

# Performance of Time and Frequency Domain Cluster Solvers Compared to Geophysical Applications

Victor Kostin<sup>1</sup>, Sergey Solovyev<sup>1</sup>✉, Andrey Bakulin<sup>2</sup> and Maxim Dmitriev<sup>2</sup>

<sup>1</sup> Institute of Petroleum Geology and Geophysics SB RAS,  
Novosibirsk, Russia

{solovevsa,kostinvi}@ipgg.sbras.ru

<sup>2</sup> Geophysics Technology, EXPEC ARC, Saudi Aramco  
Dhahran, Saudi Arabia.

{andrey.bakulin,maxim.dmitriev}@aramco.com

**Abstract.** In the framework of frequency-domain full waveform inversion (FWI), we compare the performance of two MPI-based acoustic solvers. One of the solvers is the time-domain solver developed by the SEISCOPE consortium. The other solver is a frequency-domain multi-frontal direct solver developed by us. For the high-contrast 3D velocity model, we perform the series of experiments for varying numbers of cluster nodes and shots, and conclude that in FWI applications the solvers complement each other in terms of performance. Theoretically, the conclusion follows from considerations of structures of the solvers and their scalabilities. Relations between the number of cluster nodes, the size of the geophysical model and the number of shots define which solver would be preferable in terms of performance.

**Keywords:** Geophysical problem, 3D acoustic solvers, Frequency-domain, Time-domain, Modeling, Sparse matrix, Low-Rank approximation

## 1 Introduction

Numerical simulation of acoustic wavefields is an important part of many algorithms developed to solve problems arising in exploration geophysics. In particular, it serves as an engine for full-waveform inversion (FWI) that is typically done in the frequency domain using a hierarchical multiscale strategy (e.g., [8], [10], [12] and references cited therein). For macro velocity reconstruction, repeated simulations are performed for a number of (usually) low frequencies (up to 10 Hz) at each iteration of the process. In simulations, the pressure wavefield is excited by a point source working as a harmonic oscillator at a particular frequency. In industrial applications, the number of shots can be tens of thousands or more, whereas receivers are even larger in quantity.

Any approach for frequency-domain acoustic wavefield simulation has both advantages and drawbacks that may be crucial in a particular situation. Here, we consider two representative solvers and compare their accuracy and performance.

2 Victor Kostin, Sergey Solovyev✉, Andrey Bakulin and Maxim Dmitriev

One conventional solver is based on time-domain simulation followed by Fourier transform. Alternative approaches directly tackle the Helmholtz equation. In 3D, it requires solving a system of linear equations with a huge sparse coefficient matrix. Iterative solvers are one notable member of this family [9], [2]. Comparatively new ideas of intermediate data compression [5],[13],[14] applied to solution algorithms for linear equation systems with sparse coefficient matrices make using direct solvers possible. Our second solver for comparison utilizes these ideas.

## 2 Method

The wave equation

$$\frac{\partial^2 p}{\partial t^2} - c^2(x, y, z)\Delta p = f(t)\delta(x - x_s, y - y_s, z - z_s), \quad (1)$$

can be solved in the time-domain for a particular point source position  $(x_s, y_s, z_s)$  and we can then apply the Fourier transform

$$\hat{p}(\omega, x, y, z) = \int e^{-i\omega t} p(t, x, y, z) dt \quad (2)$$

to obtain the frequency-domain solution. Here  $c(x, y, z)$  denotes velocity at point  $(x, y, z)$ ,  $f(t)$  is the source function,  $\delta$  denotes the Dirac delta function and  $p(t, x, y, z)$  is the pressure at point  $(x, y, z)$  and time  $t$ . Here we use a time domain solver developed by the SEISCOPE consortium (<https://seiscope2.osug.fr>). We refer to it as the time-domain finite-difference (TD) solver. Perfectly matched layers (PML) [3] are used to decrease the influence of the boundaries on the computational domain.

