OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 1 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

PARALLEL PERFORMANCE OF THE EXPLICIT FDTD METHODS APPLIED ON 3D ACOUSTIC WAVE EQUATION

Nabi Bux Kalhoro*
Department of Basic Science and Related Studies, The University of Larkano, Larkana Sindh Pakistan.

Shakeel Ahmed Kamboh
Department of Mathematics and Statistics, Quaid e Awam University of Engineering, Science and Technology, Nawabshah, Sindh, Pakistan

Afaque Ahmed Bhutto
Department of Basic Science and Related Studies, The University of Larkano, Larkana Sindh Pakistan.

Saeed Ahmed Rajput
Department of Basic Science and Related Studies, The University of Larkano, Larkana Sindh Pakistan.

Mansoor Ali Khaskheli
Department of social science Faculty of Biosciences Shaheed Benazir Bhutto University of Veterinary and Animal Science Sakrand Sindh, Pakistan

Jan Muhammad Shah
Department Basic science and related studies, The university of larkano, Sindh Pakistan

[*]***Corresponding author: Nabi Bux Kalhoro (**drnb_kalhoro@uolrk.edu.pk**)***

Article Info

Abstract

The 3D acoustic wave equation is a second-order linear hyperbolic partial differential equation. It is widely studied in acoustics, fluid dynamics, electromagnetism, and seismology within both science and engineering fields. Among the various numerical methods, the finite difference method is considered to be the simpler to understand and easy to implement. Based on the discretize schemes finite difference time domain method described by the explicit scheme in which spatial second order derivatives are evaluated at the previous time step. The solution of 3D acoustic wave equation may be required on the high-resolution mesh points consequently more computational resources are required. In this study parallel algorithm for the numerical solution of 3D acoustic wave equation is proposed and designed by using data parallel approach and message passing schemes. The algorithm is implemented on shared memory parallel system using MATLAB parallel computing mode. The parallel performance of the designed algorithm is analyzed on different mesh sizes and time steps. It is revealed that the proposed algorithm may reduce computational time up to 3 times as compared to sequential solution algorithm. The proposed parallel algorithm remains more efficient on P=2,3 and 4 workers while for P>4 the efficiency of the algorithm drops because of the high communication time. The results of the proposed research could be utilized in the large-scale simulations of 3D acoustic wave equations and may enable to simulate the acoustic wave pressure at more refined meshes on high performance computing systems.

Keywords: *Parallel Computing, Finite Difference Time Domain (FDTD), 3D Acoustic Wave Equation, Numerical Simulation, MATLAB Implementation.*

Introduction

The second order wave equation is a linear hyperbolic type partial differential equation arises in the modeling and simulation of sound waves, water waves and light waves. This equation is extensively studied in the fields of acoustics, fluid dynamics, electromagnetism, seismology and has numerous applications in science and engineering (Mehra et al., 2012). In one dimensional space the wave equation can be used to describe the vibration of an ideal string, in 2D it can describe the vibration of an ideal drum or membrane, and in 3D space it is often used to model and simulate the propagation of sound waves in gas or liquid (Otero et al., 2020).

The 3D acoustic wave equation models the propagation of acoustic waves in fluids, such as liquids and gases. An important application of this equation, three-dimensional, is in seismic exploration, widely used in the search for subsurface resources such as crude oil, natural gas, and minerals (Bernacki, Lanteri, et al., 2006) (Frances et al., 2015). The equation describes the time evolution of acoustic pressure, p or particle velocity V, as functions of spatial coordinates (x, y, z), and time t (Näsholm & Holm, 2013). A localized change in pressure compresses the surrounding fluid, causing further pressure variations, which propagates as acoustic waves. This dynamic behavior is mathematically described by the 3D acoustic wave equation (Garcia, 2009). Typical boundary conditions, aside from absorbing boundary conditions that "simulate" an infinite medium by preventing reflection, include zero-velocity or zero-displacement conditions for rigid surfaces and stress-free conditions (Kamboh et al., 2015) (Savioja, 2010). Such conditions are critical in simulating realistic situations and obtaining reasonably accurate simulations. Analytical solutions are available for specific, simple or idealized situations. For more complex and realistic cases, numerical methods must be used. These methods approximate the governing partial differential equations by converting them into systems of algebraic equations which can easily be implemented on computers. Among the most widely used numerical methods are finite difference methods (FDM), finite element methods (FEM), and finite volume methods (FVM). The acoustic modeling method choice would depend, for example, on its ability to deal with medium heterogeneity, scale and dimensions of the simulation domain, realistic attenuation, free-surface topography, and frequency range (Bernacki, Lanteri, et al., 2006). Again, computing efficiency in terms of memory usage and processing time is critical for practical applications. For instance, Finite difference time-domain (FDTD) methods represent the most common numerical approach for generation solutions of the 3D acoustic wave equation and have long been developed and widely implemented since the 1980s (Robertsson et al., 2012). The finite difference coefficients for spatial derivatives can be derived from the time-space domain dispersion relation (Y. Wang et al., 2014) and the Taylor series expansions can also be used (Zhang & Yao, 2012) (Chang & Liu, 2013). Further Kamboh (Kamboh et al., 2015) extended the Least Square-based optimal finite difference scheme from two-dimensional (2D) forward modeling to three-dimensional (3D) and developed a 3D acoustic optimal finite difference method with high efficiency, wide range of high accuracy and adaptability to parallel computing. A finite difference approach in the time domain using a combination of coordinate mapping and differential geometry by minimizing the error of the space domain dispersion relations is developed by (Shragge, 2014). Khokhar (Khokhar et al., 2023) investigates the behavior of Newtonian fluids in pipes filled with and without porous media under combing and separating flow configurations. To calculate the finite difference coefficients, a new 3D time-space domain finite difference method with minimized error of the time-space domain dispersion relation by using Taylor expansion was proposed by (Liu & Sen, 2009). Typically, the FDTD methods use rectangular grids although there are numerous other possible mesh topologies as well. For the room acoustic modeling, the wave-based methods and the ray-based method were studied by (Savioja, 2010). They concluded that the former target at solving the wave equation numerically whereas the latter neglect all the wave phenomena and sound act like rays. In (Kowalczyk & Van Walstijn, 2011) a method for constructing the boundary formulations for the general family of 3D non-staggered compact explicit FDTD schemes is proposed.

The first-order Mur's boundary condition is used for free space simulation (Sheu & Li, 2008). However, in practice the first-order Mur's boundary condition is poor in eliminating reflected waves from truncated edges and causes errors in simulation. They overcame this shortcoming by using Berenger's perfectly matched layers (PMLs) to minimize the interference. The sequential simulation of 3D acoustic wave equations using FDTD often becomes very slow on sequential computers therefore the need of parallel computing resources is inevitable. In this regard, to improve the efficiency of the FDTD for 3D acoustic wave equation with random absorbing boundary condition (ABC) was proposed by (Sheua & Lia, n.d.) (LIU et al., 2013) for the implementation on parallel graphic processing units. An explicit, time-domain, finite-difference (FD) numerical scheme is used to solve the system for both pressure and particle velocity wave fields on staggered spatial and temporal grids was proposed by (Garcia, 2009). The algorithm is designed to execute on parallel computational platforms by utilizing a classical spatial domain-decomposition strategy. Yoon (Yoon et al., 2003) worked some recent developments would allow 3D reverse-time migration to be done relatively inexpensively on PC-based distributed memory clusters. The new technique is computationally not intensive. However, they compare reverse-time migration images with first-arrival Kirchhoff migration images to demonstrate that 3D reverse-time migration can produce high fidelity images under the PC-based distributed memory cluster machine. Many researchers (Saarelma, 2013)(Bernacki, Fezoui, et al., 2006; Operto et al., 2007; Vaccari et al., 2011; S. Wang et al., 2012; Yu et al., 2005) parallel solvers based on FDTD solution of 3D acoustic equation on large clusters and GPU (graphics processing unit) have been proposed for the purpose of increasing the efficiency of the method and speedup of simulation. However, the parallel performances of explicit FDTD method have not been reported.

As the size of 3D acoustic wave simulations grows, it becomes computationally infeasible to use sequential solvers due to the computing time and memory requirements involved. Large-scale problems of this nature are usually only tractable on supercomputing infrastructure or HPC systems. The parallel performance of the 3D acoustic wave equation has been mostly analyzed in existing literature with explicit finite-difference time-domain (FDTD) methods implemented on advanced hardware, such as GPUs or specially designed HPC systems, which are very expensive and normally not accessible to all researchers (Allen et al., 2005)(J. D. Sheaffer et al., 2011)(Li et al., 2022)(J. Sheaffer & Fazenda, 2010)(Morales et al., 2017). There is, however, now an increasing need to investigate the feasibility and efficiency issues by running these simulations on relatively inexpensive, commonly available computing systems. For example, a parallel performance benchmark of explicit FDTD methods within MATLAB on low-cost platforms would be of interest. MATLAB is a very intuitive language with extensive, built-in parallel computing tools that could be opened for easier use by researchers without access to advanced computing resources. Exploration of this avenue would be expected to not only increase accessibility to the carrying out of large-scale acoustic simulations but also advance optimized and cost-effective solutions for actual applications in real-life problems.

This study presents the numerical solution of the 3D acoustic wave equation with the FDTD method. A parallel algorithm for the FDTD method is developed and implemented on low-cost, readily available shared memory systems using MATLAB. The performance of the parallel explicit FDTD implementation is evaluated and analyzed, with an emphasis on key metrics such as speedup and computational efficiency.

