A Technique for Large-Scale 2D Seismic Field Simulations on Supercomputers

Dmitry A. Karavaev, Valery V. Kovalevsky
Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

Abstract – **An algorithm for the simulation of elastic wave propagation in 2D isotropic inhomogeneous media with complex geometrical structure is presented. A parallel implementation of the FDTD method of fourth order in space to perform calculations on high-performance computing systems with different architectures (CPUs, GPUs, or Xeon Phi coprocessors) is discussed. The proposed technique of mathematical modeling with the use of MPI and CUDA is used to design program codes for computations for a realistic long-distance model. Program codes for single device and multi device use are developed. A large-scale geophysical model of the Baikal rift zone is reconstructed. New synthetic data depicting the structure of the seismic field for the rift zone are presented.**

Index Terms – **Simulation, seismic waves, parallel algorithm, large-scale, cluster.**

I. Introduction

THEORETICAL and experimental studies of seismic wave propagation in inhomogeneous media with complex subsurface structure are very important [1-3]. In this paper, a large-scale model of the Baikal rift zone is presented as the main object of investigation [4]. Because of geological investigations, several geophysical models of the rift zone can be found (for instance, the Baikal Explosion Seismic Transect Project and a Program for the Seismic Study of Continental Lithosphere). Experiments on mathematical modeling of elastic wave propagation will allow obtaining a new knowledge about the seismic field structure and, possibly, help us to clarify the velocity model. Obtaining of numerical results can be a difficult problem due to simulation for long distances. The rift zone has average linear sizes of several hundreds of km along the horizontal coordinate. It is of great interest to reconstruct the subsurface structure of large-scale models. It can be achieved when finding the most appropriate solution from a set of models for which simulation was performed.

Supercomputers have become an important tool for successful solving of problems in different areas of knowledge [5-7]. There is a variety of supercomputer architectures. These clusters can be designed based on CPUs, graphic cards (GPUs), or many-core computing devices known as Xeon Phi [5, 6]. Such clusters allow access to hundreds and thousands of computing cores. Today it is important to know the architecture of high-performance systems and the features of computing devices to develop program codes that will allow efficient use of the computing resources [8, 9]. Such systems and programs can help us to obtain numerical simulation results for highly detailed geophysical models. Research work on efficient parallel algorithm realizations and program code design for different supercomputer architectures is under way. This approach provides effective use of the computing capabilities of a computing system to obtain high accuracy in reasonable time [10-13].

One of the goals of the present study is to develop and present a technique in the area of seismic simulation to perform calculations on high-performance multi- and many-core computing systems.

In this case, search for and development of a mathematical method ensuring acceptable accuracy of calculations has become an important task. Various approaches and mathematical methods have been developed to help find a numerical solution to the dynamic problem of elasticity theory: finite-difference methods, finite element methods, and others. Finite difference methods are attractive because of their efficient and simple realization due to specific approximation of the partial differentials [14-17]. This method makes it possible to design parallel algorithms and program codes to perform calculations via multi core devices.

In the present study, a technique for the simulation of seismic field in inhomogeneous media is developed. Design of a parallel algorithm and development of program codes for calculations on clusters with different architectures are considered. Such program codes are mainly designed for supercomputers of the General-Purpose Siberian Supercomputer Center of the Siberian Branch of the Russian Academy of Sciences, SSCC SB RAS. The developed realization can be ported in a simple way to work with another many core architecture of computing device, for example , Xeon Phi. The Xeon Phi and GPU have different architectures and different numbers of computing cores and, thus, one can have a choice to parallelize via OpenMP or CUDA. Sometimes it is much easier to adapt an algorithm and program written on C++ to design a parallel code for the Xeon Phi. One of the features when working with CUDA is the need to copy the data between CPU and GPU for an iterative method with data distributed among the parallel processes. Nowadays there are computing architectures constructed only with directly interconnected Xeon Phi, for example, the NKS-1P cluster at SSCC. In this case there is no need in data copying.

