solve the 3D high-frequency Helmholtz equation is given in Alg. 1.

Algorithm 1. *Online computation using the SIE reduction*

1: ***function*** $\mathbf{u}$ = HELMHOLTZ SOLVER($\mathbf{f}$)
2: ***for*** $\ell = 1 : L$ ***do***
3: $\mathbf{f}^\ell = \mathbf{f}\chi_{\Omega^\ell}$ ▷ *partition the source*
4: $\mathbf{v}^\ell = (\mathbf{H}^\ell)^{-1}\mathbf{f}^\ell$ ▷ *solve local problems*
5: ***end for***
6: $\underline{\mathbf{f}} = \left(\mathbf{v}_{n^1}^1, \mathbf{v}_1^2, \mathbf{v}_{n^2}^2, \ldots, \mathbf{v}_1^L\right)^t$ ▷ *form r.h.s.*
7: $\underline{\mathbf{u}} = (\underline{\mathbf{M}})^{-1}\underline{\mathbf{f}}$ ▷ *solve Eq. 8*
8: ***for*** $\ell = 1 : L$ ***do***
9: $\mathbf{g}^\ell = \mathbf{f}^\ell + \delta(z_1 - z)\mathbf{u}_{n^{\ell-1}}^{\ell-1} - \delta(z_0 - z)\mathbf{u}_1^\ell - \delta(z_{n^\ell+1} - z)\mathbf{u}_{n^\ell}^\ell + \delta(z_{n^\ell} - z)\mathbf{u}_1^{\ell+1}$
10: $\mathbf{u}^\ell = (\mathbf{H}^\ell)^{-1}\mathbf{g}^\ell$ ▷ *inner solve*
11: ***end for***
12: $\mathbf{u} = (\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^{L-1}, \mathbf{u}^L)^t$ ▷ *concatenate*
13: ***end function***

We can observe that the only non-embarrassingly parallel stage of Alg. 1 is the solution of Eq. 8. Given that $\underline{\underline{\mathbf{M}}}$ is never explicitly formed, using an iterative method to solve Eq. 8 would be the default choice. However, $\underline{\underline{\mathbf{M}}}$ is ill-conditioned, which forces us to define another equivalent integral system that is easy to precondition.

In this case each interface unknown is decomposed (polarized) into up- and down-going components such that

$$\underline{\mathbf{u}} = \underline{\mathbf{u}}^{\uparrow} + \underline{\mathbf{u}}^{\downarrow}. \tag{9}$$

This new variable is called the polarized wavefield and is denoted

$$\underline{\underline{\mathbf{u}}} = \begin{pmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{pmatrix}. \tag{10}$$

The introduction of the polarized wavefield doubles the number of unknowns, producing an overdetermined system. We close the system using the annihilation conditions (see Section 3 of Zepeda-Núñez and Demanet (2016)) that can be encoded in matrix form as

$$\underline{\mathbf{A}}^{\uparrow}\underline{\mathbf{u}}^{\uparrow} = 0, \quad \text{and} \quad \underline{\mathbf{A}}^{\downarrow}\underline{\mathbf{u}}^{\downarrow} = 0. \tag{11}$$

Imposing Eq. 8 and the annihilation condition we obtain yet another equivalent formulation, that results in the system

$$\underline{\underline{\mathbf{M}}}\,\underline{\underline{\mathbf{u}}} = \underline{\underline{\mathbf{f}}}_s, \tag{12}$$

where

$$\underline{\underline{\mathbf{M}}} = \begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \\ \underline{\mathbf{A}}^{\downarrow} & \underline{\mathbf{A}}^{\uparrow} \end{bmatrix}, \text{ and } \underline{\underline{\mathbf{f}}}_s = \begin{pmatrix} \underline{\mathbf{f}}_s \\ 0 \end{pmatrix}. \tag{13}$$