Provided the Fourier transform  $\hat{f}(\omega)$  of the source function is not zero at the frequencies of interest, the solutions for all frequencies become available at the same time. For different source point positions, computations are done independently of each other. This explains the ideal MPI scalability of the TD solution with respect to the number of shots when one shot is assigned per one cluster node.

In the frequency-domain approach, the Helmholtz equation

$$\Delta u + \frac{\omega^2}{c^2(x, y, z)} u = \delta(x - x_s, y - y_s, z - z_s) \quad (3)$$

is solved in the domain of interest. Supplying the equation with PMLs and zero boundary conditions one comes to a boundary value problem. The finite-difference approximation of the boundary value problem for equation (3) transforms this to a system of linear equations

$$Au = f. \quad (4)$$

Coefficient matrix A of this system is sparse, complex-valued (due to use of PMLs) and symmetric. Its non-zero elements depend on the values of velocity

$c(x, y, z)$  in the computational domain and on frequency  $\omega$ . The right-hand side vector  $f$  is defined by the source point position.

The multifrontal [4] direct approach to solve the system (4) consists of factorization of a permuted ( $P$  stands for some permutation) matrix

$$\hat{A} = PAP^t = LDL^t. \quad (5)$$

into a product of triangular matrices  $L, L^t$  and a diagonal matrix  $D$ , followed by solving a system with triangular coefficient matrices

$$\begin{aligned} y &= L^{-1}P^t f; \\ z &= D^{-1}y; \\ u &= PL^{-t}z. \end{aligned} \quad (6)$$

Factorization (5) is done once and can be used for all right-hand sides. This is the main advantage of the direct solution of boundary value problems for the Helmholtz equation with many source points. We refer to this approach as the frequency-domain finite-difference (FD) direct solver.

The direct approach under consideration faces a challenge called a *fill-in phenomenon* – the triangular factor  $L$  has many more non-zero elements than matrix  $A$  that prevents an unlimited increase of the number of grid points. To overcome this difficulty, we introduce intermediate data compression [11], [6] in the solver. In this way, non-zero ( $m_{ij} \times n_{ij}$ ) blocks  $L_{ij}$  of triangular matrix  $L$  are approximated by products

$$L_{ij} \cong U_{ij}V_{ij}^t \quad (7)$$

where  $U_{ij}$  is a  $m_{ij} \times r$  matrix,  $V_{ij}$  is a  $n_{ij} \times r$  matrix with small value of rank  $r$ . Instead of storing block  $L_{ij}$  factors,  $U_{ij}$  and  $V_{ij}$  are stored. This trick helps to reduce memory requirements by 5-6 times and also reduces the computational effort. To find matrices  $U_{ij}$  and  $V_{ij}$  approach based on randomized sampling [7] is used. This data compressed technique is similar to hierarchically semi separable (HSS) methods [5],[13],[14] and the respective solver is referred to below as the HSS solver.

The HSS technique is not sufficient by itself to solve huge problems. To scale this for larger systems, the factorization in the solver is implemented on distributed memory systems with data distributed among multiple cluster nodes.

### 3 Algorithms of MPI-Parallelization

The time-domain (TD) solver is easily parallelized and scales ideally with respect to number of right hand sides ( $N_{shots}$ ). If the number of cluster nodes  $N_{nodes} > N_{shots}$  one shot can be solved on each node without any data communications between nodes. Therefore, the computational time of the TD solver is

$$T_{TD}(N_{proc}, N_{shots}) = T_{TD}(1, 1)max\left(\frac{N_{shots}}{N_{proc}}, 1\right) \quad (8)$$

4 Victor Kostin, Sergey Solovyev, Andrey Bakulin and Maxim Dmitriev

and can be estimated by solution time of 1 shot on 1 node.

The frequency-domain (FD) solver uses another type of parallelization because of the large memory and computation requirements of  $LDL^t$  factorization. This factorization requires much more memory than the TD solver, so for each problem there is a minimum number of cluster nodes required to solve it.

Another obstacle of FD parallelization is the large number of sequential operations in the factorization process that prevents achieving ideal parallelization.