2. Methodology: The governing equation 3D wave propagation of an acoustic wave equation and mathematically described by the following linear partial differential equation (Cai et al., 2015):

$$\frac{\partial^2 p(x,y,z,t)}{\partial t^2} = c^2\left(\frac{\partial^2 p(x,y,z,t)}{\partial x^2} + \frac{\partial^2 p(x,y,z,t)}{\partial y^2} + \frac{\partial^2 p(x,y,z,t)}{\partial z^2}\right), \qquad (1)$$

where the coefficient c in the equation (1) is related to the stability condition of the 3D acoustic equation and is defined as follows.

$c = \sqrt{K/\rho}$ is the speed of sound, $K = 24.42$ is the coefficient of the stiffness (bulk modulus), Gpa and $\rho = 2.16\left(\frac{kg}{m^3}\right)$ is the density of material, here salt is used as working material.

2.1 Formulation of Computational Domain with the Initial and Boundary Conditions:

In order to test and implement the methodology first the dimensions and boundary conditions are defined properly. The dimensions are taken from the SEG/EAGE (Society of Exploration Geophysicists (SEG), the European Association of Geoscientists and Engineers (EAGE), the Australian Society of Exploration Geophysicists) salt model (Chang & Liu, 2013; Näsholm & Holm, 2013; Yu et al., 2005). Figure 1 depicts the size of 13500 m × 13500 m × 4180 m which is a benchmarking model that is used to test and implement the proposed methodology. Whereas the absorbing boundary conditions are set by selecting an interior region of the domain with initial pressure field. The top of surface of the domain has naturally atmospheric pressure while other faces are set to zero pressure.

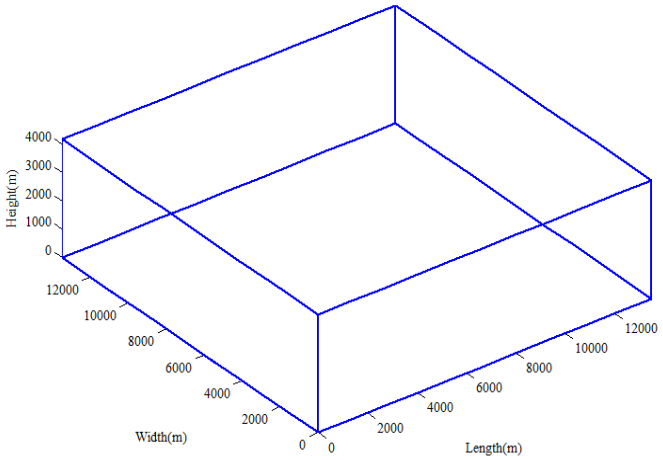


Figure 1: Schematic of computational domain

Depending upon the problem Eq. (1) is associated with the initial and boundary conditions like absorbing boundary conditions (for simulating an infinite medium), zero-velocity conditions equivalent to zero-displacement conditions (Savioja, 2010).

The boundary conditions on six faces are listed as,

$$p(13500, y, z, t) = 0, p(x, 0, z, t) = 0, p(x, 13500, z, t) = 0, p(0, y, z, t) = 0 \qquad (2)$$

$$p(x, y, 0, t) = 0, p(x, y, 4180, t) = p_{atm}, p(x, y, z, 0) = 0. \qquad (3)$$

And the source wave in the central region of the domain is chosen as follows:

$$p(x_c, y_c, z_c, 0) = 10\,cos(K(x_c + y_c + z_c)), \qquad (4)$$

$$3375 \le x_c \le 8775, \quad 3375 \le y_c \le 8775, \quad 1045 \le z_c \le 2717. \qquad (5)$$

3. The Explicit Finite Difference Discretization of 3D Acoustic Wave Equation and its Simplification:

The explicit FDTD representation of the model equation are obtained by approximating the derivatives of Eq. (1) using second order accurate finite difference schemes as follows:

$$\frac{\partial^2 p}{\partial t^2} \approx \frac{p_{i,j,k}^{n+1} - 2p_{i,j,k}^{n} + p_{i,j,k}^{n-1}}{\Delta t^2}, \qquad \frac{\partial^2 p}{\partial x^2} \approx \frac{p_{i+1,j,k}^{n} - 2p_{i,j,k}^{n} + p_{i-1,j,k}^{n}}{\Delta x^2}, \qquad (6)$$

$$\frac{\partial^2 p}{\partial y^2} \approx \frac{p^n_{i,j+1,k}-2p^n_{i,j,k}+p^n_{i,j-1,k}}{\Delta y^2}, \quad \text{and} \quad \frac{\partial^2 p}{\partial z^2} \approx \frac{p^n_{i,j,k+1}-2p^n_{i,j,k}+p^n_{i,j,k-1}}{\Delta z^2}. \tag{7}$$

Thus, the explicit discretization of Eq (1) is obtained

$$\frac{p^{n+1}_{i,j,k}-2p^n_{i,j,k}+p^{n-1}_{i,j,k}}{\Delta t^2} = c^2\left(\begin{matrix}\frac{p^n_{i+1,j,k}-2p^n_{i,j,k}+p^n_{i-1,j,k}}{\Delta x^2} + \frac{p^n_{i,j+1,k}-2p^n_{i,j,k}+p^n_{i,j-1,k}}{\Delta y^2} + \\ \frac{p^n_{i,j,k+1}-2p^n_{i,j,k}+p^n_{i,j,k-1}}{\Delta z^2}\end{matrix}\right), \tag{8}$$

Equation (8) is further simplified for the solution at each interior node (i, j, k) of the mesh at $(n + 1)^{th}$ time step as follows:

$$p^{n+1}_{i,j,k} = \frac{1}{\Delta x^2\Delta y^2\Delta z^2}\left(\begin{matrix}\left(2\Delta x^2\Delta y^2\Delta z^2 - 2c^2\Delta t^2(\Delta x^2\Delta y^2 - \Delta y^2\Delta z^2 - \Delta x^2\Delta z^2)\right)p^n_{i,j,k} + \\ (c^2\Delta t^2\Delta y^2\Delta z^2)\left(p^n_{i+1,j,k} + p^n_{i-1,j,k}\right) + (c^2\Delta t^2\Delta x^2\Delta z^2)\left(p^n_{i,j+1,k} + p^n_{i,j-1,k}\right) + \\ (c^2\Delta t^2\Delta x^2\Delta y^2)\left(p^n_{i,j,k+1} + p^n_{i,j,k-1}\right) - (\Delta x^2\Delta y^2\Delta z^2)p^{(n-1)}_{i,j,k}\end{matrix}\right), \tag{9}$$

Or equation (9) is further simplified as;

$$\frac{1}{\Delta t^2}p^{n+1}_{i,j,k} + \left(\frac{2c^2}{\Delta x^2} + \frac{2c^2}{\Delta y^2} + \frac{2c^2}{\Delta z^2} - \frac{2}{\Delta t^2}\right)p^n_{i,j,k} + \frac{1}{\Delta t^2}p^{n-1}_{i,j,k} - \frac{c^2}{\Delta x^2}p^n_{i+1,j,k} - \frac{c^2}{\Delta x^2}p^n_{i-1,j,k} - \frac{c^2}{\Delta y^2}p^n_{i,j+1,k} - \frac{c^2}{\Delta y^2}p^n_{i,j-1,k} - \frac{c^2}{\Delta x^2}p^n_{i,j,k+1} - \frac{c^2}{\Delta x^2}p^n_{i,j,k-1} = 0 \tag{10}$$

above equation (10) leads to a system of linear equations AP-b=0.
The finite difference approximation of 3D acoustic wave equation (10) is stable (S. Wang et al., 2012) iff,

$$\Delta t \leq \frac{min(\Delta x,\Delta y,\Delta z)}{\sqrt{2}c_{max}}, \tag{11}$$

where $c_{max}$is the maximum wave velocity in the medium. Once the explicit finite difference scheme of the acoustic equation is done then the domain is discretized into a 3D mesh of octahedral elements as shown in Figure 2.

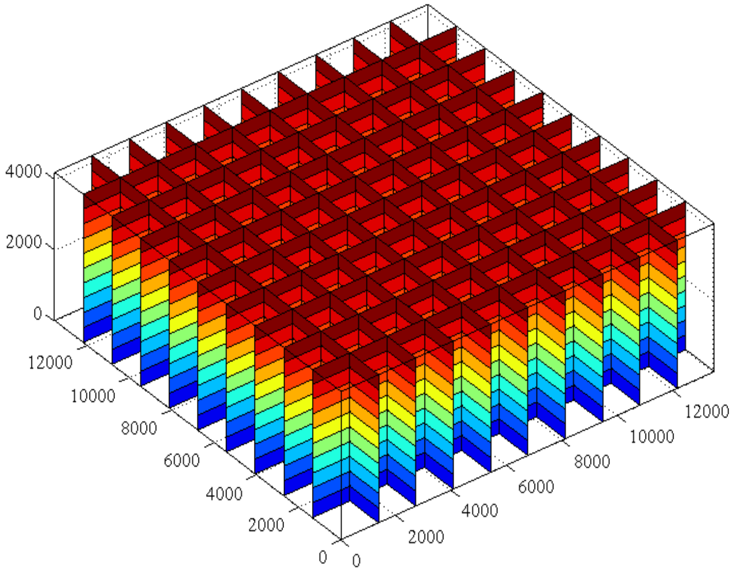


Figure 2:A schematic of 3D finite difference discretization of the domain into$10x10x10$ mesh.

