International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 6, December 2024, pp. 6519~6525
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i6.pp6519-6525

Parallel numerical simulation of the 2D acoustic wave equation

Arshyn Altybay[1,2], Dauren Darkenbayev[2,3], Nurbapa Mekebayev[4]
[1]Department of Differential Equations, Institute of Mathematical and Mathematical Modeling, Almaty, Kazakhstan
[2]Department of Computer Science, Faculty of Information Technologies, Al-Farabi Kazakh National University, Almaty, Kazakhstan
[3]Department of Computational Sciences and Statistics, Faculty of Mechanics and Mathematics, Al-Farabi Kazakh National University, Almaty, Kazakhstan
[4]Department of Computer Science, Institute of Physics, Mathematics and Computing, Kazakh National Women's Teacher Training University, Almaty, Kazakhstan

Article Info

Article history:

Received May 27, 2024
Revised Jul 12, 2024
Accepted Jul 17, 2024

Keywords:

Computer modelling
Numerical simulation
Parallel computing
Programming language
Technologies

ABSTRACT

Mathematical simulation has significantly broadened with the advancement of parallel computing, particularly in its capacity to comprehend physical phenomena across extensive temporal and spatial dimensions. High-performance parallel computing finds extensive application across diverse domains of technology and science, including the realm of acoustics. This research investigates the numerical modeling and parallel processing of the two-dimensional acoustic wave equation in both uniform and non-uniform media. Our approach employs implicit difference schemes, with the cyclic reduction algorithm used to obtain an approximate solution. We then adapt the sequential algorithm for parallel execution on a graphics processing unit (GPU). Ultimately, our findings demonstrate the effectiveness of the parallel approach in yielding favorable results.

Corresponding Author:

Dauren Darkenbayev
Department of Computer Science, Faculty of Information Technologies, Al-Farabi Kazakh National University
Al-Farabi Avenue 71, Almaty, Kazakhstan
Email: dauren.kadyrovich@gmail.com

1. INTRODUCTION

A significant facet of acoustics involves the simulation of wave patterns. Over the past few years, there has been a trend towards leveraging graphics processing units (GPUs) to enhance parallel computing speeds. Nevertheless, it was not until the introduction of a novel generation of GPUs featuring multi-core architecture that tangible advancements in this area became evident. The aim of this study is to create a parallel version of the finite difference technique for solving the two-dimensional acoustic wave equation utilizing the compute unified device architecture (CUDA) technology on a GPU. Additionally, the efficacy of parallelization will be evaluated through a comparison of the computational time required for solving the 2D wave equation on GPUs versus central processing unit (CPU) [1].

Numerical methods for wave processes are being actively studied, including finite difference, finite volume, elementary and spectral element methods, as well as various time and boundary-domain distributed methods. Each of these approaches offers unique advantages for modeling wave phenomena. However, some of these methods have notable disadvantages, particularly when it comes to transforming quadratic equations before discretization. This article is inspired by the current enthusiasm surrounding advanced compact difference techniques for resolving differential equations. These higher-order compact difference methods offer enhanced resolution on compact mesh stencils. Additionally, the study employs alternating direction implicit (ADI) methods to decompose multidimensional problems into a sequence of one-dimensional problems, ensuring both resilience and effectiveness [2].

GPU technology is used to speed up calculations when processing large meshes. GPUs, with their multi-core architecture and high degree of parallelism, offer significant advantages such as low cost, high throughput, and energy efficiency. We observed a notable boost in performance, several-fold, by leveraging GPUs [3], [4]. Programming for NVIDIA GPUs has become notably more accessible following the introduction of the CUDA programming language in late 2006. This language is relatively straightforward to grasp as its syntax closely resembles that of the C programming language.

With GPUs emerging as a viable substitute for processors in parallel computing, various parallel triangulation solvers and hybrid approaches have been deployed on GPUs [5]–[13]. For instance, Zhang *et al*. [5] initially introduced the parallel cyclic reduction (PCR) method and subsequently suggested a hybrid cyclic reduction-parallel cyclic reduction (CR-PCR) algorithm. The hybrid PCR-Thomas method was also proposed and studied by Souri [13]. The literature provides many examples of the successful use of GPUs for wave propagation modeling [14]–[24].