The idea of parallelizing the factorization step is based on the special structure of the A- and L-factors which results from the permutation process in equation (5) [4]. This structure can be associated with the elimination tree [6]. The details of such parallelization, including the scaling measurements are shown in paper [6]. The scalability of this approach is good for a small number of cluster nodes and becomes worse for many cluster nodes.

Performance can be improved in the case of many shots (i.e., when the solver time is more than the factorization time) by using a modification of MPI-parallelization for the solver. The details of this improvement are below.

HSS computational time is the sum of  $T_{FCT}$  (reordering+factorization) and  $T_{SLV}$  (solve time, i.e. inversion  $LDL^t$  factors) :

$$T_{HSS}(N_{proc}, N_{shots}) = T_{FCT}(N_{proc}) + T_{SLV}(N_{proc}, 1)N_{shots} \quad (9)$$

Starting from some number of nodes,  $T_{HSS}$  doesn't decrease with increasing  $N_{proc}$ . The reason is the poor scalability both of factorization and solve steps for many cluster nodes. Tests show  $T_{FCT}(N_{proc}) \approx T_{FCT}(\hat{N})$  and  $T_{SLV}(N_{proc}, 1) \approx T_{SLV}(\hat{N}, 1)$  for a constant  $\hat{N}$ .

One idea for improving scalability is separating all shots into groups and solve these groups in parallel on different sets of cluster nodes. The  $M = N_{proc}/\hat{N}$  groups (sets) have been proposed:

- set #1: Shots  $[1 \dots \frac{N_{shots}}{M}]$  are solved on the  $[1 \dots \hat{N}]$  nodes;
- set #2: Shots  $[\frac{N_{shots}}{M} + 1 \dots 2 * \frac{N_{shots}}{M}]$  are solved on the  $[\hat{N} + 1 \dots 2 * \hat{N}]$  nodes;
- ...
- set #M: Shots  $[(M - 1) * \frac{N_{shots}}{M} + 1 \dots M * \frac{N_{shots}}{M}]$  are solved on the  $[(M - 1)\hat{N} + 1 \dots N_{proc}]$  nodes.

The computational time can be written as:

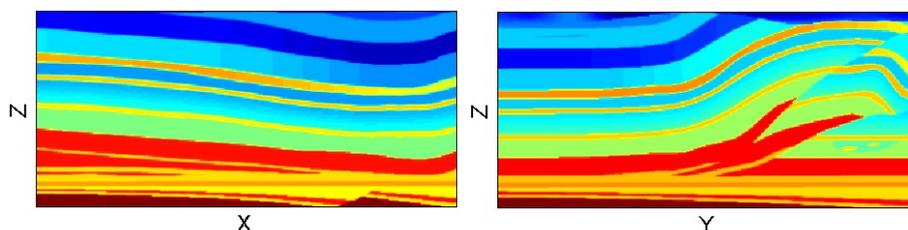
$$T_{HSS}(N_{proc}, N_{shots}) = \begin{cases} T_{FCT}(N_{proc}) + T_{SLV}(N_{proc}, 1)N_{shots}, & \text{if } N_{proc} \leq \hat{N} \\ T_{FCT}(\hat{N}) + T_{SLV}(\hat{N}, 1)\hat{N}max\left(\frac{N_{shots}}{N_{proc}}\right), & \text{if } N_{proc} > \hat{N}. \end{cases} \quad (10)$$

Selecting  $\hat{N}$  can be done through trial and error. The final scalability and computation time both for time-domain and frequency domain solvers are presented below

## 4 Numerical Experiments

Performance and accuracy tests were run on the Shaheen II supercomputer (<https://www.hpc.kaust.edu.sa/content/shaheen-ii>) with the following hardware characteristics: 2×Intel®Xeon®CPU E5-2698 v3 @2.3 GHz per cluster node, 128 GB RAM/per node. Theoretical Peak (Rpeak) performance of one node is 1,2TFlop/s. The main data storage solution is a Lustre Parallel file system based on Cray Sonexion 2000 with a usable storage capacity of 17.2 PB delivering around 500 GB/s of I/O throughput.

