Bull. Nov. Comp. Center, Num. Anal., 18 (2016), 57–66

Supercomputer-aided comparison of the efficiency of using different mathematical statements of the 3D geophysical problem*

A.F. Sapetina

Abstract. In order to create systems of vibroseismic monitoring for earthquake-prone areas it is needed to carry out the simulation of seismic wave propagation in the media typical of volcanic structures. To this end it is required to develop supercomputer technologies for decreasing the computation time and simulation of "big" 3D elastic media. Analysis of the efficiency of two parallel implementations of algorithms for solving the 3D elastodynamic problem written in different terms is carried out with the co-design. The software has been developed for both approaches and optimized for the architecture of a supercomputer equipped with GPU. A comparative analysis of the computation time, the size of the memory used and the software scalability is made.

Keywords: elastic waves, 3D simulation, co-design, finite difference schemes, hybrid cluster, GPU.

1. Introduction

The development of a highly-efficient program tool for a modern supercomputer system is a specific complex scientific problem. Its solution increasingly depends on the architecture of a supercomputer. Also, it subordinates not only the choice of algorithms for a concrete application problem to a computing architecture, but requires the co-design of algorithms at all stages of solving a problem: from its a statement to the selection of development tools. The concept of the co-design in the context of mathematical modeling of physical processes is understood as constructing physical and mathematical models of a certain phenomenon, numerical method, parallel algorithm and its software implementation, with the effective use of the supercomputer architecture. The co-design is successfully applied to the development of software when modeling diverse physical processes on supercomputers. For example, it is used in solving problems of plasma physics [1], astrophysics [1, 2] and many others. In the approach proposed, comparison of the efficiency of using various physical and mathematical statements becomes relevant when developing an appropriate software.

In this study, the efficiency of the parallel implementation of the two 3D algorithms for a hybrid supercomputer equipped with graphics cards is

*Supported by the RFBR under Grant 16-07-00434 A.

compared. The investigated algorithms perform the numerical simulation of seismic wave fields in 3D elastic media typical of volcanic structures, based on different statements of the dynamic elasticity. The first algorithm solves the problem set in terms of the velocities of displacements and stress, and the second one solves the same problem, but written in terms of displacements.

The simulation of seismic wave fields in 3D elastic media can be relevant to create systems of vibroseismic monitoring for volcanic structures. The creation of such systems requires preliminary and attendant complex studies of concrete volcanoes. Usually, the relief of an object under investigation is sufficiently complicated. Thus, this does not allow one to maintain an observational system for solving the inverse geophysical problem. Therefore it is necessary to solve a set of direct problems by varying the geometry and selecting the parameters of a simulated medium so that the results of numerical and physical experiments were close.

The complexity and scale of a simulated area necessitate the creation of a high-performance software that would allow solving rapidly enough the direct geophysical problem for the "big" 3D elastic media. Let us note that the performance is especially important for the attendant simulation in the process of monitoring and eruption forecasting of a real volcano. Thus, we need a software tool that would use a supercomputer and allow carrying out rapidly enough the selection of assumed data of a volcanic structure for interpreting a real experiment.

The two dynamic elasticity problem statements are considered and described in the next section to study the efficiency of the algorithms in terms of the memory used and implementation speed with supercomputer systems with graphic cards. On this basis, the software optimized for the GPU architecture and hybrid cluster has been developed, and the time spent on the realizations obtained is compared.

2. The description of mathematical statements of the elastodynamic problem

To simulate seismic waves in complicated elastic inhomogeneous media, we can solve the complete system of elasticity equations with appropriate initial and boundary conditions written down in terms of the displacement velocity vector $\vec{u}$ and the stress tensor $\vec{\sigma}$:

$$\vec{u} = (u, v, w)^T, \quad \vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T.$$

As the domain of simulation we take an isotropic 3D-inhomogeneous elastic medium of complex subsurface geometry which is a parallelepiped, one of whose sides is a free surface (the plane $z = 0$). Let us consider a rectangular Cartesian coordinate system where the axis Oz is directed

vertically downwards and the axes Ox, Oy are along the free surface. The constitutive equations can be expressed in the vector form as

$$\rho\frac{\partial\vec{u}}{\partial t} = A\vec{\sigma} + \vec{F}(t,x,y,z), \qquad \frac{\partial\vec{\sigma}}{\partial t} = B\vec{u}, \tag{1}$$

$$A = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial z} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix},$$

$$B = \begin{bmatrix} (\lambda+2\mu)\frac{\partial}{\partial x} & \lambda\frac{\partial}{\partial y} & \lambda\frac{\partial}{\partial z} \\ \lambda\frac{\partial}{\partial x} & (\lambda+2\mu)\frac{\partial}{\partial y} & \lambda\frac{\partial}{\partial z} \\ \lambda\frac{\partial}{\partial x} & \lambda\frac{\partial}{\partial y} & (\lambda+2\mu)\frac{\partial}{\partial z} \\ \mu\frac{\partial}{\partial y} & \mu\frac{\partial}{\partial x} & 0 \\ \mu\frac{\partial}{\partial z} & 0 & \mu\frac{\partial}{\partial x} \\ 0 & \mu\frac{\partial}{\partial z} & \mu\frac{\partial}{\partial y} \end{bmatrix},$$

where t is the time, $\rho(x,y,z)$ is the density, $\lambda(x,y,z)$, $\mu(x,y,z)$ are the Lame coefficients.