A series of basic algebraic operations rearranges the rows of $\underline{\underline{\mathbf{M}}}$ as

$$\underline{\underline{\mathbf{M}}} = \begin{bmatrix} \underline{\mathbf{D}}^{\downarrow} & \underline{\mathbf{U}} \\ \underline{\mathbf{L}} & \underline{\mathbf{D}}^{\uparrow} \end{bmatrix}, \tag{14}$$

whose sparsity pattern in shown in Fig. 1 (right) and where $\underline{\mathbf{D}}^{\downarrow}$ and $\underline{\mathbf{D}}^{\uparrow}$ have an identity on the diagonal, thus easily invertible using a block backsubstitution. Given the structure of the matrix $\underline{\underline{\mathbf{M}}}$, we can solve Eq. 12 in few iterations using GMRES (see Saad and Schultz (1986)) preconditioned by a block Gauss-Seidel iteration, which is given by

$$P_{\text{GS}} \begin{pmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{pmatrix} = \begin{pmatrix} \left(\underline{\mathbf{D}}^{\downarrow}\right)^{-1} \underline{\mathbf{u}}^{\downarrow} \\ \left(\underline{\mathbf{D}}^{\uparrow}\right)^{-1} \left(\underline{\mathbf{u}}^{\uparrow} - \underline{\mathbf{L}} \left(\underline{\mathbf{D}}^{\downarrow}\right)^{-1} \underline{\mathbf{u}}^{\downarrow}\right) \end{pmatrix}. \tag{15}$$

We apply the preconditioner, which relies on the application of $(\underline{\mathbf{D}}^{\downarrow})^{-1}$ and $(\underline{\mathbf{D}}^{\uparrow})^{-1}$ as presented in Alg. 2 and 3 respectively, by performing a sequence of solves in a sequential fashion.

Algorithm 2. *Downward sweep, application of* $(\underline{\mathbf{D}}^{\downarrow})^{-1}$

function $\underline{\mathbf{u}}^{\downarrow}$ = DOWNWARD SWEEP($\underline{\mathbf{v}}^{\downarrow}$)
  $\mathbf{u}_{n^1}^{\downarrow,1} = -\mathbf{v}_{n^1}^{\downarrow,1}$
  $\mathbf{u}_{n^1+1}^{\downarrow,1} = -\mathbf{v}_{n^1+1}^{\downarrow,1}$
  for $\ell = 2 : L-1$ ***do***
    $\widetilde{\mathbf{f}}^{\ell} = -\delta(z_0 - z)\mathbf{u}_{n^{\ell-1}+1}^{\downarrow,\ell-1} + \delta(z_1 - z)\mathbf{u}_{n^{\ell-1}}^{\downarrow,\ell-1}$
    $\mathbf{w}^{\ell} = (\mathbf{H}^{\ell})^{-1}\widetilde{\mathbf{f}}^{\ell}$
    $\mathbf{u}_{n^{\ell}}^{\downarrow,\ell} = \mathbf{w}_{n^{\ell}} - \mathbf{v}_{n^{\ell}}^{\downarrow,\ell}$; $\mathbf{u}_{n^{\ell}+1}^{\downarrow,\ell} = \mathbf{w}_{n^{\ell}+1} - \mathbf{v}_{n^{\ell}+1}^{\downarrow,\ell}$
  end for
  $\underline{\mathbf{u}}^{\downarrow} = \left(\mathbf{u}_{n^1}^{\downarrow,1}, \mathbf{u}_{n^1+1}^{\downarrow,1}, \mathbf{u}_{n^2}^{\downarrow,2}, ..., \mathbf{u}_{n^{L-1}}^{\downarrow,L-1}, \mathbf{u}_{n^{L-1}+1}^{\downarrow,L-1}\right)^t$
end function