For the purpose of numerical comparison, we use a Overthrust (referred below as OT) model (see [1]) with dimensions of  $9 \times 9 \times 4.5$ km. In this model, velocity varies between 2 300 m/s and 6 000 m/s. Example model cross-sections are shown in Figure 1.



**Fig. 1.** The OT velocity model showing cross-sections in the X-Z plane (left), and the Y-Z (right).

Table 1 lists the parameters of the numerical experiments used to evaluate the solvers in the OT model. To provide approximately the same number of

**Table 1.** Parameters of numerical experiments.

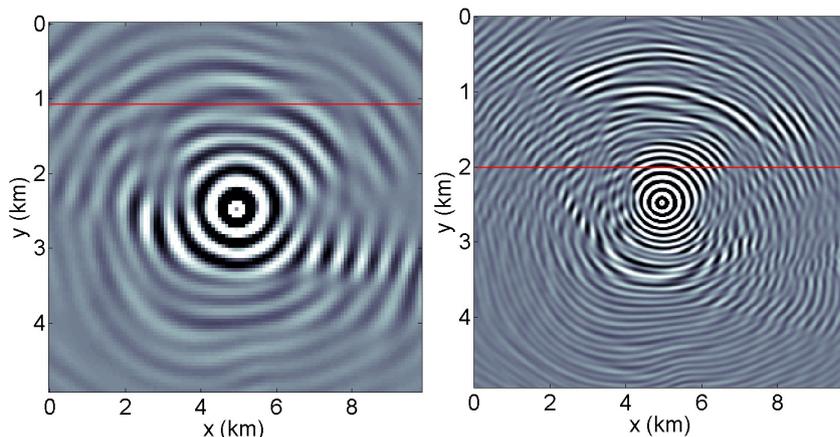
$\nu(Hz)$	$h(m)$	$ppw$	$N_x$	$N_y$	$N_z$	$N$	$dt(s)$
5	90	5.1	110	110	52	$0.6 \cdot 10^6$	0.00742
7	60	5.4	165	165	78	$2.1 \cdot 10^6$	0.00495
15	30	5.1	330	330	155	$17 \cdot 10^6$	0.00247

points per wavelength (ppw) for frequencies 5, 7 and 15, we computed solutions using different grid steps,  $h$ , in meters. The corresponding numbers of cells along each axis are shown in columns  $N_x$ ,  $N_y$  and  $N_z$  along with the total numbers of grid points ( $N$ ). The time discretization step used for TD is shown in the last column. The TD solver computations were performed using a time interval  $[0, 10s]$ .

The sources were placed at a depth of one grid point at the center of the domain. Receivers were placed throughout the X-Y plane at the same depth. To

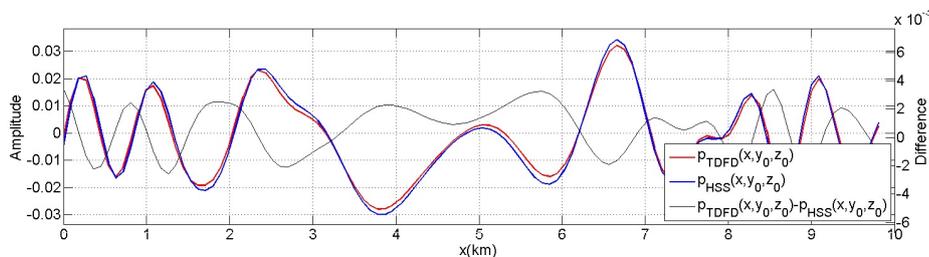
6 Victor Kostin, Sergey Soloviyev, Andrey Bakulin and Maxim Dmitriev

demonstrate that the HSS solver is working correctly, we provide 2D snapshots of the real part of the computed wavefield taken at the depths of the receiver plane (Figure 2). Figures 3 and 4 show TD and HSS solutions along selected profiles from Figure 2, as well as the magnified difference between two solutions.