In the paper, the problem statement and the mathematical method are presented and briefly discussed. To make the amplitude of waves reflected from the artificial boundaries of the computation area, the PMLs (Perfectly Matched Layers) are applied [18-23]. In the proposed study, a parallel

978-1-5386-7054-5/18/$31.00 ©2018 IEEE

realization for numerical modeling of 2D seismic wave propagation is discussed. The MPI and CUDA/OpenMP can be used for parallelization. The main features of the programs developed are presented, for example, the use of buffer arrays. The behavior of a multi-GPU program code is studied in various tests. New results of seismic field simulation for large-scale model on a high-performance cluster are presented. It is shown that practical application of the technique being proposed for scientific purposes is possible.

II. Problem Definition

In this paper, a solution to a forward problem of geophysics of the full wave field simulation for the isotropic case of inhomogeneous elastic media is considered. To solve the problem, a system of equations of elasticity theory (1) written in terms of the velocities and stresses is presented. The solution to the problem is considered in a two-dimensional Cartesian coordinate system with Ox (horizontal) and Oz (vertical) axes. The model area has the form of a rectangle whose upper boundary (z = 0) is taken as the free surface (see Fig. 1). Models with only plane geometry of the free surface are considered in the present study.

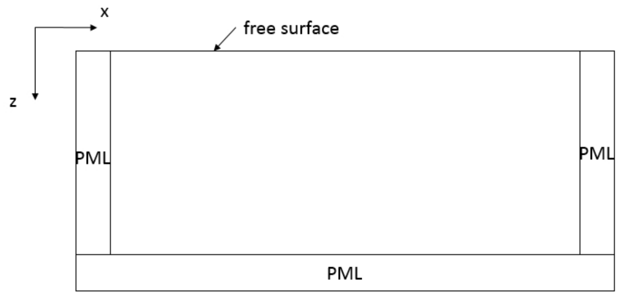


Fig. 1. Schematic representation of 2D volume.

The elastic medium is described by the following three parameters: ρ, is the density, Vp and Vs are the longitudinal and transverse wave velocities, respectively. The geometry structure of the rectangular region of interest is determined by the distribution of the values of the elastic parameters. It should be noted that in the equations of the system (1) these parameters are constituents of the Lamé parameters. These parameters are two-dimensional functions of the coordinates.

$$\begin{aligned} \rho u_t &= \tau_{xx,x} + \tau_{xz,z} \quad , \\ \rho w_t &= \tau_{xz,x} + \tau_{zz,z} \quad , \\ \tau_{xx,t} &= (\lambda + 2\mu) u_x + \mu u_z + f_x \quad , \\ \tau_{zz,t} &= \mu u_x + (\lambda + 2\mu) u_z + f_z \quad , \\ \tau_{xz,t} &= \mu (u_z + w_x) \end{aligned} \tag{1}$$

where

$$u_t = \frac{\partial u}{\partial t} \quad , \tau_{ij,j} = \frac{\partial \tau_{ij}}{\partial x_j} \quad , x_1 = x, x_2 = z$$

The problem is to be solved with corresponding zero initial and boundary conditions.

To obtain a successful solution without waves reflected from the interior boundaries, the PML (Perfectly Matched Layers) technique was applied [22, 23]. For this, rectangular subregions of small size are located along the side boundaries and along the bottom boundary of the rectangular region (see Fig. 1). In this area, the calculation formulas of the C-PML (convolutional PML) are used.

A point source is used to perform the calculations and generate seismic waves. In (1), the source enters as f_x and f_z, respectively. The source can be realized as a “comprehensive force” being inside the modeling area excluding PML layers, or as a “vertical force” located at the free surface.