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

function $\underline{\mathbf{u}}^{\uparrow}$ = UPWARD SWEEP($\underline{\mathbf{v}}^{\uparrow}$)
  $\mathbf{u}_0^{\uparrow,L} = -\mathbf{v}_0^{\uparrow,L}$
  $\mathbf{u}_1^{\uparrow,L} = -\mathbf{v}_1^{\uparrow,L}$
  for $\ell = L-1 : 2$ ***do***
    $\widetilde{\mathbf{f}}^{\ell} = -\delta(z_{n^{\ell}+1} - z)\mathbf{u}_0^{\uparrow,\ell+1} + \delta(z_{n^{\ell}} - z)\mathbf{u}_1^{\uparrow,\ell+1}$
    $\mathbf{w}^{\ell} = (\mathbf{H}^{\ell})^{-1}\widetilde{\mathbf{f}}^{\ell}$
    $\mathbf{u}_1^{\uparrow,\ell} = \mathbf{w}_1^{\ell} - \mathbf{v}_1^{\uparrow,\ell}$; $\mathbf{u}_0^{\uparrow,\ell} = \mathbf{w}_0^{\ell} - \mathbf{v}_0^{\uparrow,\ell}$
  end for
  $\underline{\mathbf{u}}^{\uparrow} = \left(\mathbf{u}_0^{\uparrow,2}, \mathbf{u}_1^{\uparrow,2}, \mathbf{u}_0^{\uparrow,3}, ..., \mathbf{u}_0^{\uparrow,L}, \mathbf{u}_1^{\uparrow,L}\right)^t$
end function

PIPELINING AND PARALLELIZATION

The method of polarized traces, in its matrix-free version, can be implemented using highly tailored distributed linear algebra with selective inversion, in a way that is reminiscent of the parallel sweeping preconditioner in Poulson et al. (2013), although with a much lower number of iterations for convergence. However, that would require a detailed communication pattern and complex code that is out of the scope of this paper.

Instead, we used the modularity of the method to link against different off-the-shelf sparse linear algebra direct solvers. In particular, we linked against UMFPACK (Davis (2004)), and STRUMPACK (Rouet et al. (2015)) for shared memory solvers and SuperLUDIST (Xia et al. (2010)) and MUMPS (Amestoy et al. (2001)) for the distributed solvers.

The method was implemented with two levels of MPI communicators. One level is reserved for the layers, and within each layers we used another MPI communicator to call the distributed linear algebra libraries. In the case of a shared memory solver we use only one MPI process per layer, thus having a completely transparent MPI implementation.

The method of polarized traces has a sequential bottleneck when applying the inverse $\underline{\mathbf{D}}^{\uparrow}$ and $\underline{\mathbf{D}}^{\downarrow}$ by backsubstitution. The application of the preconditioner using Alg. 2 and 3 would require to have only one layer working at a given time, while the others remain idle as depicted in Fig. 2. To remedy this problem, we pipeline the application of the preconditioner to multiple right-hand sides.

The main objective of pipelining is to balance the load of the processors, reducing the idle time thus increasing the computational efficiency. If we suppose that each local solve is performed in time $\gamma(n)$, then following Fig. 2 each GMRES iterations can be performed in $5\gamma(n) + 2L\gamma(n)$ plus communication costs.

In the implementation of the pipelined algorithm, only the application of the preconditioner is pipelined as shown in Fig. 2 where the application of $\underline{\mathbf{M}}$ is still performed in parallel. We can observe that for the pipelined algorithm the runtime of a GMRES iteration is given by

$$5R\gamma(n) + 2(L+R)\gamma(n). \tag{16}$$

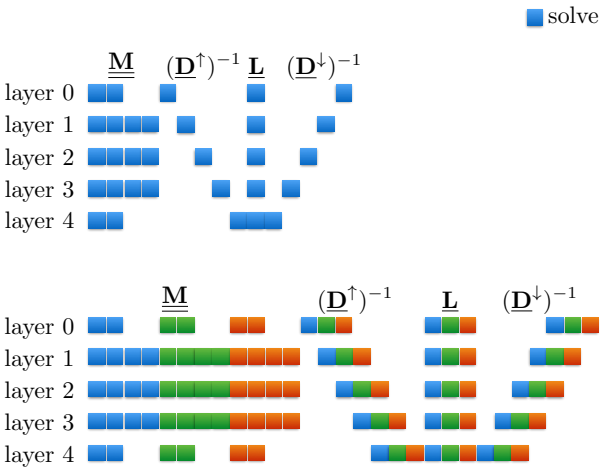

Figure 2: Sketch of the load of each node in the GMRES iteration for 1 rhs (top), and for 3 rhs treated in a pipelined fashion (bottom).