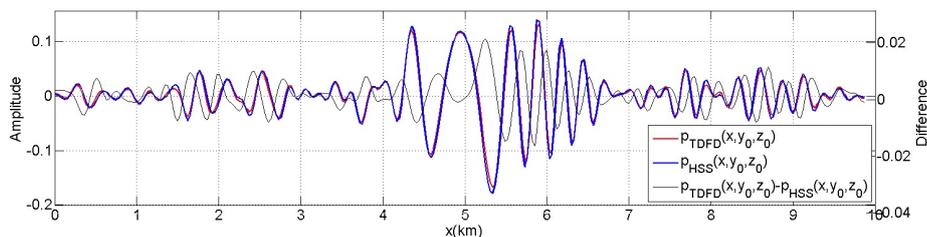
**Fig. 2.** Snapshots of the solutions (real part) computed at the plane of surface receivers using a HSS solver. The left picture is for 5 Hz, and the right is for 15 Hz.

Good agreement can be observed between curves in Figure 3 for the 5 Hz and in Figure 4 for the 15 Hz solutions.



**Fig. 3.** Real part of the frequency-domain solution at 5 Hz shown along the red profile from Figure 2. Red and blue lines correspond to TDFD and HSS solutions, respectively; black line represents the magnified (x5) difference between them.

Four metrics were computed to quantify the difference between solutions. Three of them are simple relative differences of two grid functions  $u$  and  $v$  computed using classical functional norms:



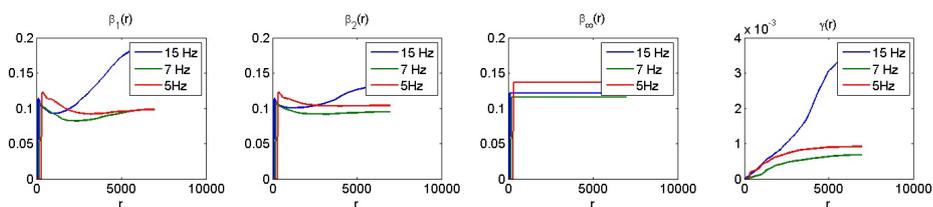
**Fig. 4.** Real part of the frequency-domain solution at 15 Hz shown along the red profile from Figure 2. Red and blue lines correspond to TDFD and HSS solutions, respectively; black line represents the magnified (x5) difference between them.

$$\beta_k(u, v) = \frac{\|u - v\|_k}{\|u\|_k}, \quad k = 1, 2, \infty \quad (11)$$

If computations of the norms in formula (11) are done for spheres of radius  $r$  and centered at the source point, one gets function  $\beta_k(r)$ . The wavefields  $u, v$  in formula (11) have singularities at the source locations. Therefore, we exclude a sphere of small radius  $r_0$  around the source from the volume of interest. The last (fourth) metric is computed in a similar way using:

$$\gamma(u, v) = \left| 1 - \frac{(u, v)}{\|u\| \|v\|} \right|. \quad (12)$$

The function  $\gamma(r)$  is getting by the similar way as  $\beta_k(r)$ . Graphs of functions  $\beta_k(r)$  and  $\gamma(r)$  are shown in Figure 5. The radius  $r_0$  of a small excluded sphere was taken to be equal to  $10h/3$ . We clearly observe that the discrepancy between the two solvers generally increases with the volume radius and frequency, as is generally expected. A somewhat larger jump between 5 and 15 Hz may suggest insufficient grid size for the complex OT model with high acoustic contrasts.



**Fig. 5.** Difference between solutions obtained with Seiscope and HSS solvers measured as a function of the domain size.