III. Theory

When choosing a method for numerically finding a solution, itis important that it is flexible enough and is capable of designing a parallel realization to perform calculations on high-end computing architectures. It should also provide sufficient accuracy of calculations for long distances. In this case, a finite-difference method of fourth order with respect to space was chosen [24]. In addition, a realization of the finite-difference approximation of the free surface boundary conditions was made [25]. The finite-difference approximations of the equations to calculate seismic wave propagation have a general form (3):

$$\tau^{n+1}_{xx,i,j} = \tau^{n}_{xx,i,j} + \frac{\Delta t}{\Delta x}((\lambda+2\mu)_{i,j}[A(u^n_{i+\frac{3}{2},j} - u^n_{i-\frac{3}{2},j}) + B(u^n_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j})] + \lambda_{i,j}[A(w^n_{i+\frac{3}{2},j} - w^n_{i-\frac{3}{2},j}) + B(w^n_{i+\frac{1}{2},j} - w^n_{i-\frac{1}{2},j}) + f_x^{n+1}]), A = -\frac{1}{24}, B = \frac{9}{8} \tag{3}$$

In (2) i, j are the grid coordinates and $n, n+1$ are the time layers. $\Delta t, \Delta x$ are the grid step sizes in time and space, respectively. A finite-difference scheme was designed using staggered grids. In this case the velocities and stresses are located at different points with respect to the grid cell (see Fig.2). In the present study, the horizontal component of the seismic field u and the stresses τ_{xx}, τ_{zz} are located on the free surface.

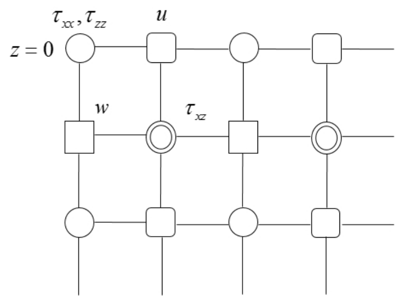


Fig. 2. Locations of velocity and stress components for the FD operators on a 2D mesh.

The C-PML formulas include damping functions along the x and z coordinate axes. In the PML zones each spatial derivative is modified in the form of (4):

$$\frac{\partial}{\partial x_j} = \frac{1}{k_j}\frac{\partial}{\partial x_j} + \psi_j \tag{4}$$

where the so-called "memory variable" ψ_j has a time dependent evolution formula with damping coefficients a_j and b_j (5):

$$b_j = e^{-(\frac{d_j}{k_j}+\alpha_j)\Delta t}, a_j = \frac{d_j}{k_j(d_j + k_j\alpha_j)}(b_j - 1) \qquad (5)$$

$$k_j \geq 1, d_j \geq 0, \alpha_j \geq 0$$

A detailed description of the C-PML technique being used can be found in [23].

III. Parallel Implementation

All the calculation programs are designed for clusters based on CPUs (Central Processing Units) or special computing devices, for example, GPUs (Graphics Processing Units). The program codes were parallelized based on the data decomposition approach. The 1D domain decomposition technique was used to organize subdomains (layers) along the Oz axis for parallel computation for each of them. Therefore, the full modeling area of rectangular form consists of several regions (rectangular sub-domains), see Fig. 3.

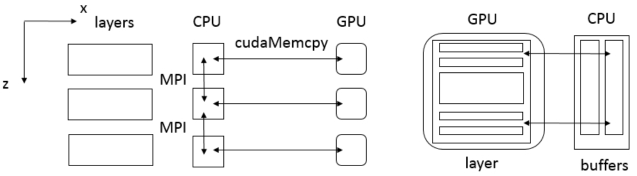


Fig. 3. Multiple device organization of parallel computations on CPU+GPU cluster.