In this work, we aim to address the numerical implementation of two-dimensional acoustic wave propagation on a GPU, providing insights into the potential advantages and efficiency of this approach. Leveraging the parallel computing capabilities of GPUs can significantly accelerate the computation time, making it feasible to simulate larger and more complex wave fields in a reasonable timeframe. Additionally, this implementation can offer a scalable solution for real-time applications and extensive parameter studies, highlighting the GPU's role as a powerful tool in computational acoustics.

2. GOVERNING EQUATION AND NUMERICAL SIMULATION

We investigate the acoustic wave equation in two dimensions.

$$\frac{\partial^2 H}{\partial t^2} - c^2(x,y)\left(\frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2}\right) = q(t,x,y), (t,x,y)\epsilon[0,T]\times[0,X]\times[0,Y], \quad (1)$$

The initial conditions are given as:

$$H(0,x,y) = H_0(x,y), x, y\epsilon[0,X]\times[0,Y] \quad (2)$$

and

$$\frac{\partial H(0,x,y)}{\partial t} = 0, x, y\epsilon[0,X]\times[0,Y] \quad (3)$$

The boundary conditions are:

$$H(t,x,0) = 0, H(t,x,Y) = 0, t\epsilon[0,T], x\epsilon[0,X]\times[0,Y] \quad (4)$$

and

$$H(t,0,y) = 0, H(t,X,y) = 0, t\epsilon[0,T], y\epsilon[0,X]\times[0,Y] \quad (5)$$

In this context, H represents the wave function, q denotes the source term, and c indicates the wave speed. For the homogeneous scenario, c remains uniform, and a Gaussian function is employed as the source term.

In our numerical modeling approach, we establish a space-time grid with increments h_1, h_2, and τ, respectively, for the variables x, y, and t:

$$\omega^{\tau}_{h_1 h_2} = \left\{x_i = ih_1, i = \overline{0,N}; y_j = jh_2, j = \overline{0,N};\ t^n = n\tau, n = 0,1,2 \ldots \frac{T}{\tau}\right\} \quad (6)$$

Here, we set $h_1 = X/N_1$, $h_2 = Y/N_2$, and $\tau = T/M$. Using this grid, we apply the finite difference technique to approximate the differential equation presented in problem (1) along with the accompanying conditions (2) to (5). For simplicity, let's define N as $N_1 = N_2$ and denote h as $h_1 = h_2$. Now, let's consider the implicit finite difference scheme for addressing the problem outlined in (1)-(5).

$$\frac{H_{i,j}^{n+1}-2H_{i,j}^{n}+H_{i,j}^{n-1}}{\tau^2}-\frac{c_{i,j}}{h^2}\left(H_{i+1,j}^{n+1}-2H_{i,j}^{n+1}+H_{i-1,j}^{n+1}+H_{i,j+1}^{n+1}-2H_{i,j}^{n+1}+H_{i,j-1}^{n+1}\right)=q_{i,j}^{n}, \quad (7)$$

For $(n; i;\ j)\ \epsilon\ \omega_{h_1h_2}^{\tau}$ with initial conditions

$$H_{i,j}^{0}=\varphi_{i,j}, H_{i,j}^{1}-H_{i,j}^{0}=\tau\varphi_{i,j}, \quad (8)$$

For $(i,j)\in\overline{0,N}\times\overline{0,N}$, and with boundary conditions

$$H_{0,j}^{n}=0, H_{N,j}^{n}=0,\ H_{i,0}^{n}=0, H_{i,N}^{n}=0, \quad (9)$$

For $(j,n)\in\overline{0,N}\times\overline{0,M}$ and $(i,n)\in\overline{0,N}\times\overline{0,M}$, respectively. The implicit scheme is inherently stable regardless of the chosen step sizes and achieves an accuracy of $O(\tau+|h^2|)$ as discussed in [25]. The difference (7) is addressed using the implicit alternating direction method (ADI), which involves splitting it into two separate subtasks.

$$\frac{H_{i,j}^{n+1}-2H_{i,j}^{n}+H_{i,j}^{n-1/2}}{\tau^2}-\frac{c_{i,j}}{h^2}\left(H_{i+1,j}^{n+\frac{1}{2}}-2H_{i,j}^{n+\frac{1}{2}}+H_{i-1,j}^{n+\frac{1}{2}}\right)=q_{i,j}^{n}, \quad (10)$$

$$\frac{H_{i,j}^{n+1}-2H_{i,j}^{n+1/2}+H_{i,j}^{n-1/2}}{\tau^2}-\frac{c_{i,j}}{h^2}\left(H_{i,j+1}^{n+1}-2H_{i,j}^{n+1}+H_{i,j-1}^{n+1}\right)=q_{i,j}^{n+1/2}, \quad (11)$$