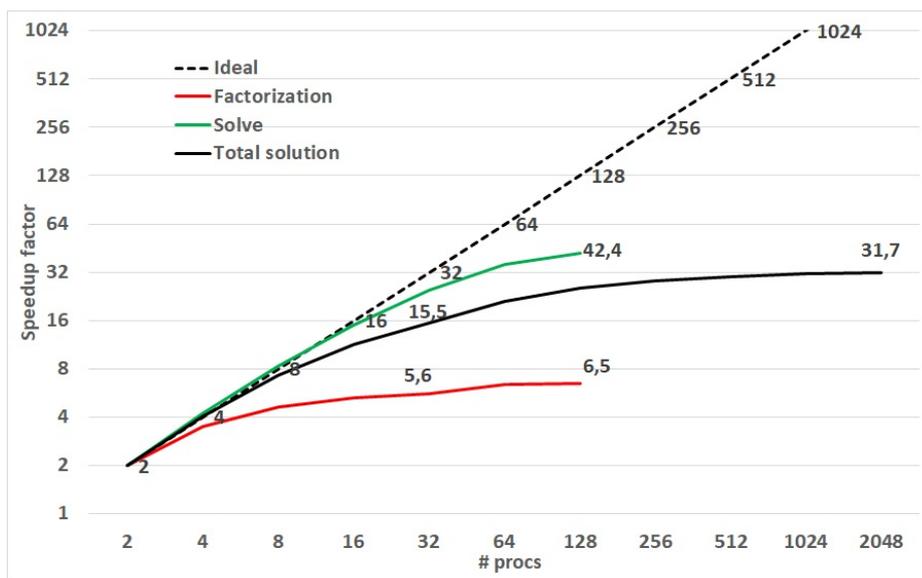
## 5 Comparison of Numerical Performance

The relative performance of the TD and HSS solvers is measured on various number of cluster nodes ( $N_{nodes}$ ) and number of shots ( $N_{shots}$ ).

To get a TD solution for specific frequencies for FWI such as  $5Hz$ ,  $7Hz$  and  $15Hz$  the forward problem is solved in the time domain (1) on a  $30m$  mesh and the result forward Fourier transformed to extract the frequency information. The total computational time for the TD solver is used for comparison with the FD solver.

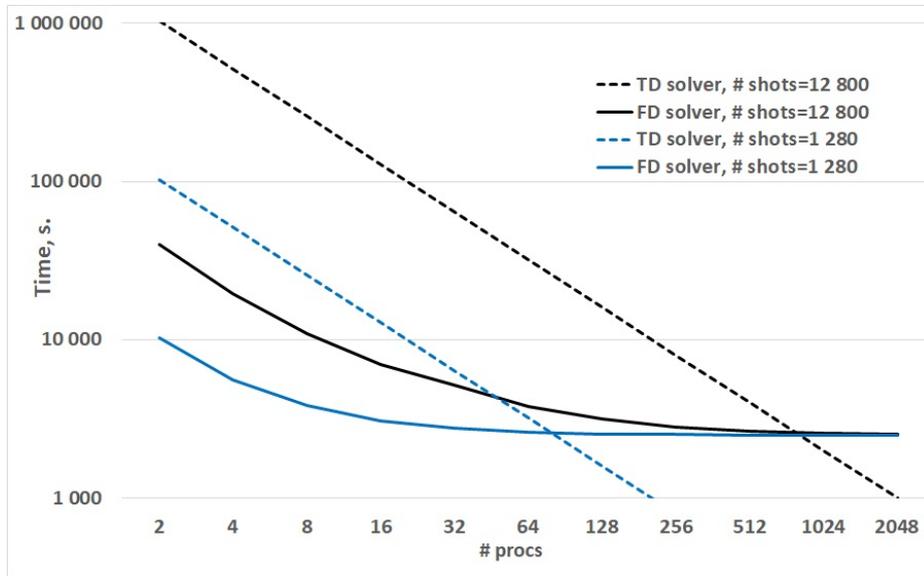
To get a solution using the FD solver for these frequencies, the Helmholtz problem (3) is solved for each frequency  $5Hz$ ,  $7Hz$  and  $15Hz$  using parameters from Table 1. The sum of these three FD times are presented below. The computational times for coarse ( $90m$ ) and medium ( $60m$ ) meshes are significantly less than times for a fine mesh ( $30m$ ). Table 2 shows the ratio of the TD and FD solver computation times. The sum of times for three meshes is used for the  $T_{HSS}$  values.