Thus, the parallel computing technique includes design of program codes realizing the 1D topology for the MPI processes, data exchange procedures, and calculation parts. The main program includes allocation of memory on the computing devices, devices management, initialization of parallel processes, creation of 1D MPI (Message Passing Interface) topology, performing data send/receive procedures, and data copying between the CPUs and computing devices (in the GPUs realization). To create the topology, the MPI_Dims_create, MPI_Cart_create, MPI_Cart_coords, and MPI_Cart_shift procedures are used. Their specifications and sample codes can be easily found. The calculation part of the algorithm was realized in another program code. This code includes computing procedures to perform 1D and 2D calculations for the mesh partitioned between layers. Such procedures can be parallelized with the OpenMP (Open Multi-Processing) or CUDA (Compute Unified Device Architecture) technology [24]. Each layer is computed on the (CPU, GPU, or Xeon Phi) device. Several types of computing procedures have been developed. One of them is to perform calculations for a set of points located on the boundaries of neighboring layers (devices), see Fig.4. Such points are computed first in a parallel realization. Another procedure runs the non-blocked data exchange procedures MPI_Isend and MPI_Irecv for the buffers allocated on the CPUs. At the same time 2D parallel computations (CUDA kernels) are run on the GPUs. Therefore, all data exchanges are "hidden behind" the execution of the procedures for 2D calculations for the other points of the mesh.

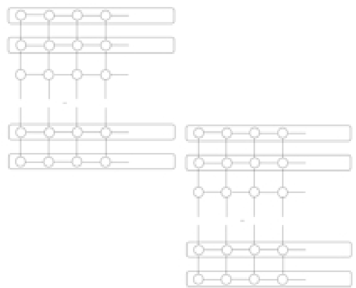


Fig. 4. Representation of grid cell locations for data exchanges between neighboring layers/subdomains (computing devices).

In the program codes, all arrays that can be 2D data, for example, the velocity and stress fields and elastic parameter values on the 2D mesh, are converted into 1D arrays. It allows easy operation with CUDA.

The buffers are designed for data exchanges between the neighboring devices located at separate computing nodes of the cluster. This approach is necessary for a multi-GPU realization, because each device has its own memory space. There are two buffer types: for data exchange with the bottom layer and with the top layer in the linear topology. In contrast to the parallel realization for the FD scheme of second-order accuracy with respect to space [25], in the present study it is important to make data exchanges between two sets of 1D points placed on the top or bottom of a layer (see Fig.4). The buffers contain information about the values of velocities and stresses that need to be copied to the top or bottom layer with respect to the selected one, excluding the layers with the MPI ranks 0 and N-1 in the topology (see Fig. 3). Buffers are initialized only on the CPUs. The values in the grid cells (points) are interchanged in the following way. After the points on the layer boundaries are calculated, they should be copied into the corresponding parts of the buffers on the CPU with "cudaMemcpy" procedures. The next step the MPI_Isend and MPI_Irecv procedures are runs. At the same time, the 2D kernels are performed on the GPUs for every MPI process. After all the exchange procedures are completed, the MPI processes can check this with the MPI_Wait. After the data from the CPU buffers are copied back into the 1D arrays, the next iteration step can be executed. To present a well-developed program code for multi-GPU use, some tests for the parallel algorithm are carried out.

The parallel realization proposed works only with the "global memory" common to all parallel threads on the GPU. The FD computations were performed with procedures run on GPUs with the following code lines: <<<dim3(nblocks_x,1,1),dim3(block_lenght,1,1)>>> for a 1D parallelization and <<<dim3(nblocks_x,nblocks_y, 1),dim3(block_lenght,block_height,1)>>> for a 2D one. This representation is common because of the code executed on the GPU via "kernels". The number of parallel working threads is controlled by the values of block_length and block_height. The procedures are parallelized with the indexes along the Ox and Oz axes depending on threadIdx.x, blockIdx.x, and block_length/block_height.

TABLE I
BEHAVIOR OF THE PROGRAM CODES IN TESTS

	Number of GPU devices			
Test	1	3	6	9
Acceleration	882 sec	474 sec	264 sec	211 sec
Scalability	-	395 sec	482 sec	482 sec

For the scalability test, a model of inhomogeneous media with a mesh of 9998x2000 cell points per one process was taken. The number of time iterations was 28873. The scalability test was performed for 3, 6, and 9 computing devices. The results of the program code behavior for the test are given in Tab. I. The presented time values show almost the same results. It should be noted that the result for thee computing devices differs from that for the rest ones. It is due to program code runs on one computing node and, thus, there is no need for data exchange between devices placed on different computing nodes.