3. NUMERICAL METHOD

The alternating direction implicit (ADI) method, which utilizes finite difference techniques, has historically been a well-established method for addressing differential equations in complex, high-dimensional settings. Initially proposed by Peaceman and Ratchford [26], the method has seen several refinements over time [27]–[31]. Characterized by implicit finite difference operations, the ADI method ensures full stability in problems devoid of mixed derivatives and maintains a considerable stability margin in cases where mixed derivatives are present [10]. By employing the implicit sub-circuit, we apply the cyclic reduction (CR) method along the x-direction to compute the grid function $H_{i,j}^{k+1/2}$. Subsequently, in the second fractional time step utilizing sub-circuit, the CR method is utilized along the y-axis direction, yielding the grid function $H_{i,j}^{k+1}$. The ADI method demonstrates second-order accuracy of $O(\tau^2+h^2)$. The ensuing numerical simulations are detailed below. All calculations were performed with the Python programming language and employed the cyclic reduction technique. For the simulations, a time step of $\tau=0.001$ and a spatial step of $h=1$ were used. Results were visualized with the Matplotlib library, and the findings from the numerical experiments are illustrated in Figures 1 to 3.

These figures depict both the time function of the source and the propagation of the wave through a medium with varying properties at multiple time points. In such a medium, the speed of the wave fluctuates according to the medium's structural variations. Our analysis focuses on how the wave propagates through a medium with distinct wave velocities in the white and blue regions.