It is expected that the elastic medium parameters are dependent on the three spatial variables x, y, and z.

The initial conditions are the following:

$$\vec{\sigma}|_{t=0} = 0, \quad \vec{u}|_{t=0} = 0 \tag{2}$$

and the boundary conditions at the free surface are:

$$\sigma_{xz}|_{z=0} = \sigma_{yz}|_{z=0} = \sigma_{zz}|_{z=0} = 0. \tag{3}$$

In the numerical simulation based on the solution of equations (1)–(3) it is needed to store, at least, 12 three-dimensional arrays containing information about the unknown values $\vec{u}$, $\vec{\sigma}$ and the medium parameters ρ, λ, μ in the computer memory at each time step. A large amount of memory is required for the simulation of the "big" 3D media.

This amount can be decreased by turning to the formulation of the elastodynamic problem in terms of the displacement vector $\vec{U} = (U,V,W)^T$ and thereby reducing the number of unknowns. In this case, the constitutive equations for the same simulated area are the following:

$$\rho\frac{\partial^2\vec{U}}{\partial t^2} = C\vec{U} + \vec{F}(t,x,y,z),$$

$$C = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix},$$

$$C_1 = \left[\begin{array}{ccc} (\lambda+2\mu)\frac{\partial^2}{\partial x^2} + \mu\left(\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) & (\lambda+\mu)\frac{\partial^2}{\partial x \partial y} & (\lambda+\mu)\frac{\partial^2}{\partial x \partial z} \end{array} \right],$$

$$C_2 = \left[\begin{array}{ccc} (\lambda+\mu)\frac{\partial^2}{\partial y \partial x} & (\lambda+2\mu)\frac{\partial^2}{\partial y^2} + \mu\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right) & (\lambda+\mu)\frac{\partial^2}{\partial y \partial z} \end{array} \right],$$

$$C_3 = \left[\begin{array}{ccc} (\lambda+\mu)\frac{\partial^2}{\partial z \partial x} & (\lambda+\mu)\frac{\partial^2}{\partial z \partial y} & (\lambda+2\mu)\frac{\partial^2}{\partial z^2} + \mu\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) \end{array} \right].$$

In both statements it is assumed that the right-hand side (the mass force) is the following: $\vec{F}(t,x,y,z) = F_x\vec{i} + F_y\vec{j} + F_z\vec{k}$, where $\vec{i},\vec{j},\vec{k}$ are the unit direction vectors of the coordinate axes.

3. The method of solving the problem

The most "flexible" and widespread technique is a finite difference method in the case of a three-dimensional elastodynamic problem. In this study, in order to numerically solve equations (1)–(3) we apply the well-known Verrier finite difference scheme on a staggered grid [3–5]. The calculation of its difference coefficients uses integral conservation laws. The method is of second order of approximation with respect to time and space [3]. We consider only uniform grids. As an example, let us present a few finite difference equations of the scheme used:

$$\frac{\rho_{i,j,k} + \rho_{i-1,j,k}}{2} \frac{u^{n+1}_{i-1/2,j,k} - u^{n}_{i-1/2,j,k}}{\tau} =$$
$$\frac{\sigma^{n+1/2}_{xxi,j,k} - \sigma^{n+1/2}_{xxi-1,j,k}}{h} + \frac{\sigma^{n+1/2}_{xyi-1/2,j+1/2,k} - \sigma^{n+1/2}_{xyi-1/2,j-1/2,k}}{h} +$$
$$\frac{\sigma^{n+1/2}_{xzi-1/2,j,k+1/2} - \sigma^{n+1/2}_{xzi-1/2,j,k-1/2}}{h} + f^{n}_{xi,j,k},$$
$$\frac{\sigma^{n+1/2}_{xzi-1/2,j,k-1/2} - \sigma^{n-1/2}_{xzi-1/2,j,k-1/2}}{\tau} =$$
$$M^{(1)}_{i-1/2,j,k-1/2}\left(\frac{u^{n}_{i-1/2,j,k} - u^{n}_{i-1/2,j,k-1}}{h} + \frac{w^{n}_{i,j,k-1/2} - w^{n}_{i-1,j,k-1/2}}{h}\right),$$

where

$$M^{(1)}_{i-1/2,j,k-1/2} = \left(\frac{1}{4}\left(\frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i-1,j,k}} + \frac{1}{\mu_{i,j,k-1}} + \frac{1}{\mu_{i-1,j,k-1}}\right)\right)^{-1}.$$