The scaling is shown in the Figure 6.



**Fig. 6.** Scalability of FD solver (HSS) for 12 800 sources (measured up to 128 cluster nodes and estimated up to 2048 ones).

For particular pairs of values ( $N_{nodes}, N_{shots}$ ), the solvers run times (in seconds) are provided in Table 2 in the form of fractions, where run time  $T_{TD}$  is put as the numerator and  $T_{HSS}$  as the denominator.



**Fig. 7.** Timing of TD and FD solvers (measured up to 128 cluster nodes and estimated up to 2048 ones).

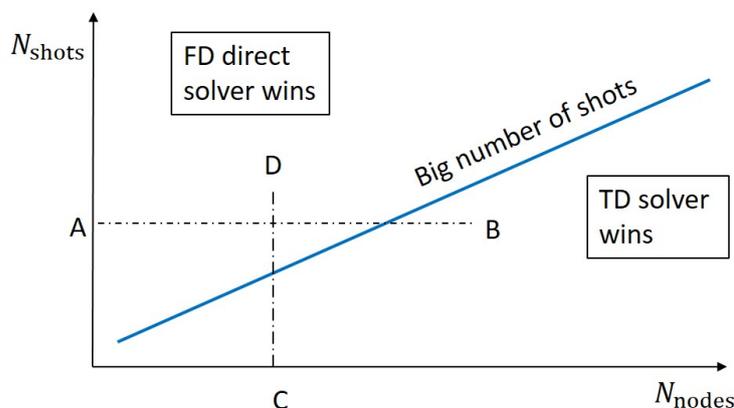
**Table 2.** Timing results for two solvers represented as ratios with  $T_{TD}/T_{HSS}$ . Note that for ratios  $> 1.0$  the HSS solver becomes relatively more efficient.

$N_{shots}$	1	128	1280	12800
$N_{nodes}$				
32	161/2945	644/2978	6440/3275	64400/6245
64	161/2625	322/2648	3220/2857	32200/4945
128	161/2555	161/2575	1610/2757	16100/4570

10 Victor Kostin, Sergey Soloviyev, Andrey Bakulin and Maxim Dmitriev

The TD solver scales ideally with respect to  $N_{shots}$ . Measured run time for one shot on one node is 161 sec. 128 shots can be computed on 32 nodes for 644 sec, on 128 nodes for 161 sec, but further increases in  $N_{nodes}$  do not lead to run time reduction. Scalability of the HSS solver for one shot is determined by the scalability of factorization and decays with increasing  $N_{nodes}$ . Increasing  $N_{shots}$  gives additional opportunity to improve scalability.

Figure 8 illustrates how the FD and TD solvers complement each other when comparing  $N_{shots}$  and  $N_{nodes}$ .



**Fig. 8.** Relative performance of the TD and FD solvers shown with number of shots (vertical axis) versus number of nodes (horizontal axis). The blue line defines the line of equal performance.

The blue line in Figure 8 defines the line of equal performance of the two solvers. For a given number of nodes ( $N_{nodes}$ ), this line defines the number of shots that is "big enough" to fully reap the benefits of the HSS solver and reach the numerical performance of the TD solver. For example, for a problem with a comparatively small number of shots which has to be solved on a particular cluster (a point close to point C of line segment CD), the fastest way would be using the TD solver. If the number of shots increases (the point moves to D along CD) then a FDFD direct solver becomes faster. Another way to look at it is to fix the number of shots, such as defined by horizontal line AB. Then moving from point A towards B along horizontal segment AB means solving the same problem on clusters with increasing number of nodes. For a fixed number of shots, a FDFD direct solver would usually be faster for comparatively small clusters, but on larger clusters TD is more efficient.