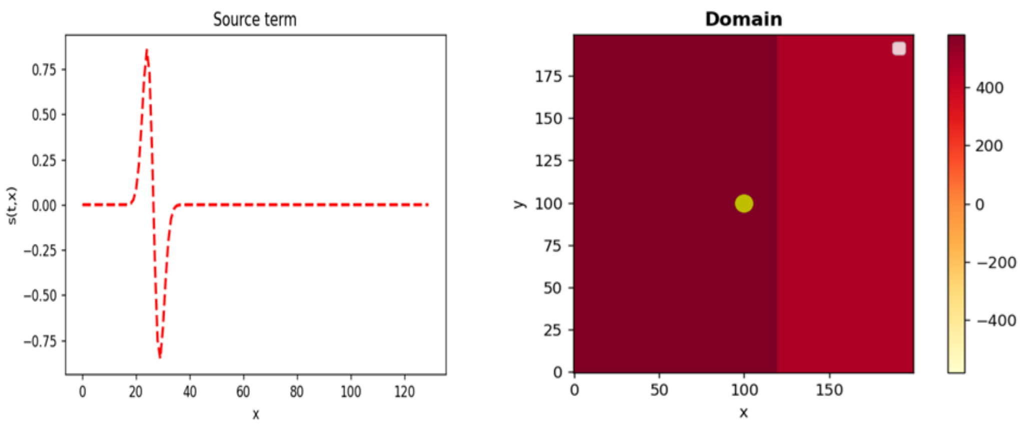

Figure 1. Source function and two-layer heterogeneous medium

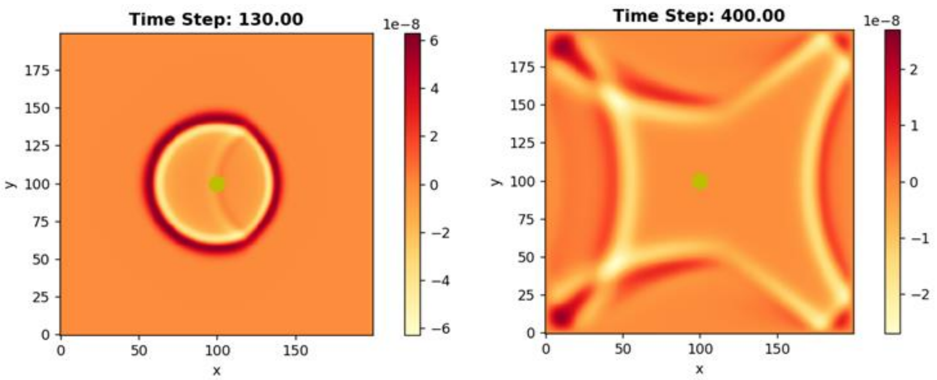

Figure 2. Wave propagation at different time steps

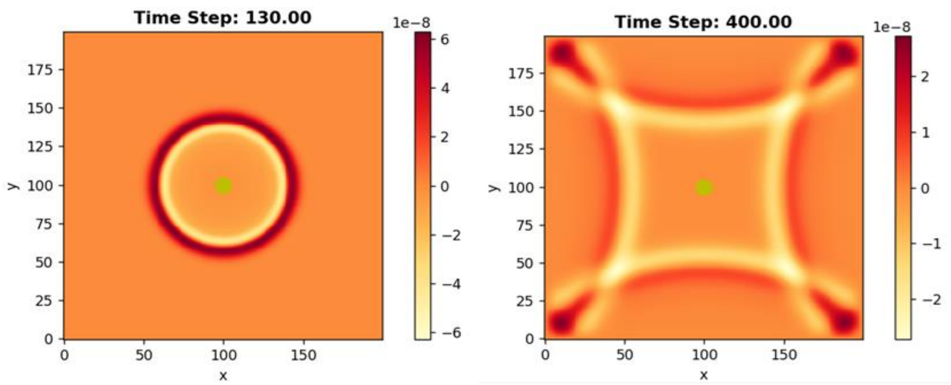

Figure 3. Wave propagation through a heterogeneous medium at different time intervals

4. PARALLEL EXECUTION USING THE GRAPHICS PROCESSING UNIT

A graphics processing unit (GPU) represents a massively parallel, multi-core processor renowned for its substantial computational capacity. Its affordability, impressive floating point throughput, and efficient memory access have garnered growing interest among experts in advanced computing performance [24]. The CUDA implementation encompasses three primary stages of computation: data transfer to GPU global memory, execution of the CUDA kernel [25], and retrieval of results from GPU memory to CPU memory. Although various parallel approaches were investigated in [32], this study opts for the CUDA approach, utilizing the cyclic reduction technique. The procedure for solving the problem is detailed in Algorithm 1.

Algorithm1. Execution of 2D wave equation

```
1. Compute initial condition matrix H_0
       Initialize matrix H with H_0 based on the initial condition (2).
2. Time-stepping loop:
   while t < t_end do:
3.Solve in the x-direction:
        For each row j=0,...,m
         For each column i=0,...,m
           calculate the tridiagonal system elements a_i,b_i,c_i,f_i
           call the function Cyclic_Re(a_i,b_i,c_i,f_i,y_i,m) to solve the tridiagonal system.
           calculate matrix H_x
4.Solve in the y-direction:
        For each column i=0,...,m
         For each row j=0,...,m
            calculate the tridiagonal system elements a_j,b_j,c_j,f_j
            call the function Cyclic_Re(a_j,b_j,c_j,f_j,y_j,m)
            calculate matrix H_y
5.Update the matrices:
           Swap H with H_x
           Swap H_0 with H_y.
           Update the time step t by incrementing t by Δt.
6. End of time-stepping loop.
```

In this context, H, H_0, H_x, H_y represent $\mathrm{H}_{i,j}^{k-1/2}, \mathrm{H}_{i,j}^{k}, \mathrm{H}_{i,j}^{k+1/2}, \mathrm{H}_{i,j}^{k+1}$ respectively. Within the $Cyclic\_Re()$ function, there are three device functions: $CRForw()$, $CR\_d()$, and $CRBackw()$, alongside one host function, $calDim()$. Initially, the block size needs to be calculated based on the matrix size and the forward and backward sub-steps. This involves a cycle where we iterate over the computation

```
for (i=0; i<log2(m+1)-1; i++) {
       ste_N=(m-pow(2.0, i+1))/pow(2.0, i+1)+1;
       calDim(ste_N,&dimBlock,&dimGrid);
       CRForwa<<>>(d_a, d_b, d_c, d_f, n, ste_N, i);
}
```

where $log2(m+1)-1$ is the step number, and the ste_N variable is used to determine the required block size. After the $calDim()$ function has determined the block size, the $CRBackw()$ function is executed $log2(m+1)-1$ times. Thus, the system can reduce the equation by one. The blocks are synchronized, followed by the invocation of the $CR\_d()$ function, which computes two unknowns. After that, a loop is used with the following structure:

```
for (i=log2(m+1)-2; i>=0; i) {
       ste_N=(m-pow(2.0, i+1))/pow(2.0, i+1)+1;
       calDim(ste_N, &dimBlock, &dimGrid);
       CRBackw<<>>(d_a, d_b, d_c, d_f, d_x, n, ste_N, i);
}
```

In this context, the backward substitution process is executed $log2(m+1)-2$ times, as the initial backward substitution sub-step is handled by the $calDim$ function. Subsequently, the d_x array is computed, and then this computed data is transferred from the device to the host using the *cudaMemcpy* function with the parameters $y, d\_x, sizeof(double)*n$, and $cudaMemcpyDeviceToHost$.