The acceleration test was performed for a mesh of 9998x2000 cell points for all devices. The total mesh was partitioned among the devices into subdomains of approximately equal computational size, number of cell points. In the test two program codes were used. The first code is for a single-GPU, and the second one, for multi-GPU use. The results are presented in Tab. I.

IV. EXPERIMENTAL RESULTS

The proposed technique of 2D mathematical modeling in geophysics and the developed program codes are designed for full seismic field simulation for long distances. The purpose is to study the subsurface structure and determine the values of the elastic parameters for the Baikal rift zone. This object is characterized by sufficiently large linear sizes along the coordinate axes. It is over 400 km along the Ox axis and about 80 km along the Oz axis. Wellbore data were used to prepare input data with the distribution of elastic parameters onto the mesh. These files contain information about the parameter values on vertical horizons (points with the z coordinate in the km format and with the corresponding values of the elastic parameters). To reconstruct the rift zone structure and to define the elastic parameter values into the grid cells, a model builder program was used. This program can help to construct a geophysical model based on borehole input data using the cubic spline interpolation algorithm. In the program, first 1D interpolation along the Oz axis is used, and then 1D interpolation along the Oz axis. This method helps to get interpolated values for a 2D mesh. As a result of the program operation, three data sets were obtained for the 2D seismic simulation.

To reconstruct the model, more than 10 reference points showing 1D distribution of the elastic parameter values in the vertical profiles were used. The model area of 400 km x 73 km was reconstructed (see Fig. 5). This model demonstrates highly inhomogeneous non-contrast character of distribution of the elastic values. It is obtained from input data with a maximum value of Vp of 8.3452 km/s and a minimum value of Vs of 2.4672 km/s.

Fig. 5. Subsurface structure of the Baikal rift zone model. Distribution of S-wave velocity.

To simulate the seismic field along a 400 km profile, a seismic source with a dominant frequency of 7 Hz was used. A source of the "vertical force" type has been chosen. It means that the force was applied to a selected point (grid cell) placed on the free surface. The x coordinate of the source location was taken about 15 km. For sufficient accuracy of the results, 50 grid points per minimum wavelength were taken in the calculations. These results have been obtained in studies on the algorithm verification using test models.

A mesh with 56743 x 10356 grid points was reconstructed to simulate a 2D full seismic field. The program output files are presented in the binary format. The files contain information about the amplitudes of the vertical and horizontal (positive or negative) components of the seismic field on the mesh. Moreover, files with seismograms are obtained, since in natural experiments only seismogram data can be obtained for the subsequent investigations. The so-called snapshots of the seismic field help in interpreting the seismograms and revealing different types of elastic waves.

All calculations of the seismic field for the rift zone were carried out on the NKS-30T cluster of the SSCC SB RAS, http://www.sscc.icmmg.nsc.ru. The program code for calculations on CPUs was used. 12 computational nodes of the cluster with 12 cores in one node were used in the calculations. Each core is allocated one MPI process. In Fig. 6 one can see the propagation of elastic waves in the rift zone structure. All pictures in Fig. 6 were obtained with the Matlab software.

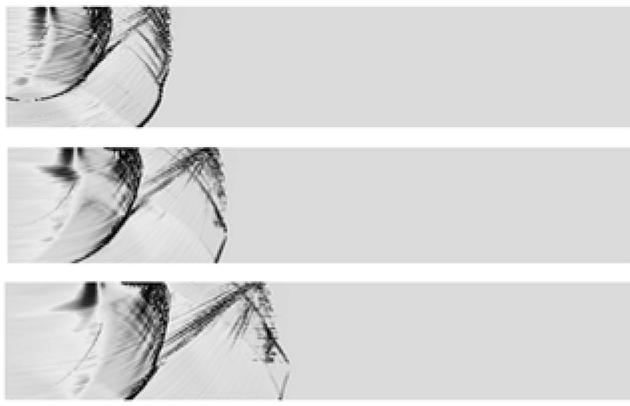


Fig. 6. Snapshots of the vertical component of the rift zone seismic field for time series of 15, 20, and 25 seconds (from top to bottom).