4. Sequential Implementation of FDTD Numerical Solution on MATLAB:

The Equation (10) is solved iteratively by writing a sequential MATLAB code and is tested on a single worker of MALAB. The acoustic pressure distribution is obtained on each interior node. The simulation profiles on different mesh sizes are achieved in Figure 3(a-c).

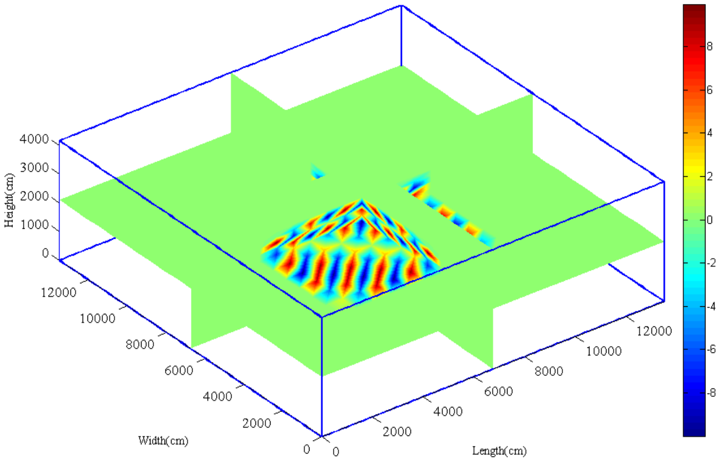


Figure 3(a): Simulation profile of acoustic pressure p at $dt = 0.1$ and $20x20x20$ mesh size $(gr = 2)$

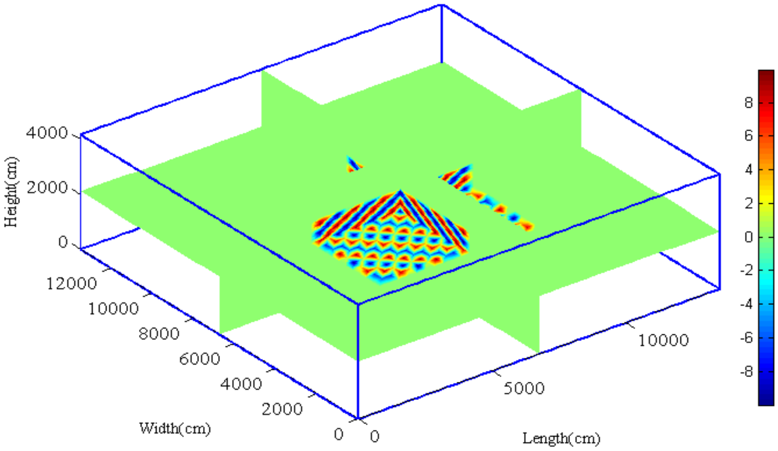


Figure 3(b): Simulation profile of acoustic pressure p at $dt = 0.1$ and $40x40x40$ mesh size $(gr = 4)$

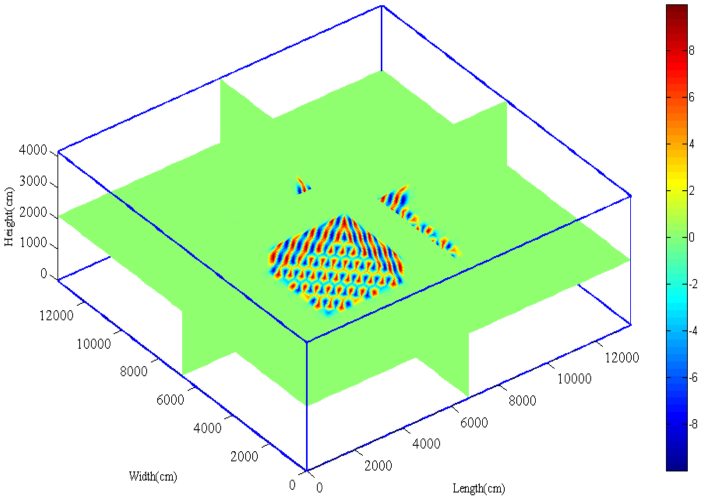


Figure 3(c): Simulation profile of acoustic pressure p at $dt = 0.1$ and $50x50x50$ mesh size $(gr = 5)$

5. Design of Parallel Algorithms for Explicit FDTD Schemes:

To design the parallel algorithm for the 3D acoustic wave equation based on the FDTD method the most straightforward approach may be the data parallel approach in which the portions of complete mesh are distributed among the parallel workers such that each worker computes the solution on its own data and finally the results are collected to every worker. In addition to distribution of the sub domains the dependencies are treated. For that the message passing schemes are required to share the boundaries of distributed domain among the parallel workers. Therefore, the distribution and message passing schemes related to the FDTD discretization of the domain are formulated properly.

Let $(l+1)$ mesh data points lie on any axis and required to distribute among P workers then $\lfloor (l+1)/P \rfloor = l_2$ points are assigned to all workers equally. But if $r = mod((l+1), P)$ then $r < P$ data points are remaining behind. Therefore, we need to find the maximum possible number of workers that may distribute r data points equally. Assume that there are w_{max}workers are going to utilize this data equally then w_{max} must satisfy the modular expression $mod(r, w_{max}) = 0$, where $1 \le\ w_{max} \le P-1$. Once, w_{max} is found the one can easily distribute rest of the data points to w_{max}workers equally. This states that first w_{max}workers will have $l_2 + 1$ data points and other $P - w_{max}$workers will have $l_2$data points. This distribution scheme is generalized for each individual worker and described by the global indices of the distribution data as follows:

$$\begin{cases} 1+(p-1)(l_2+1) \le i \le l_2+1+(p-1)(l_2+1), & when\ p \le w_{\max} \\ w_{\max}+1+(p-1)l_2 \le i \le w_{\max}+l_2+(p-1), & p > w_{\max} \end{cases} \tag{12}$$

where *p* is the index of worker and *i* is the global index of mesh points along the partitioned axis.

It is also attempted to formulate schemes for sharing the neighboring boundaries among the parallel workers. Assume that Φ is a mapping for sending the data from p^{th} worker to its left $(p-1)^{th}$ or $(p+1)^{th}$ worker that sends the neighborhood boundary $V(x_{nbhd}, y, z)$, x_{nbhd} may be the first or the last boundary of p^{th} worker. It can be expressed mathematically as follows:

$$\Phi(p, p-1) = V(x_{nbhd}, y, z)\,, \tag{13}$$

$$\Phi(p, p-1) = V(x_{nbhd}, y, z)\,, \tag{14}$$

Eq. (13) and Eq. (14) are mappings for sending data to the left and the right workers respectively. The general for of these schemes is given below;

$$\begin{cases} \Phi(p, p+1) = V(l_2+1, y, z), if\ 1 < p \le w_{\max} \\ \Phi(p, p+1) = V(l_2, y, z), if\ w_{\max} < p < P \\ \Phi(p, p-1) = V(1, y, z),\ if\ 1 < p \le P \end{cases} , \tag{15}$$

Similarly, assume that Ψ is the mapping for receiving the data by p^{th} worker from left $(p-1)^{th}$ or right $(p+1)^{th}$ worker that receives the neighborhood boundary$V(x_{nbhd}, y, z)$. It can be represented mathematically as given below:

$$\Psi(p-1, p) = V(x_{nbhd}, y, z)\,, \tag{16}$$

$$\Psi(p-1, p) = V(x_{nbhd}, y, z)\,, \tag{17}$$

Eq. (16) and Eq. (17) are mappings for receiving data from the left and the right workers respectively. The general for of these schemes is given as follows,

$$\begin{cases} \Psi(p+1, p) = V(1, y, z), if\ 1 \le p < P \\ \Psi(p-1, p) = V(l_1+1, y, z), if\ 2 \le p \le w_{\max} \\ \Psi(p-1, p) = V(l_1, y, z), if\ w_{\max} < p \le P \end{cases} , \tag{18}$$

After defining the distribution and message passing schemes the next step is to solve the acoustic equation on each worker *W* and set the error tolerance on local worker and then start the iterations until the solution converges to the predefined error.

6. Parallel Implementation of the Designed Parallel Algorithms For 3D Acoustic Wave Equation:

After designing the parallel algorithms for the explicit FDTD methods implemented on shared memory systems by writing the user defined codes on MATLAB parallel computing interfaces.
The parallel computing system was configured to maximum 16 workers on local scheduler or job manager. A typical view of such parallel shared memory parallel system of MATLAB is exhibited as depicted in Figure 4.