5. RESULTS AND DISCUSSION

In the following section, we present the findings derived from experiments conducted on a desktop system equipped with a GeForce RTX 2080 setup featuring 4352 cores. The system also comprises an NVIDIA GPU, Intel Core (TM) i7-9800X Processor clocked at 3.80 GHz, and 64 GB RAM. The modeling parameters are configured as follows: the grid size remains consistent in both directions, with $\Delta x = \Delta y$, while the time step Δt is set to 0.05. The simulation duration is T = 10.0, resulting in a total of 200-time steps. For a more comprehensive evaluation, we conducted tests using four different computational domain sizes: 512×512, 1024×1024, 2048×2048, and 4096×4096.

The effectiveness of a parallel algorithm is assessed by measuring its acceleration. Acceleration is determined by comparing the best runtime of a sequential algorithm with the longest runtime of the parallel algorithm for a given problem.

Table 1 provides insights into the execution durations (in seconds) for both the sequential implementation (CPU time) and the CUDA implementation (GPU time). Additionally, the table presents the execution times of cyclic reduction methods for tasks, along with the corresponding acceleration coefficients achieved on diverse devices. This comparative analysis highlights the efficiency gains offered by the GPU implementation, demonstrating the potential for substantial performance improvements when using parallel computing techniques over traditional sequential methods.

Table 1. Performance measurements and acceleration analysis using the Intel Core (TM) i7-9800X, 3.80 GHz, and NVIDIA RTX 2080 TI architecture

Mesh sizes	CPU time	GPU time	Acceleration
512 × 512	6.45	2.87	2.25
1024 × 1024	29.36	8.06	3.64
2048 × 2048	125.47	30.45	4.12
4096 × 4096	525.59	78.33	6.71

6. CONCLUSION

This article explores the computational resolution of a two-dimensional wave equation utilizing an implicit solution strategy. A technique for resolving the two-dimensional wave equation computationally via the finite difference method is outlined. Furthermore, a strategy for parallelizing the cyclic reduction method on a graphics processor is introduced. We devised a method for parallelizing the round-robin technique on NVIDIA GPUs and showcased its acceleration capabilities. Our experimentation with accelerating the round-robin method yielded promising results. We see that it was successful: the speedup achieved when implemented on the GPU was 6.71.

ACKNOWLEDGMENT

This research is funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP14972032).

REFERENCES

[1] A. Altybay, M. Ruzhansky, and N. Tokmagambetov, "A parallel hybrid implementation of the 2D acoustic wave equation," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 21, no. 7–8, pp. 821–827, Jul. 2020, doi: 10.1515/ijnsns-2019-0227.

[2] A. Klöckner, T. Warburton, J. Bridge, and J. S. Hesthaven, "Nodal discontinuous Galerkin methods on graphics processors," *Journal of Computational Physics*, vol. 228, no. 21, pp. 7863–7882, Nov. 2009, doi: 10.1016/j.jcp.2009.06.041.

[3] N. Bell and M. Garland, "Efficient sparse matrix-vector multiplication on CUDA," NVIDIA Technical Report NVR-2008-004, 2008.

[4] E. Elsen, P. LeGresley, and E. Darve, "Large calculation of the flow over a hypersonic vehicle using a GPU," *Journal of Computational Physics*, vol. 227, no. 24, pp. 10148–10161, Dec. 2008, doi: 10.1016/j.jcp.2008.08.023.

[5] Y. Zhang, J. Cohen, and J. D. Owens, "Fast tridiagonal solvers on the GPU," *ACM SIGPLAN Notices*, vol. 45, no. 5, pp. 127–136, Jan. 2010, doi: 10.1145/1837853.1693472.

[6] Y. Zhang, J. Cohen, A. A. Davidson, and J. D. Owens, "A hybrid method for solving tridiagonal systems on the GPU," in *GPU Computing Gems Jade Edition*, Elsevier, 2012, pp. 117–132. doi: 10.1016/b978-0-12-385963-1.00011-3.