One of the advantages of the method of polarized traces is its lower memory requirement to store the intermediate representation of a solution compared to other methods. We only solve for the degrees of freedom involved in the SIE, which is $N^{2/3}n/q$, where q is the thickness of the layers in grid-points. This reduced memory requirement combined with the efficient preconditioner results in a lower memory footprint for the GMRES iteration.

COMPLEXITY

If we suppose that each layer has $\mathcal{O}(q \times n^2)$ grid points, i.e., they are q grid points thick. The complexity of multifrontal methods is known to be $\mathcal{O}(q^3n^3)$ for the factorization and $\gamma(n) = \mathcal{O}(q^2n^2 \log N)$ for the application. Then the complexity of the solver $\mathcal{O}(N \log N)$, given that we perform $L \sim n$ sequential solves to apply the preconditioner, and the number of iterations for convergences grows slowly.

For multiple right-hand sides, the situation is a bit different. We decompose the number of operations in the application of $\underline{\underline{\mathbf{M}}}$ and in the application of the preconditioner. The application of $\underline{\underline{\mathbf{M}}}$ to R right-hand sides can be done in $\mathcal{O}(Rq^2n^2 \log N)$ time. From Eq. 16, the application of the preconditioner can be performed to R right-hand sides in $\mathcal{O}(Lq^2n^2 \log N)$ as long as $R = \mathcal{O}(L)$. If R is larger than L, the rest of the right-hand sides are treated sequentially in $\mathcal{O}(Rq^2n^2 \log N)$ time. Now, using the fact that $L \sim n$ and the $N = n^3$, we obtain the advertised runtime of $\mathcal{O}(\max(1, R/n)N \log N)$.

NUMERICAL EXPERIMENTS

For the numerical experiments we used the SEAM model in Fig. 3. In order to support the claims, we solved the Helmholtz equation for different frequencies for one and for $R = \mathcal{O}(n)$ right-hand sides. We used MUMPS as a direct linear solver within each layer, the experiments were performed in a SGI cluster composed of nodes with dual Intel Xeon E5-2670 processors and 64 Gigabytes of RAM.

Table 1 shows the average online runtime for *one* and for R right-hand sides using the pipelined method of polarized traces. We observe that the number of iterations increases slowly with respect to the ω and N, and that the runtimes scale better than expected with respect to N, i.e., $\mathcal{O}(N \log N)$. This behavior is given by the large number of nodes available, but given the lack of scalability of the linear solver and increasing communication costs, we would expect the runtimes to increase, thus achieving the aforementioned asymptotic runtime.

Number of unknowns (N)	$6.5 \cdot 10^5$	$5.1 \cdot 10^6$	$4.2 \cdot 10^7$
Frequency [Hz]	0.75	1.5	3.0
Number of cores	11	88	880
Number of layers (L)	11	22	44
Number of rhs (R)	11	22	44
Number of iterations	6	6	8
Offline time [s]	24.3	80.7	141.8
Online time 1 rhs [s]	34.5	107.8	429
Online time R rhs [s]	154.1	757.6	2504.0

Table 1: Runtime for solving the Helmholtz equation with the SEAM model and number of iterations for a reduction of the residual to 10^{-7}, for problems of different sizes with an increasing number of rhs.