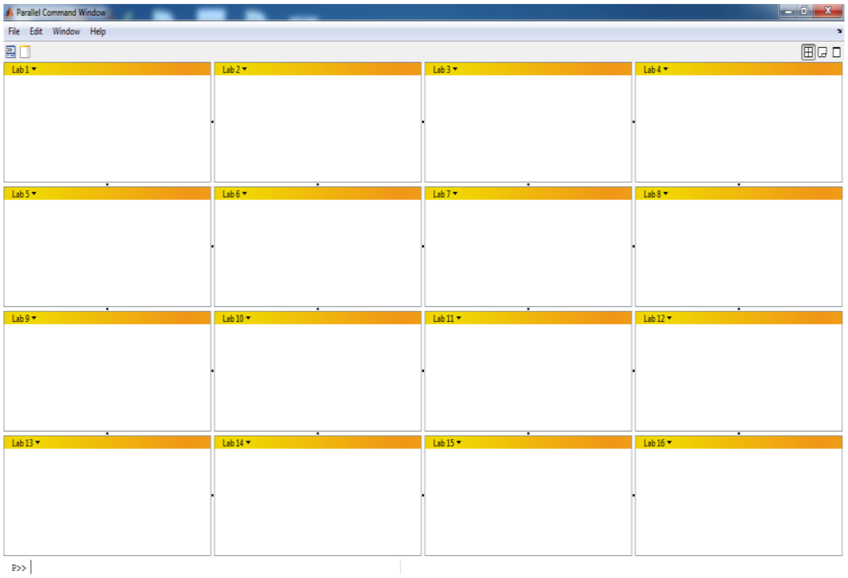


Figure 4: A typical view of 16 parallel workers where the parallel algorithm is running

7. Performance Evaluation of Parallel Algorithm

The performance of the designed algorithm is evaluated at different grid sizes and time step where the speedup and efficiency of the algorithms is analyzed. The performance metrics are defined as:
$t_s\ (sec)$ is the sequential time taken to solve the problem on single worker, $t_p\ (sec)$ is the parallel time taken to solve the problem on P parallel workers, $t_c\ (sec)$ is the communication time taken to solve the problem on P parallel workers, $S_P = t_s/t_p$ is the parallel speedup achieved on P parallel workers, $E_P = S_P/P$ is the parallel efficiency achieved on P parallel workers. The designed algorithm was run on different workers at different grid sizes and improvement in the solution time was analyzed, which will be discussed in the next section.
The performance of the parallel algorithm for the parallel solution of 3D acoustic wave equation are evaluated and discussed. The significant implementation results are obtained by testing the algorithm on different mesh sizes and the sequential time, parallel computing time, communication time, speedup and efficiency are recorded for each run with respect to different number of parallel workers as shown in Tables (1-15).

Table 1: Parallel performance of the algorithm for $Gr = 1\ dt = 0.2$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	0.07000	0.00100	0.08000	1.14286	0.57143
$P = 4$	0.21000	0.14000	0.08000	0.38095	0.09524
$P = 8$	2.93000	2.37000	0.08000	0.02730	0.00341

$P = 12$	4.70000	3.67000	0.08000	0.01702	0.00142
$P = 16$	7.16000	5.44000	0.08000	0.01117	0.00070

Table 2: Parallel performance of the algorithm for $Gr = 2\ dt = 0.2$ on different parallel workers

Number of Workers, P	**Parallel time, t_p (sec)**	**Communication time, t_c (sec)**	**Sequential time, t_s (sec)**	**Speedup $S_p = \frac{t_s}{t_p}$**	**Efficiency $E_p = \frac{S_p}{P}$**
$\boldsymbol{P = 2}$	0.38000	0.02000	0.79000	2.07895	1.03947
$\boldsymbol{P = 4}$	0.43000	0.15000	0.79000	1.83721	0.45930
$\boldsymbol{P = 8}$	2.10000	1.81000	0.79000	0.37619	0.04702
$\boldsymbol{P = 12}$	4.59000	3.99000	0.79000	0.17211	0.01434
$\boldsymbol{P = 16}$	9.25000	7.60000	0.79000	0.08541	0.00534

Table 3: Parallel performance of the algorithm for $Gr = 3\ dt = 0.2$ on different parallel workers

Number of Workers, P	**Parallel time, t_p (sec)**	**Communication time, t_c (sec)**	**Sequential time, t_s (sec)**	**Speedup $S_p = \frac{t_s}{t_p}$**	**Efficiency $E_p = \frac{S_p}{P}$**
$\boldsymbol{P = 2}$	1.12000	0.06000	2.80000	2.50000	1.25000
$\boldsymbol{P = 4}$	1.15000	0.23000	2.80000	2.43478	0.60870
$\boldsymbol{P = 8}$	3.31000	1.55000	2.80000	0.84592	0.10574
$\boldsymbol{P = 12}$	5.50000	3.78000	2.80000	0.50909	0.04242
$\boldsymbol{P = 16}$	11.14000	9.64000	2.80000	0.25135	0.01571

Table 4: Parallel performance of the algorithm for $Gr = 4\ dt = 0.2$ on different parallel workers

Number of Workers, P	**Parallel time, t_p (sec)**	**Communication time, t_c (sec)**	**Sequential time, t_s (sec)**	**Speedup $S_p = \frac{t_s}{t_p}$**	**Efficiency $E_p = \frac{S_p}{P}$**
$\boldsymbol{P = 2}$	2.85000	0.07000	6.95000	2.43860	1.21930
$\boldsymbol{P = 4}$	2.68000	0.34000	6.95000	2.59328	0.64832
$\boldsymbol{P = 8}$	8.62000	5.51000	6.95000	0.80626	0.10078
$\boldsymbol{P = 12}$	11.68000	6.72000	6.95000	0.59503	0.04959
$\boldsymbol{P = 16}$	13.58000	8.95000	6.95000	0.51178	0.03199

Table 5: Parallel performance of the algorithm for $Gr = 5\ dt = 0.2$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	6.10000	0.12000	14.50000	2.37705	1.18852
$P = 4$	5.17000	0.34000	14.50000	2.80464	0.70116
$P = 8$	34.82000	12.02000	14.50000	0.41643	0.05205
$P = 12$	28.39000	13.77000	14.50000	0.51074	0.04256
$P = 16$	186.34000	156.83000	14.50000	0.07781	0.00486

Table 6: Parallel performance of the algorithm for $Gr = 1\ dt = 0.1$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	0.20000	0.02000	0.28000	1.40000	0.70000
$P = 4$	0.48000	0.21000	0.28000	0.58333	0.14583
$P = 8$	4.81000	3.67000	0.28000	0.05821	0.00728
$P = 12$	14.36000	8.25000	0.28000	0.01950	0.00162
$P = 16$	23.34000	16.06000	0.28000	0.01200	0.00075

Table 7: Parallel performance of the algorithm for $Gr = 2\ dt = 0.1$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	1.22000	0.09000	2.81000	2.30328	1.15164
$P = 4$	1.28000	0.26000	2.81000	2.19531	0.54883
$P = 8$	6.77000	5.53000	2.81000	0.41507	0.05188
$P = 12$	9.84000	7.78000	2.81000	0.28557	0.02380
$P = 16$	15.09000	11.88000	2.81000	0.18622	0.01164

Table 8: Parallel performance of the algorithm for $Gr = 3\ dt = 0.1$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	4.56000	0.31000	10.60000	2.32456	1.16228
$P = 4$	4.45000	0.51000	10.60000	2.38202	0.59551
$P = 8$	7.84000	3.51000	10.60000	1.35204	0.16901

$P = 12$	21.18000	14.96000	10.60000	0.50047	0.04171
$P = 16$	56.55000	48.48000	10.60000	0.18744	0.01172

Table 9: Parallel performance of the algorithm for $Gr = 4$ $dt = 0.1$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	11.16000	0.44000	26.17000	2.34498	1.17249
$P = 4$	11.10000	1.39000	26.17000	2.35766	0.58941
$P = 8$	45.24000	21.76000	26.17000	0.57847	0.07231
$P = 12$	56.52000	39.36000	26.17000	0.46302	0.03859
$P = 16$	88.34000	75.34000	26.17000	0.29624	0.01852

Table 10: Parallel performance of the algorithm for $Gr = 5$ $dt = 0.1$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	23.06000	0.74000	52.06000	2.25759	1.12879
$P = 4$	21.02000	1.66000	52.06000	2.47669	0.61917
$P = 8$	55.28000	26.54000	52.06000	0.94175	0.11772
$P = 12$	182.19000	67.46000	52.06000	0.28575	0.02381
$P = 16$	414.87000	345.00000	52.06000	0.12549	0.00784

Table 11: Parallel performance of the algorithm for $Gr = 1$ $dt = 0.067$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$P = 2$	0.43000	0.03000	0.60000	1.39535	0.69767
$P = 4$	0.36000	0.06000	0.60000	1.66667	0.41667
$P = 8$	6.26000	5.32000	0.60000	0.09585	0.01198
$P = 12$	14.48000	10.41000	0.60000	0.04144	0.00345

$\boldsymbol{P = 16}$	16.39000	12.64000	0.60000	0.03661	0.00229

Table 12: Parallel performance of the algorithm for $Gr = 2$ $dt = 0.067$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$\boldsymbol{P = 2}$	2.79000	0.15000	6.45000	2.31183	1.15591
$\boldsymbol{P = 4}$	2.80000	0.56000	6.45000	2.30357	0.57589
$\boldsymbol{P = 8}$	9.85000	6.90000	6.45000	0.65482	0.08185
$\boldsymbol{P = 12}$	21.62000	17.65000	6.45000	0.29833	0.02486
$\boldsymbol{P = 16}$	20.93000	15.17000	6.45000	0.30817	0.01926

Table 13: Parallel performance of the algorithm for $Gr = 3$ $dt = 0.067$on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
$\boldsymbol{P = 2}$	10.99000	0.45000	23.55000	2.14286	1.07143
$\boldsymbol{P = 4}$	9.46000	0.75000	23.55000	2.48943	0.62236
$\boldsymbol{P = 8}$	20.87000	13.17000	23.55000	1.12841	0.14105
$\boldsymbol{P = 12}$	30.42000	17.45000	23.55000	0.77416	0.06451
$\boldsymbol{P = 16}$	67.49000	55.87000	23.55000	0.34894	0.02181