[7] A. Davidson and J. D. Owens, "Register packing for cyclic reduction: a case study," *UC Davis: Institute for Data Analysis and Visualization*, Mar. 2011. doi: 10.1145/1964179.1964185.

[8] A. Davidson, Y. Zhang, and J. D. Owens, "An auto-tuned method for solving large tridiagonal systems on the GPU," in *2011 IEEE International Parallel & Distributed Processing Symposium*, May 2011, pp. 956–965. doi: 10.1109/IPDPS.2011.92.

[9] D. Goddeke and R. Strzodka, "Cyclic reduction tridiagonal solvers on GPUs applied to mixed-precision multigrid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 1, pp. 22–32, Jan. 2011, doi: 10.1109/tpds.2010.61.

[10] H.-S. Kim, S. Wu, L. Chang, and W. W. Hwu, "A scalable tridiagonal solver for GPUs," *2011 International Conference on Parallel Processing*, Taipei, Taiwan, 2011, pp. 444-453, doi: 10.1109/icpp.2011.41.

[11] N. Sakharnykh, "Tridiagonal solvers on the GPU and applications to fluid simulation," in *NVIDIA GPU Technology Conference*, 2009, pp. 17–19.

[12] Z. Wei, B. Jang, Y. Zhang, and Y. Jia, "Parallelizing alternating direction implicit solver on GPUs," *Procedia Computer Science*, vol. 18, pp. 389–398, 2013, doi: 10.1016/j.procs.2013.05.202.

[13] M. Souri, P. Akbarzadeh, and H. Mahmoodi Darian, "Parallel Thomas approach development for solving tridiagonal systems in GPU programming-Steady and unsteady flow simulation," *Mechanics and Industry*, vol. 21, no. 3, p. 303, 2020, doi: 10.1051/meca/2020013.

[14] M. A. Diaz, M. A. Solovchuk, and T. W. H. Sheu, "High-performance multi-GPU solver for describing nonlinear acoustic waves in homogeneous thermoviscous media," *Computers and Fluids*, vol. 173, pp. 195–205, Sep. 2018, doi: 10.1016/j.compfluid.2018.03.008.

[15] D. Michéa and D. Komatitsch, "Accelerating a three-dimensional finite-difference wave propagation code using GPU graphics cards: Accelerating a wave propagation code using GPUs," *Geophysical Journal International*, May 2010, doi: 10.1111/j.1365-246x.2010.04616.x.

[16] R. Mehra, N. Raghuvanshi, L. Savioja, M. C. Lin, and D. Manocha, "An efficient GPU-based time domain solver for the acoustic wave equation," *Applied Acoustics*, vol. 73, no. 2, pp. 83–94, Feb. 2012, doi: 10.1016/j.apacoust.2011.05.012.

[17] M. Lastra, M. J. Castro Díaz, C. Ureña, and M. de la Asunción, "Efficient multilayer shallow-water simulation system based on GPUs," *Mathematics and Computers in Simulation*, vol. 148, pp. 48–65, Jun. 2018, doi: 10.1016/j.matcom.2017.11.008.

[18] A. P. Engsig-Karup, H. B. Bingham, and O. Lindberg, "An efficient flexible-order model for 3D nonlinear water waves," *Journal of Computational Physics*, vol. 228, no. 6, pp. 2100–2118, Apr. 2009, doi: 10.1016/j.jcp.2008.11.028.

[19] D. Greaves, "Application of the finite volume method to the simulation of nonlinear water waves," in *Advances in Numerical Simulation of Nonlinear Water Waves*, World Scientific, 2010, pp. 357–396. doi: 10.1142/9789812836502_0010.

[20] G. R. Richter, "An explicit finite element method for the wave equation," *Applied Numerical Mathematics*, vol. 16, no. 1–2, pp. 65–80, Dec. 1994, doi: 10.1016/0168-9274(94)00048-4.

[21] D. Komatitsch and J. Tromp, "Spectral-element simulations of global seismic wave propagation-II. Three-dimensional models, oceans, rotation and self-gravitation," *Geophysical Journal International*, vol. 150, no. 1, pp. 303–318, Jul. 2002, doi: 10.1046/j.1365-246x.2002.01716.x.