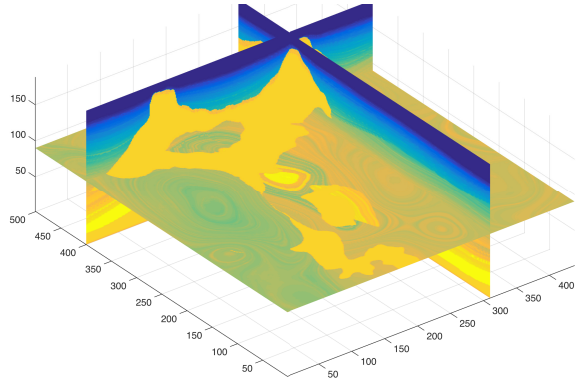

Figure 3: SEAM model.

CONCLUSION

We have presented a novel, fast and parallel solver for the high-frequency Helmholtz equation, which is able to solve R right-hand sides simultaneously with a *sublinear* asymptotic runtime $\mathcal{O}(\max(1, R/n)N \log N)$.

ACKNOWLEDGMENTS

We would like to thank TOTAL E&P, for its generous support; and the SEG Advanced Modeling Program (SEAM), for making available the SEAM model to us.

REFERENCES

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.

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.

Bebendorf, M., 2008, Hierarchical matrices: A means to efficiently solve elliptic boundary value problems: Springer-Verlag, volume **63** *of* Lecture Notes in Computational Science and Engineering (LNCSE). (ISBN 978-3-540-77146-3).

Bérenger, J.-P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: Journal of Computational Physics, **114**, 185–200.

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.

Chen, Y., 1997, Inverse scattering via Heisenberg's uncertainty principle: Inverse Problems, **13**, 253.

Chen, Z., and X. Xiang, 2013, A source transfer domain decomposition method for Helmholtz equations in unbounded domain part II: Extensions: Numerical Mathematics: Theory, Methods and Applications, **6**, 538–555.

Davis, T. A., 2004, Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method: ACM Transactions on Mathematical Software, **30**, 196–199.

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.

Duff, I. S., and J. K. Reid, 1983, The multifrontal solution of indefinite sparse symmetric linear: ACM Trans. Math. Softw., **9**, 302–325.

Engquist, B., and L. Ying, 2011a, Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation: Communications on Pure and Applied Mathematics, **64**, 697–735.

———, 2011b, Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers: Multiscale Modeling & Simulation, **9**, 686–710.

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.

Gander, M. J., and F. Nataf, 2005, An incomplete LU preconditioner for problems in acoustics: Journal of Computational Acoustics, **13**, 455–476.

George, A., 1973, Nested dissection of a regular finite element mesh: SIAM Journal on Numerical Analysis, **10**, 345–363.

Gillman, A., A. Barnett, and P. Martinsson, 2014, A spectrally accurate direct solution technique for frequency-domain scattering problems with variable media: BIT Numerical Mathematics, 1–30.

Johnson, S., 2010, Notes on perfectly matched layers (PMLs).

Liu, F., and L. Ying, 2015, Recursive sweeping preconditioner for the 3D Helmholtz equation: ArXiv e-prints.

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.

Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain; part 1: Theory and verification in a physical scale model: Geophysics, **64**, 888–901.

Rouet, F.-H., X. S. Li, P. Ghysels, and A. Napov, 2015, A distributed-memory package for dense Hierarchically Semi-Separable matrix computations using randomization: ArXiv e-prints.

Saad, Y., and M. H. Schultz, 1986, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems: SIAM J. Sci. Stat. Comput., **7**, 856–869.

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.

Stolk, C., 2013, A rapidly converging domain decomposition method for the Helmholtz equation: Journal of Computational Physics, **241**, 240–252.

Stolk, C. C., 2015, An improved sweeping domain decomposition preconditioner for the Helmholtz equation: ArXiv e-prints.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

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.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: GEOPHYSICS, **74**, WCC1–WCC26.

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.

Zepeda-Núñez, L., and L. Demanet, 2016, The method of polarized traces for the 2D Helmholtz equation: Journal of Computational Physics, **308**, 347 – 388.