Table 14: Parallel performance of the algorithm for $Gr = 4$ $dt = 0.067$on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
P=2	25.48000	1.10000	57.67000	2.26334	1.13167
P=4	22.76000	1.57000	57.67000	2.53383	0.63346
P=8	70.50000	23.22000	57.67000	0.81801	0.10225
P=12	100.13000	39.08000	57.67000	0.57595	0.04800
P=16	222.47000	153.27000	57.67000	0.25923	0.01620

Table 15: Parallel performance of the algorithm for $Gr = 5$ $dt = 0.067$ on different parallel workers

Number of Workers, P	Parallel time, t_p (sec)	Communication time, t_c (sec)	Sequential time, t_s (sec)	Speedup $S_p = \frac{t_s}{t_p}$	Efficiency $E_p = \frac{S_p}{P}$
P=2	51.02000	2.37000	113.79000	2.23030	1.11515
P=4	58.51000	3.52000	113.79000	1.94480	0.48620
P=8	126.21000	58.94000	113.79000	0.90159	0.11270
P=12	255.17000	195.67000	113.79000	0.44594	0.03716
P=16	356.19000	250.71000	113.79000	0.63219	0.04129

In the following sections the obtained results are visualized and interpreted.

8. Analysis of Parallel Time:

To see the reduction in computing time the simulation of 3D acoustic wave equation the designed algorithm is run on $2, 4, 8, 12$ and 16 parallel workers independently. While the grid is refined by a factor of $1, 2, 3, 4$ and 5 that is $20 \times 20 \times 20, 40 \times 40 \times 40, \ldots, 100 \times 100 \times 100$ mesh size respectively. The parallel computing time are noted for the $5, 10,$ and 15 time steps as shown in the following Figures 5 (a-c). It is found that the parallel time is reduced on $P = 2$ and $P = 4$ for all three cases but for higher number of workers and more refined mesh the parallel time is increased. However, for the small mesh size the $P = 8$ workers are also suitable. But more than $P = 8$ workers are not suitable due to the communication time. Also, the parallel computing time increases approximately exponentially.

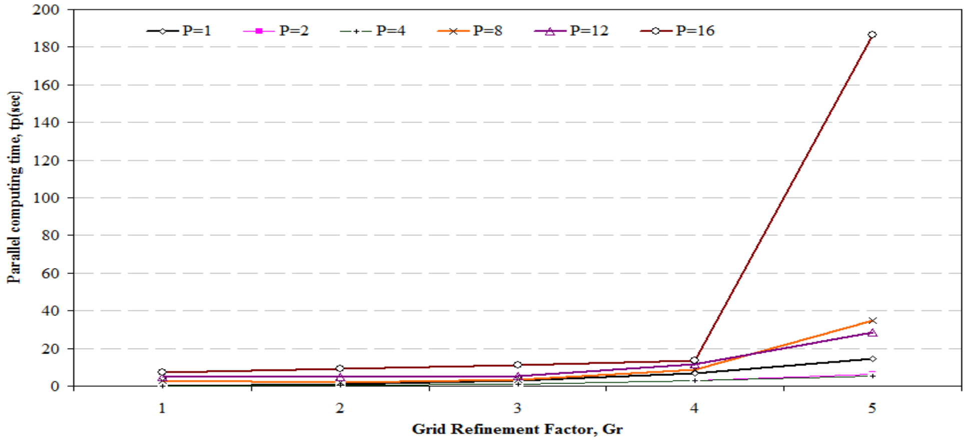


Figure 5(a): Parallel computing time at different grid refinements Gr and time step $dt = 0.2$.

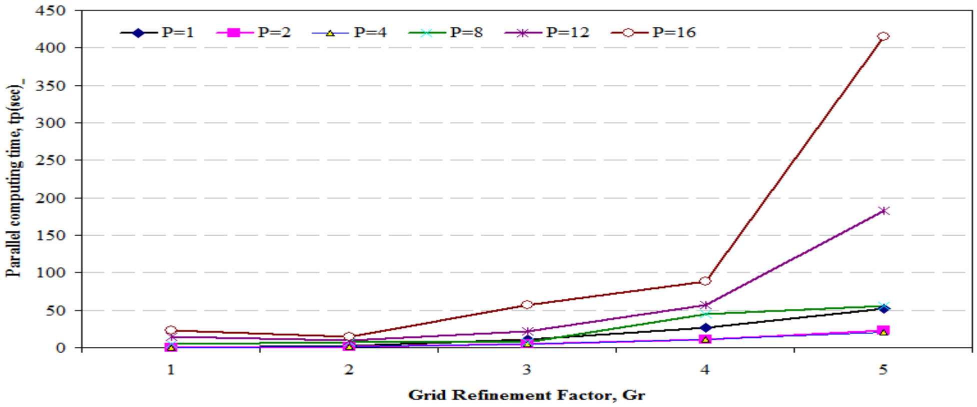


Figure 5(b): Parallel computing time at different grid refinements Gr and time step $dt = 0.1$.

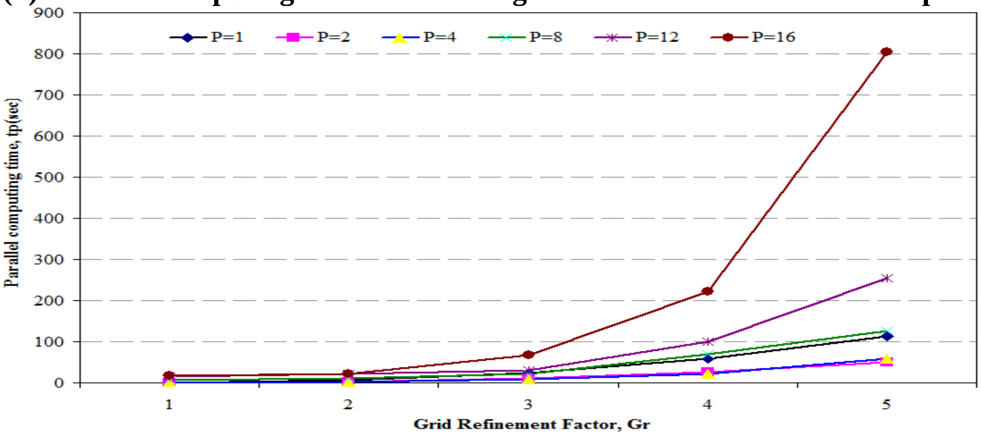
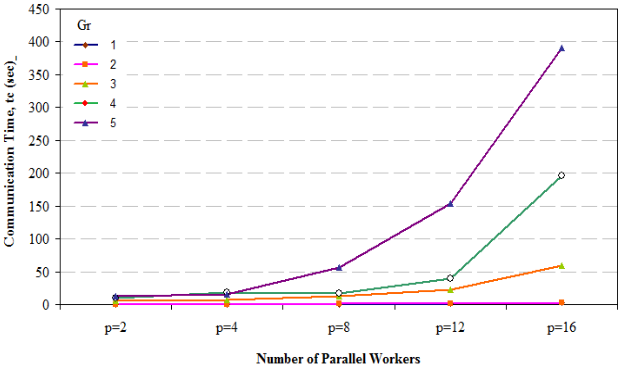


Figure 5(c): Parallel computing time at different grid refinements Gr and time step $dt = 0.067$



9. Analysis of Communication Time:

Similarly, the communication time with respect to different number of workers and grid refinement factor has been analyzed and shown in the following Figure 6. The behavior of communication time is also exponentially increasing as the data size increases to be shared among the processors.

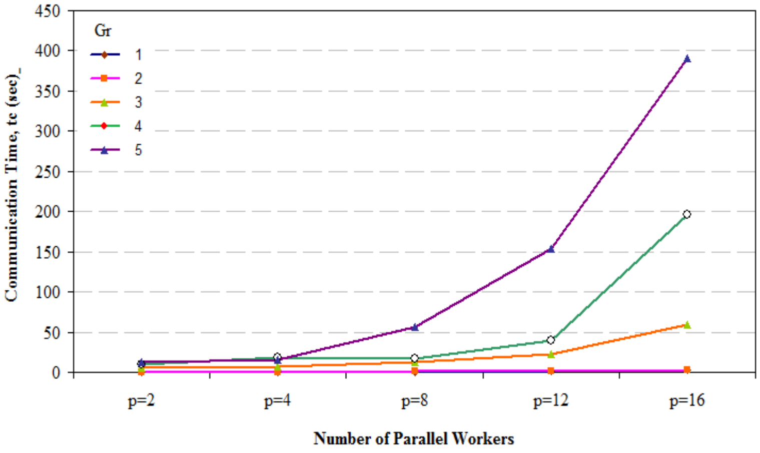


Figure 6: Communication time, t_c at different grid refinements, Gr and at time step $dt = 0.067$.