[22] A. Altybay, A. Nakiskhozhayeva, and D. Darkenbayev, "Numerical simulation and parallel computing of the acoustic wave equation," 2024. doi: 10.1063/5.0194676.

[23] H.-L. Liao and Z.-Z. Sun, "A two-level compact ADI method for solving second-order wave equations," *International Journal of Computer Mathematics*, vol. 90, no. 7, pp. 1471–1488, Jul. 2013, doi: 10.1080/00207160.2012.754016.

[24] A. Altybay, D. Darkenbayev, and N. Mekebayev, "Numerical simulation and GPU computing for the 2D wave equation with variable coefficient," *International Journal of Simulation and Process Modelling*, vol. 1, no. 1, 2024, doi: 10.1504/IJSPM.2024.10062867.

[25] A. A. Samarskii, *The theory of difference schemes*. CRC Press, 2001. doi: 10.1201/9780203908518.

[26] D. W. Peaceman and H. H. Rachford, Jr., "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, Mar. 1955, doi: 10.1137/0103003.

[27] R. M. Weiss and J. Shragge, "Solving 3D anisotropic elastic wave equations on parallel GPU devices," *Geophysics*, vol. 78, no. 2, pp. F7----F15, Mar. 2013, doi: 10.1190/GEO2012-0063.1.

[28] G. Balakayeva, G. Kalmenova, D. Darkenbayev, and C. Phillips, "Development of an application for the thermal processing of oil slime in the industrial oil and gas sector," *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, vol. 13, no. 2, pp. 20–26, Jun. 2023, doi: 10.35784/iapgos.3463.

[29] G. T. Balakayeva, C. Phillips, D. K. Darkenbayev, and M. Turdaliyev, "Using NoSQL for processing unstructured big data," *NEWS of National Academy of Sciences of the Republic of Kazakhstan*, vol. 6, no. 438, pp. 12–21, Dec. 2019, doi: 10.32014/2019.2518-170x.151.

[30] D. Darkenbayev, A. Altybay, Z. Darkenbayeva, and N. Mekebayev, "Intelligent data analysis on an analytical platform," *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, vol. 14, no. 1, pp. 119–122, Mar. 2024, doi: 10.35784/iapgos.5423.

[31] R. Baimakhan, Z. Kadirova, A. Seinassinova, A. Baimakhan, and Z. Abdiakhmetova, "Mathematical modeling of the water saturation algorithm of the mountain slope on the example of the catastrophic landslide of the Northern Tien Shan Ak Kain," *Mathematical Modelling of Engineering Problems*, vol. 8, no. 3, pp. 467–476, Jun. 2021, doi: 10.18280/mmep.080318.

[32] A. Temirbekov, A. Altybay, L. Temirbekova, and S. Kasenov, "Development of parallel implementation for the Navier-Stokes equation in doubly connected areas using the fictitious domain method," *Eastern-European Journal of Enterprise Technologies*, vol. 2, no. 4 (116), pp. 38–46, Apr. 2022, doi: 10.15587/1729-4061.2022.254261.

BIOGRAPHIES OF AUTHORS

Arshyn Altybay Ph.D., associate professor of the Department of Computer Science, Faculty of Information Technologies, Al-Farabi Kazakh National University and researcher at the Institute of Mathematics and Mathematical Modeling, Almaty, Kazakhstan. Research interests: numerical simulation of PDE, numerical analysis, high-performance computing, numerical methods, parallel programming, machine learning, deep learning in inverse problems. He can be contacted at email: arshyn.altybay@gmail.com.

Dauren Darkenbayev Ph.D., associate professor of the Department of Computer Science, Faculty of Information Technologies and Department of Computational Sciences and Statistics, Faculty of Mechanics and Mathematics, Al-Farabi Kazakh National University, Almaty, Kazakhstan. His research interests include big data processing, mathematical, and computer modeling, development of computer systems for the educational process, machine learning, deep learning in inverse problems. He can be contacted at email: dauren.kadyrovich@gmail.com.

Nurbapa Mekebayev Ph.D. in computer science from Al-Farabi Kazakh National University, Almaty, Kazakhstan, in 2020. Currently, he is a senior researcher at the Institute of Information and Computing Technologies in Almaty, Kazakhstan, and an associate professor at the Department of Computer Science of the Kazakh National Women's Teacher Training University in Almaty, Kazakhstan. His research interests include machine learning, deep learning inverse problems and computational linguistics. He can be contacted at email: nurbapa@mail.ru.