Various types of elastic waves (reflected, refracted, surface, and others) are presented, see Fig. 6. The obtained synthetic data contain a new knowledge about the seismic field structure and the object of study.

V. CONCLUSIONS

A technique to solve a dynamic problem of elasticity theory associated with modeling of the full seismic field from a point source has been proposed. A parallel implementation of the FDTD method for numerical experiments has been presented. The developed mathematical modeling technique and program codes for calculations on high-performance computing systems with CPUs or GPUs/Xeon Phi's are suitable for a wide range of geophysical models. The programs use a single GPU device for test or small size calculations and multiple device parallelism for long-distance calculations. With the help of the developed programs, a geophysical model of the Baikal rift zone has been reconstructed. To determine the values of the elastic parameters at the grid points, wellbore input data were used. Such data contain information about the distribution of the elastic parameter values along the vertical profiles. For the model, a FD simulation of seismic wave propagation has been made. For such a geophysical object, synthetic data in the form of seismic wave snapshots have been presented. All theoretical experiments were carried out on the high-performance cluster of the SSCC SB RAS. The results obtained can help to carry out study on comparison of synthetic and experimental data, which will allow refining and correcting the geophysical model.

This work was supported by the Russian Foundation for Basic Research (projects no. 16-07-01052, no. 17-07-00872) and by the Russian Science Foundation (project no. 18-11-00044).

REFERENCES

[1] Maeda T., Takemura S. and Furumura T. OpenSWPC: an open-source integrated parallel simulation code for modeling seismic wave propagation in 3D heterogeneous viscoelastic media // Earth, Planets and Space 69:102, 2017. DOI 10.1186/s40623-017-0687-2

[2] Komatitsch, D., et al. High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster // J. Comput. Phys. 229(20), 7692–7714

[3] Glinskiy, B., Sapetina, A., Martynov, V., Weins, D., Chernykh, I. The Hybrid-Cluster Multilevel Approach to Solving the Elastic Wave Propagation Problem // PCT 2017: Parallel Computational Technologies pp 261-274

[4] Mordvinova V.V., Artemiev A.A. The three-dimensional shear velocity structure of lithosphere in the southern Baikal rift system and its surroundings // Russian Geology and Geophysics, 2010, Vol. 51, No. 6. pp. 694-707. (in Russian)

[5] Heinecke A., Breuer A., Bader M., Dubey P. High Order Seismic Simulations on the Intel Xeon Phi Processor (Knights Landing) // High Performance Computing, 2016, pp 343-362

[6] Tobin J., Breuer A., Heinecke A., Yount C., Cui Y. Accelerating Seismic Simulations Using the Intel Xeon Phi Knights Landing Processor // In: Kunkel J., Yokota R., Balaji P., Keyes D. (eds) High Performance Computing. ISC 2017. Lecture Notes in Computer Science, 2017, vol 10266. Springer. pp 139-157

[7] Glinskiy B., Kulikov I., Chernykh I., Snytnikov A., Sapetina A., Weins D. The Integrated Approach to Solving Large-Size Physical Problems on Supercomputers // In: Voevodin V., Sobolev S. (eds) Supercomputing. RuSCDays 2017. Communications in Computer and Information Science, 2017, vol 793. Springer, Cham. https://doi.org/10.1007/978-3-319-71255-0_22

[8] Kulikov, I., Chernykh, I., Glinsky, B. Numerical simulations of astrophysical problems on massively parallel supercomputers // AIP Conference Proceedings 1776, 090006 2016. https://doi.org/10.1063/1.4965370

[9] Iturrarán-Viveros U. and Molero-Armenta M. GPU computing with OpenCL to model 2D elastic wave propagation: exploring memory usage // Comput. Sci. Disc. 8, 2015.