10. Analysis of Parallel Speed Up:

Speed up is an important performance metric that is mainly used to see how fast the parallel algorithm works as compared to sequential algorithm. The speed up of the designed parallel algorithm for 3D acoustic wave equation is analyzed for the different time steps. By increasing the time steps the number of mesh points increases. For three different values of dt the speed up is determined as shown in the following Figures 7 (a-c). For $dt = 0.2$ the maximum speed up is achieved for $Gr = 5$ on $P = 4$ workers and the maximum speed up is obtained at $Gr = 3$ on $P = 2$. Similarly, for $dt = 0.1$ the maximum speed up is achieved for $Gr = 5$ on $P = 4$ workers, the maximum speed up is obtained at $Gr = 2$ on $P = 2$ and maximum speed up is observed at $Gr = 3$ on $P = 8$. Finally, for $dt = 0.067$ the maximum speed up is achieved for $Gr = 5$ on $P = 4$ workers, the maximum speed up is obtained at $Gr = 2$ on $P = 2$ and maximum speed up is observed at $Gr = 3$ on $P = 8$. It is concluded that among all the workers $P = 4$ workers is the best choice for large mesh sizes.

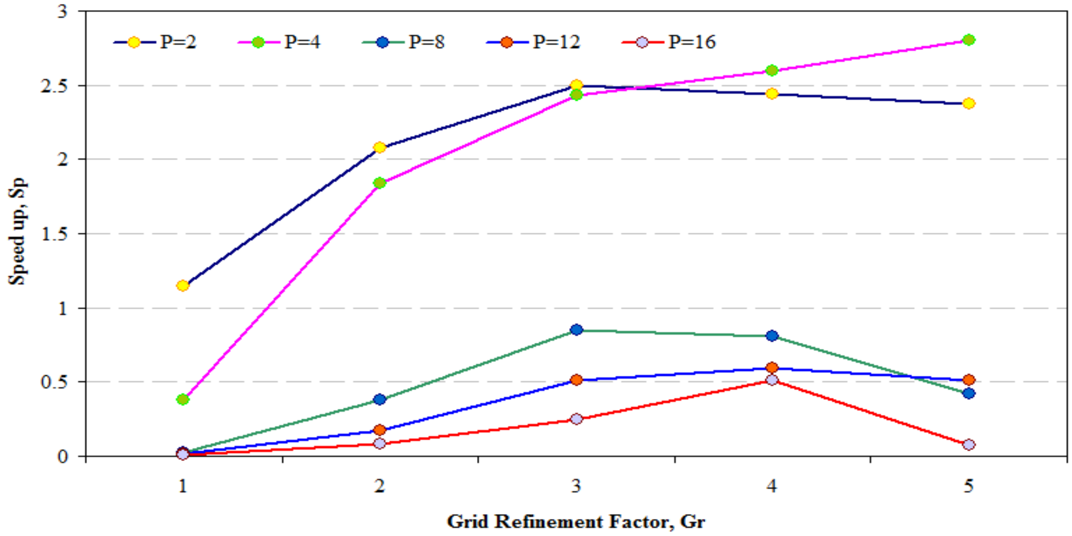


Figure 7(a): Speed up at different grid refinements, Gr and time step $dt = 0.2$.

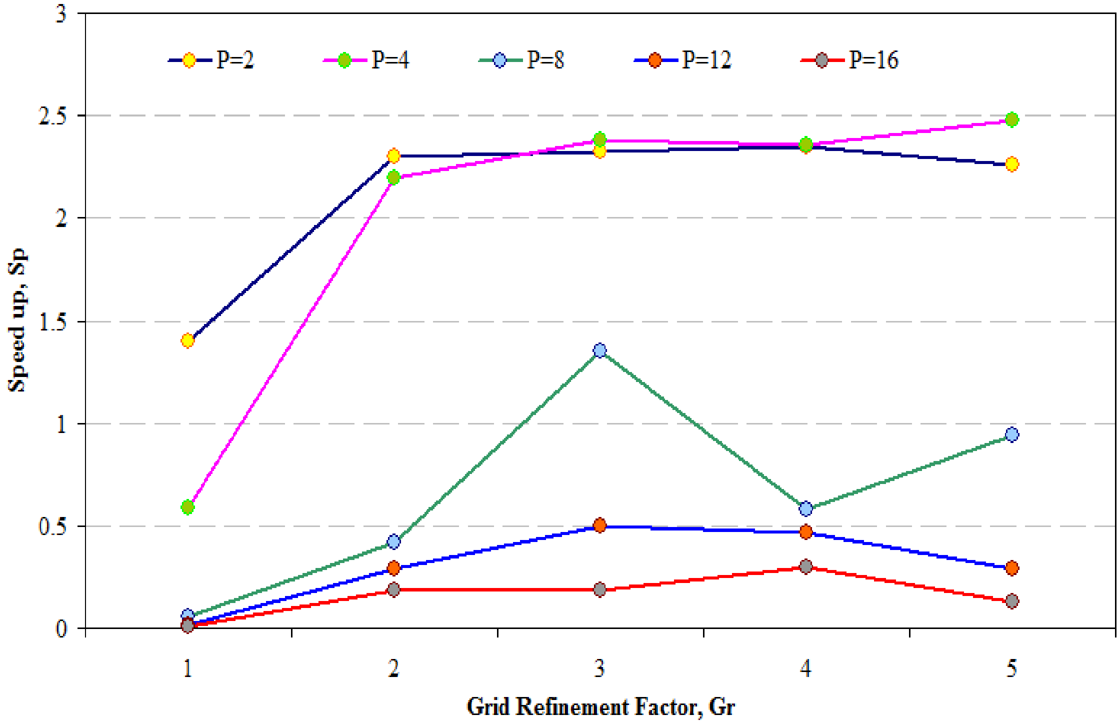


Figure 7(b): Speed up at different grid refinements, Gr and time step $dt = 0.1$.

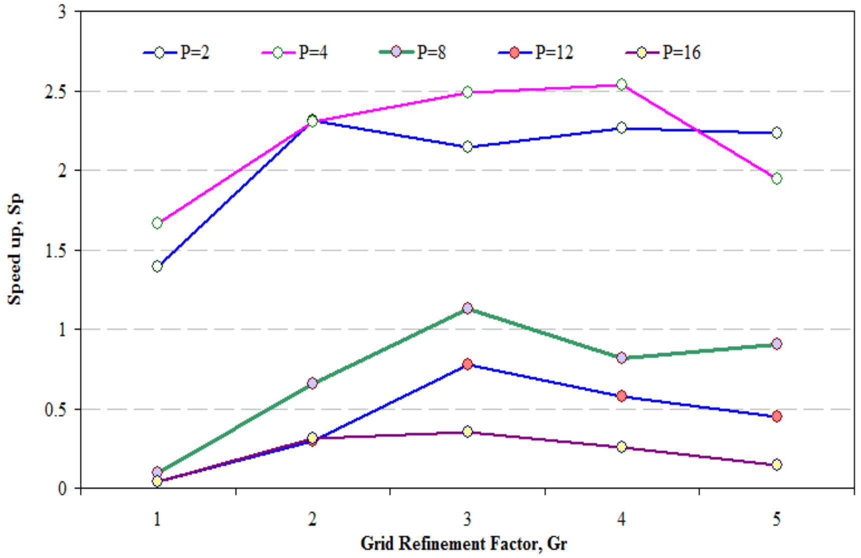


Figure 7(c): Speed up at different grid refinements, Gr and time step $dt = 0.067$.

11. Analysis of Parallel Efficiency:

Parallel efficiency is also an important performance metric that is mainly used to see how efficiently the parallel algorithm works as compared to sequential algorithm. The efficiency of the designed parallel algorithm for 3D acoustic wave equation is analyzed for the different time steps. For three different values of dt the speed up is determined as shown in the following Figures 8 (a-c). For all cases $dt = 0.2$ the maximum efficiency is achieved for $Gr = 3$ on $P = 2$ workers and the maximum efficiency is obtained at $Gr = 3$ on $P = 2$ then for $Gr = 5$ on $P = 4$. Similarly, for $dt = 0.1$ the maximum efficiency is achieved for $Gr = 5$ on $P = 4$ workers, the maximum speed up is obtained at $Gr = 2$ on $P = 2$ and maximum speed up is observed at $Gr = 3$ on $P = 8$. Finally, for $dt = 0.067$ the maximum efficiency is achieved for $Gr = 2$ on $P = 2$ workers, the maximum speed up is obtained at $Gr = 2$ on $P = 2$ It is concluded that among all the workers $P = 2$ and $P = 4$ workers are the best choice for large mesh sizes.

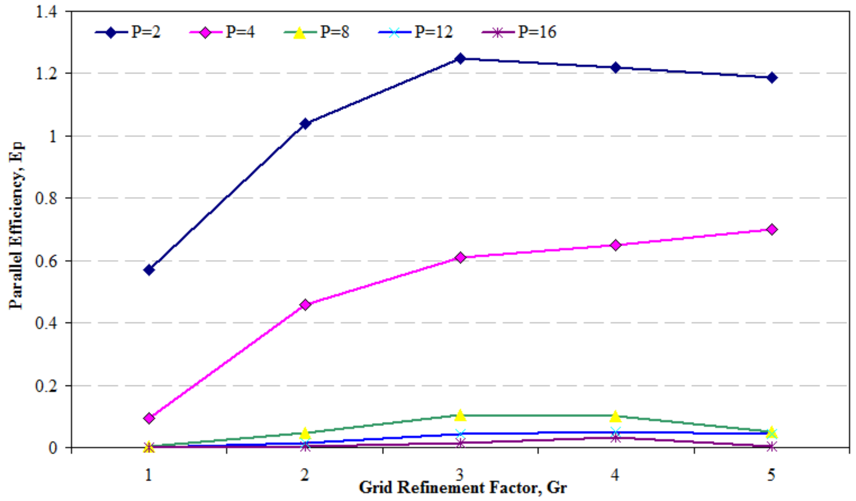


Figure 8(a): Parallel efficiency at different grid refinements, Gr and time step $dt = 0.2$.