## 6 Conclusions

We compare two solvers for the acoustic wave equation for different cluster sizes and variable numbers of shots. The time-domain finite-difference (TD) solver scales perfectly, with effort linearly increasing with number of shots. In contrast, a direct frequency-domain HSS solver obtains the global solution for the entire domain and all the shots. HSS computational effort is non-linearly dependent on available resources, and the benefit/cost ratio increases sharply with increasing number of shots. We have performed a series of numerical experiments with real-world scenarios using the Overthrust velocity model and the Shaheen II supercomputer. These experiments demonstrate the existence of the line of equal performance, comparing number of shots versus number of available nodes. The TD solver usually wins in the segment of comparatively large values of the ratio  $N_{nodes}/N_{shots}$  and the HSS solver wins where the ratio is lower.

While generally larger numbers of nodes become available with time (as computing power becomes cheaper), this is offset by the rapidly increasing trace density of seismic acquisition systems that are constantly growing in number of shots and receivers per square kilometer. These two concurrent trends would likely maintain a need for both types of solvers depending on exact survey geometry, available computing power and number of shots for a particular FWI problem. Therefore, an optimal FWI toolbox should contain both solvers so that the most efficient one can be used for a specific scenario based on the line of equal performance revealed in this study

**Acknowledgments.** We also appreciate KAUST for providing access to Shaheen II supercomputer.

## References

1. Aminzadeh, F., J. Brac, and T. Kuntz, 1997, 3-D salt and overthrust models, in SEG/EAGE modelling series: SEG Book Series.
2. Belonosov, M., M. Dmitriev, V. Kostin, D. Neklyudov and V. Tcheverda, 2017, An Iterative Solver for the 3D Helmholtz Equation: Journal of Computational Physics, 345, 330-344, doi: 10.1016/j.jcp.2017.05.026
3. Collino F., Tsogka C. Application of the perfectly matched layer absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media Geophysics. 2001. 66. 294–307.
4. Duff, I. S., and J. K. Reid, 1983, The multifrontal solution of indefinite sparse symmetric linear systems: ACM Transactions on Mathematical Software (TOMS), 9, no. 3, 302-325, doi: 10.1145/356044.356047.
5. Hackbusch, W., 1999, A sparse matrix arithmetic based on H-Matrices. Part I: Introduction to H-matrices: Computing, 62, 2, 89-108, doi: 10.1007/s006070050015.
6. Kostin, V., S. Solovyev, H. Liu, and A. Bakulin, 2017, HSS cluster-based direct solver for acoustic wave equation: 87th Annual International Meeting, SEG Technical Program Expanded Abstracts, 4017-4021.

12 Victor Kostin, Sergey Solovyev✉, Andrey Bakulin and Maxim Dmitriev

7. Martinsson, P.G., 2011, A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix: *SIAM Journal on Matrix Analysis and Applications*, 32, no. 4, 1251-1274.
8. Mulder, W.A., and R.E. Plessix, 2004, How to choose a subset of frequencies in frequency-domain finite-difference migration: *Geophysical Journal International*, 158, 3, 801-812.
9. Plessix, R.E., 2007, A Helmholtz iterative solver for 3D seismic-imaging problems: *Geophysics*, 72, 5, SM185-SM194, doi: 10.1190/1.2738849
10. Shin, C. and Y.H. Cha, 2008, Waveform inversion in the Laplace domain: *Geophysical Journal International*, 173, 3, 922-931.
11. Solovyev, S., D. Vishnevsky and H. Liu, 2015, Multifrontal hierarchically semiseparable solver for 3D Helmholtz problem using optimal 27-point finite-difference scheme: *77th EAGE Conference and Exhibition Expanded Abstracts*.
12. Virieux, J., S. Operto, H. Ben-Hadj-Ali, R. Brossier, V. Etienne, F. Sourber, L. Giraud and A. Haidar, 2009, Seismic wave modeling for seismic imaging: *The Leading Edge*, 28, 5, 538-544.
13. Wang, S., M. de Hoop 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.
14. Xia, J., 2013, Efficient structured multifrontal factorization for large sparse matrices: *SIAM Journal of Scientific Computing*, 35, A832-A860, 10.1137/1208670