[10] Li Y., Métivier L., Brossier R., Han B. and Virieux J. 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver // Geophysics, 2015, Vol. 80, No. 3. pp. T101–T118

[11] Castro M., Francesquini E., Dupros F., Aochi H., Navaux P., et al. Seismic Wave Propagation Simulations on Low-power and Performance-centric Manycores // Parallel Computing, Elsevier, 2016, Volume 54, pp. 108-120

[12] Qawasmeh A., Chapman B., Hugues M., Calandra H. GPU Technology Applied to Reverse Time Migration and Seismic Modeling via OpenACC // Proceeding PMAM '15 Proceedings of the Sixth International Workshop on Programming Models and Applications for Multicores and Manycores, 2015. pp. 75-85

[13] Darmawan J. B. B. and Mungkasi S. Performance of parallel computation using CUDA for solving the one-dimensional elasticity equations // J. Phys.: Conf. Ser. 801 012080, 2017, doi:10.1088/1742-6596/801/1/012080

[14] O'Reillya O., Lundquist T., M.Dunhama E., Nordströmb J. Energy stable and high-order-accurate finite difference methods on staggered grids // Journal of Computational Physics, 2017, 346, pp. 572–589

[15] O'Reilly O., Nordström J., Kozdon J.E., Dunham E.M. Simulation of earthquake rupture dynamics in complex geometries using coupled finite difference and finite volume methods // J. Commun. Phys. 2015, 17(2). pp. 337–370, http://dx.doi.org/10.4208/cicp.111013.120914a.

[16] Ren Z., Liu Y. Acoustic and elastic modeling by optimal time-space-domain staggered-grid finite-difference schemes // Geophysics, 2015, 80(1). pp. T17–T40.

[17] Yong P., Huang J., Li Z., Liao W., Qu L., Li Q., Liu P. Optimized Equivalent Staggered-grid FD Method for Elastic Wave Modeling Based on Plane Wave Solutions // Geophysical Journal International, 2017, Volume 208, Issue 2, pp. 1157–1172

[18] Assi H., Cobbold R. S. A second-order, perfectly matched layer formulation to model 3D transient wave propagation in anisotropic elastic media // The Journal of the Acoustical Society of America 140, 2016, 3261. https://doi.org/10.1121/1.4970329

[19] Collino F. and Tsogka C. Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media // Geophysics, 66(1), 2001, pp. 294-307. https://doi.org/10.1190/1.1444908

[20] Komatitsch, D. and Tromp J. A Perfectly Matched Layer (PML) absorbing condition for the second-order elastic wave equation // Geophys. J. Int. 154, 2003. pp. 146-153

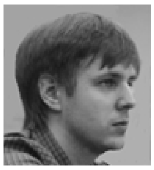
[21] Komatitsch D., Martin R. An unsplit convolutional Perfectly Matched Layer improved at grazing incidence for the seismic wave equation // Geophysics, 2007, vol. 72. pp. 155-167

[22] Levander A. Fourth-order finite difference P-SV seismograms // Geophysics 53, 1988. pp. 1425–1436

[23] Moczo P. Introduction to Modeling Seismic Wave Propagation by the Finite-Difference Method // Lecture Notes, Kyoto University, 1998.

[24] CUDA C Programming Guide. 2017

[25] Karavaev D.A., Glinsky B.M., Kovalevsky V.V. A Technology of 3D Elastic Wave Propagation Simulation Using Hybrid Supercomputers // CEUR Workshop Proceedings 1st Russian Conference on Supercomputing Days, 2015, Vol. 1482. pp. 26-33



Karavaev Dmitry, PhD in Physical and Mathematical Sciences, Researcher. Research interests: parallel computing, algorithms design, program codes development, simulation. Author and co-author of more than 30 scientific papers.

Kovalevsky Valery, Professor (Engineering), Deputy Director for Science at the Institute of Computational Mathematics and Mathematical Geophysics SB RAS. Research interests: mathematical simulation in geophysics, vibroseismic investigations, geophysical informatics. Author and co-author of more than 150 scientific papers.