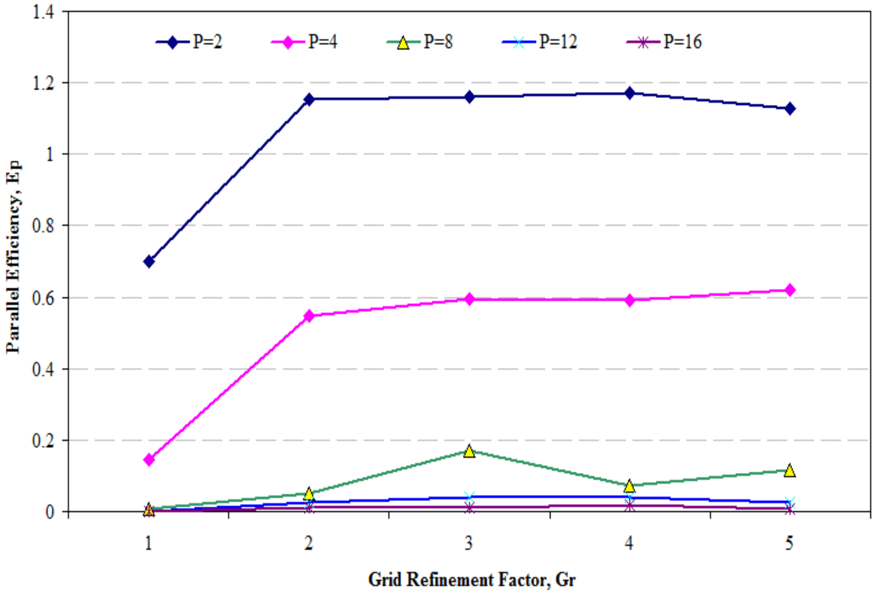


Figure 8(b): Parallel efficiency at different grid refinements, Gr and time step $dt = 0.1$.

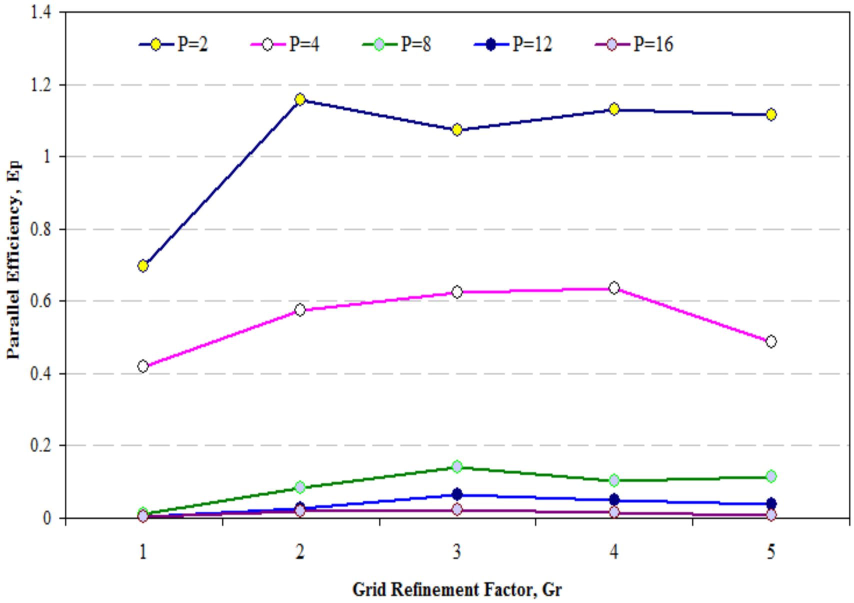


Figure 8(c): Parallel efficiency at different grid refinements, Gr and time step $dt = 0.067$.

12. Analysis of Message Passing Profiles:

In this study the message passing interface (MPI) of MATLAB is used for parallel implementation of the designed algorithm. Therefore, the MPI profiles of each run of the program are also obtained to see the communication patterns among the workers. Few samples of MPI profiles of the parallel algorithm for $P = 2$, and $P = 4$ are shown by the Figure 9 and Figure 10 respectively. These profiles are useful to know about the amount of data exchanged among the workers, communication time among the workers and communication waiting time to synchronize the computations. In the figures a pair wise comparison between the workers is provided the values of the data transfer, communication time and communication waiting time labeled by a color bar on the right of the figures.

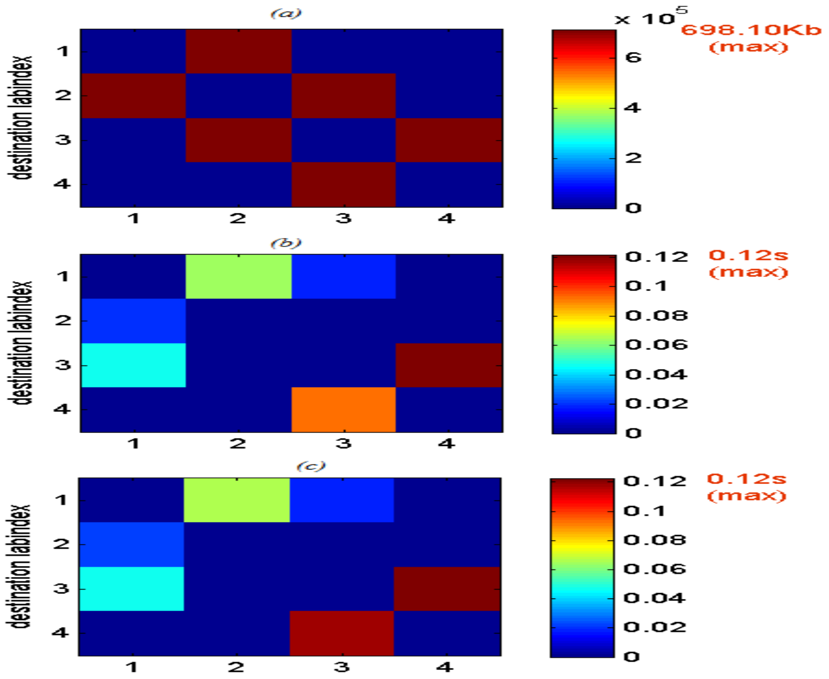


Figure 9: Message passing profiles using 4(four) parallel workers at $Gr = 3$ and at $dt = 0.067$, (a) Maximum amount of data exchanged among workers, (b) Maximum communication time per worker, (c) Maximum communication waiting time per worker.

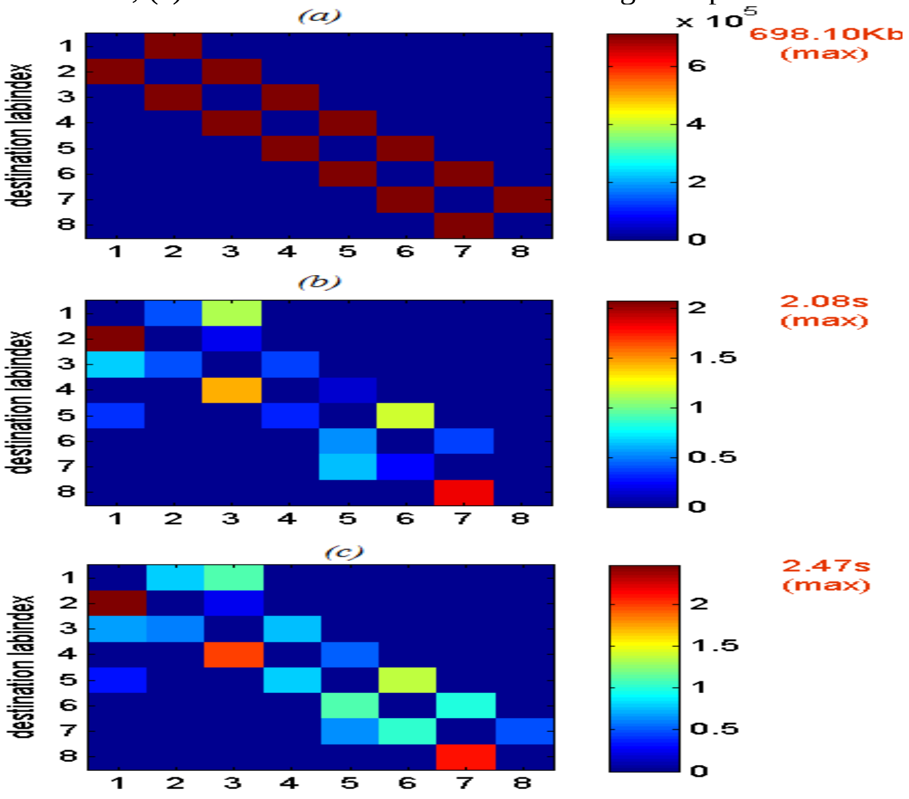


Figure 10: Message passing profiles using 8(eight) parallel workers at $Gr = 3$ and at $dt = 0.067$,

(a) Maximum amount of data exchanged among workers, (b) Maximum communication time per worker, (c) Maximum communication waiting time per worker.

Conclusion:

This study developed a parallel algorithm for the numerical simulation of the 3D acoustic wave equation using the FDTD method. The algorithm was implemented on a shared memory parallel system utilizing MATLAB's parallel computing capabilities. A 3D rectangular domain was selected, defined by specified initial and boundary conditions. The algorithm employed a data-parallel approach, with data distribution and message-passing schemes designed to align with the discretization scheme. The proposed parallel algorithm demonstrated significant efficiency, achieving up to a 3-fold reduction in computational time compared to its sequential counterpart. Optimal performance was observed when using 2–4 workers, but efficiency declined with more workers due to increased communication overhead. Additionally, communication time increased exponentially with finer grid refinement or a larger number of workers, posing limitations to scalability. These findings underscore the potential of parallel computing for enhancing the efficiency of large-scale numerical simulations while highlighting the need to balance computational and communication costs.

References

Allen, C. M., Eldek, A. A., Elsherbeni, A. Z., Smith, C. E., Huang, C. P., & Lee, K. (2005). Applications Pr E oo f Pr E oo f. *Robotics*, *53*(7), 3–7.

Bernacki, M., Fezoui, L., Lanteri, S., & Piperno, S. (2006). Parallel discontinuous Galerkin unstructured mesh solvers for the calculation of three-dimensional wave propagation problems. *Applied Mathematical Modelling*, *30*(8), 744–763. https://doi.org/10.1016/j.apm.2005.06.015

Bernacki, M., Lanteri, S., & Piperno, S. (2006). Time-domain parallel simulation of heterogeneous wave propagation on unstructured grids using explicit, nondiffusive, Discontinuous Galerkin methods. *Journal of Computational Acoustics*, *14*(1), 57–81. https://doi.org/10.1142/S0218396X06002937

Cai, X. H., Liu, Y., Ren, Z. M., Wang, J. M., Chen, Z. De, Chen, K. Y., & Wang, C. (2015). Three-dimensional acoustic wave equation modeling based on the optimal finite-difference scheme. *Applied Geophysics*, *12*(3), 409–420. https://doi.org/10.1007/s11770-015-0496-y

Chang, S. L., & Liu, Y. (2013). A truncated implicit high-order finite-difference scheme combined with boundary conditions. *Applied Geophysics*, *10*(1), 53–62. https://doi.org/10.1007/s11770-012-0342-4

Frances, J., Otero, B., Bleda, S., Gallego, S., Neippa, C., Marqueza, A., & Belendez, A. (2015). Multi-GPU and multi-CPU accelerated FDTD scheme for vibroacoustic applications. *Computer Physics Communications*, *191*(1), 43–51. https://doi.org/10.1016/j.cpc.2015.01.017

Garcia, J. R. S. (2009). *3D finite-difference time-domain modeling of acoustic wave propagation based on domain decompostion*.

Kamboh, S. A., Labadin, J., Amur, K. B., & Soomro, M. A. (2015). *Parallel Numerical Solution of Linear PDEs Using Implicit and Explicit Finite Difference Methods*. *8*(1), 98–104.

Khokhar, R. B., Bhutto, A. A., Siddiqui, N. F., Shaikh, F., & Bhutto, I. A. (2023). Numerical analysis of flow rates, porous media, and Reynolds numbers affecting the combining and separating of Newtonian fluid flows. *VFAST Transactions on Mathematics*, *11*(1), 217–236.

Kowalczyk, K., & Van Walstijn, M. (2011). Room acoustics simulation using 3-D compact explicit FDTD schemes. *IEEE Transactions on Audio, Speech and Language Processing*, *19*(1), 34–46. https://doi.org/10.1109/TASL.2010.2045179

Li, Y., Meyer, J., Lokki, T., Cuenca, J., Atak, O., & Desmet, W. (2022). Benchmarking of finite-difference time-domain method and fast multipole boundary element method for room acoustics. *Applied Acoustics*, *191*, 1–25. https://doi.org/10.1016/j.apacoust.2022.108662

LIU, S.-W., WANG, H.-Z., CHEN, S.-C., & KONG, X.-N. (2013). Implementation strategy of 3D reverse time migration on GPU/CPU clusters. *Chinese Journal of Geophysics*, *56*(10), 3487–3496.

Liu, Y., & Sen, M. K. (2009). A new time-space domain high-order finite-difference method for the acoustic wave equation. *Journal of Computational Physics*, *228*(23), 8779–8806. https://doi.org/10.1016/j.jcp.2009.08.027

Mehra, R., Raghuvanshi, N., Savioja, L., Lin, M. C., & Manocha, D. (2012). An efficient GPU-based time domain solver for the acoustic wave equation. *Applied Acoustics*, *73*(2), 83–94. https://doi.org/10.1016/j.apacoust.2011.05.012

Morales, N., Chavda, V., Mehra, R., & Manocha, D. (2017). MPARD: A high-frequency wave-based acoustic solver for very large compute clusters. *Applied Acoustics*, *121*, 82–94. https://doi.org/10.1016/j.apacoust.2017.01.009

Näsholm, S. P., & Holm, S. (2013). On a fractional Zener elastic wave equation. *Fractional Calculus and Applied Analysis*, *16*(1), 26–50. https://doi.org/10.2478/s13540-013-0003-1

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J. Y., Giraud, L., & Ali, H. B. H. (2007). 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. *Geophysics*, *72*(5). https://doi.org/10.1190/1.2759835

Otero, B., Rojas, O., Moya, F., & Castillo, J. E. (2020). Alternating direction implicit time integrations for finite difference acoustic wave propagation: Parallelization and convergence. *Computers and*

Fluids, *205*. https://doi.org/10.1016/j.compfluid.2020.104584

Robertsson, J. O., Blanch, J. O., Nihei, K., & Tromp, J. (2012). Numerical modeling of seismic wave propagation: Gridded two-way wave-equation methods. *SEG Geophysics Reprint Series*, *28*(28). http://library.seg.org/doi/pdf/10.1190/1.9781560803089%5Cnhttp://swlgs.seg.org/documents/10161/74355/204A.pdf

Saarelma, J. (2013). *Finite-difference time-domain solver for room acoustics using graphics process-ing units*.

Savioja, L. (2010). Real-time 3D finite-difference time-domain simulation of low- and mid-frequency room acoustics. *Proceedings of the International Conference on Digital Audio Effects, DAFx*, 1–8.

Sheaffer, J. D., Fazenda, B. M., Angus, J. A. S., & Murphy, D. T. (2011). A simple multiband approach for solving frequency dependent problems in numerical time domain methods. *Proceedings of Forum Acusticum*, *c*, 269–274.

Sheaffer, J., & Fazenda, B. (2010). FDTD / K-DWM SIMULATION OF 3D ROOM ACOUSTICS ON GENERAL PURPOSE GRAPHICS HARDWARE USING COMPUTE UNIFIED DEVICE ARCHITECTURE (CUDA) Propagation of Sound Spatio-Temporal Discretisation. *Proceedings of the Institute of Acoustics*, *32*(5).

Sheu, Y.-L., & Li, P.-C. (2008). Simulations of photoacoustic wave propagation using a finite-difference time-domain method with Berenger's perfectly matched layers. *The Journal of the Acoustical Society of America*, *124*(6), 3471–3480. https://doi.org/10.1121/1.3003087

Sheua, Y.-L., & Lia, P.-C. (n.d.). Photoacoustic Wave Propagation Simulations Using the Finite-Difference Time-Domain Method with Berenger's Perfectly Matched Layers. *J. Acoust. Soc. Am*, *124*(6), 20008.

Shragge, J. (2014). Solving the 3D acoustic wave equation on generalized structured meshes: A finite-difference time-domain approach. *Geophysics*, *79*(6), T363–T378. https://doi.org/10.1190/geo2014-0172.1

Vaccari, A., Cala' Lesina, A., Cristoforetti, L., & Pontalti, R. (2011). Parallel implementation of a 3D subgridding FDTD algorithm for large simulations. *Progress in Electromagnetics Research*, *120*(August), 263–292. https://doi.org/10.2528/PIER11063004

Wang, S., de Hoop, M. V., Xia, J., & Li, X. S. (2012). Massively parallel structured multifrontal solver for time-harmonic elastic waves in 3-D anisotropic media. *Geophysical Journal International*, *191*(1), 346–366. https://doi.org/10.1111/j.1365-246X.2012.05634.x

Wang, Y., Liang, W., Nashed, Z., Li, X., Liang, G., & Yang, C. (2014). Seismic modeling by optimizing regularized staggered-grid finite-difference operators using a time-space-domain dispersion-relationship-preserving method. *Geophysics*, *79*(5), T277–T285. https://doi.org/10.1190/geo2014-0078.1

Yoon, K., Shin, C., Sangyong, S. U. H., Lines, L. R., & Hong, S. (2003). 3D reverse-time migration using the acoustic wave equation: An experience with the SEG/EAGE data set. *Leading Edge (Tulsa, OK)*, *22*(1), 38–41. https://doi.org/10.1190/1.1542754

Yu, W., Liu, Y., Su, T., Hunag, N. T., & Mittra, R. (2005). A robust parallel conformal finite-difference time-domain processing package using the MPI library. *IEEE Antennas and Propagation Magazine*, *47*(3), 39–59. https://doi.org/10.1109/MAP.2005.1532540

Zhang, J. H., & Yao, Z. X. (2012). Optimized finite-difference operator for broadband seismic wave modeling. *Geophysics*, *78*(1), A13–A18. https://doi.org/10.1190/GEO2012-0277.1