To solve the problem in terms of displacements, we use a similar finite difference scheme on staggered grids. The finite difference equation for the component U is as follows:

$$\frac{\rho_{i,j,k}+\rho_{i-1,j,k}}{2}\frac{U^{n+1}_{i-1/2,j,k}-2U^{n}_{i-1/2,j,k}+U^{n-1}_{i-1/2,j,k}}{\tau^2}=$$
$$(\lambda_{i,j,k}+2\mu_{i,j,k})\frac{U^{n}_{i+1/2,j,k}-U^{n}_{i-1/2,j,k}}{h^2}+$$
$$\lambda_{i,j,k}\left(\frac{V^{n}_{i,j+1/2,k}-V^{n}_{i,j-1/2,k}}{h^2}+\frac{W^{n}_{i,j,k+1/2}-W^{n}_{i,j,k-1/2}}{h^2}\right)-$$
$$(\lambda_{i-1,j,k}+2\mu_{i-1,j,k})\frac{U^{n}_{i-1/2,j,k}-U^{n}_{i-\frac{3}{2},j,k}}{h^2}-$$
$$\lambda_{i-1,j,k}\left(\frac{V^{n}_{i-1,j+1/2,k}-V^{n}_{i-1,j-1/2,k}}{h^2}+\frac{W^{n}_{i-1,j,k+1/2}-W^{n}_{i-1,j,k-1/2}}{h^2}\right)+$$
$$M^{(2)}_{i-1/2,j+1/2,k}\left(\frac{U^{n}_{i-1/2,j+1,k}-U^{n}_{i-1/2,j,k}}{h^2}+\frac{V^{n}_{i,j+1/2,k}-V^{n}_{i-1,j+1/2,k}}{h^2}\right)-$$
$$M^{(2)}_{i-1/2,j-1/2,k}\left(\frac{U^{n}_{i-1/2,j,k}-U^{n}_{i-1/2,j-1,k}}{h^2}+\frac{V^{n}_{i,j-1/2,k}-V^{n}_{i-1,j-1/2,k}}{h^2}\right)+$$
$$M^{(1)}_{i-1/2,j,k+1/2}\left(\frac{U^{n}_{i-1/2,j,k+1}-U^{n}_{i-1/2,j,k}}{h^2}+\frac{W^{n}_{i,j,k+1/2}-W^{n}_{i-1,j,k+1/2}}{h^2}\right)-$$
$$M^{(1)}_{i-1/2,j,k-1/2}\left(\frac{U^{n}_{i-1/2,j,k}-U^{n}_{i-1/2,j,k-1}}{h^2}+\frac{W^{n}_{i,j,k-1/2}-W^{n}_{i-1,j,k-1/2}}{h^2}\right),$$

where

$$M^{(2)}_{i-1/2,j-1/2,k}=\left(\frac{1}{4}\left(\frac{1}{\mu_{i,j,k}}+\frac{1}{\mu_{i-1,j,k}}+\frac{1}{\mu_{i,j-1,k}}+\frac{1}{\mu_{i-1,j-1,k}}\right)\right)^{-1}.$$

Let us note that the number of operations for the calculation of unknown values at one time step for this finite difference scheme is greater than the previous one. It should affect the calculation speed.

4. Parallel implementation and optimizations

Modern supercomputers are increasingly equipped with accelerators. This trend is also supported by the leaders of the TOP500 list. The development of a program code for such hybrid systems requires additional knowledge and time, but allows one to gain a significant increase in performance. Graphic accelerators are well suited for solving finite difference equations, because of their massively parallel architecture and easy access to the device memory. Thus, we have developed the programs with allowance for specific features of the architecture of the hybrid cluster equipped with GPU NKS-30T+GPU (the Siberian Supercomputer Center: http://www2.sscc.ru). It consists of 40 computer nodes HP SL390s G7, each one equipped with two six-core CPU Xeon X5670 and three NVIDIA Tesla M2090 graphics cards on the

Fermi architecture. Each card contains 1 GPU with 512 cores and 6 Gb of RAM GDDR5. Totally, NKS-30T+GPU contains 80 CPU (480 cores) and 120 GPU (61440 cores). Its peak performance is 85 TFLOPS.

The programs written in the programming language C++ with CUDA (Compute Unified Device Architecture) and MPI (Message Passing Interface), which make possible to use simultaneously a large number of parallel processes and ultimately to attain a maximum efficiency.

The efficient use of a hybrid architecture requires parallelization and optimizing the simulation algorithms that are based on the knowledge of the architecture of a cluster, its components, and appropriate program facilities. Since the methods of solving the compared statements of the elastodynamic problem are similar in their nature, the comparison of the correctness, adaptation and optimization of these methods is carried out in a similar way.

For the parallelization we decompose the computational domain in layers along one of the coordinate axes (Figure 1). Each layer is calculated at a separate node, where, in turn, it is sub-divided into sub-layers along the other coordinate axis (to attain a better scaling) according to the number of graphics accelerators at a node. In such implementation, each graphics card calculates its own grid domain inside the sub-layer at each time step independent of other cards, except for the points at the interface between two adjacent domains. These points are common to each of domains, and, to continue the calculation, it is necessary to exchange information about the required values among the "neighbors". Let us note that the data for the exchange have the equal size in both approaches. For an exchange between computing nodes we use non-blocking asynchronous data transfer with MPI. The exchange between the graphics cards is carried out by means of CUDA.

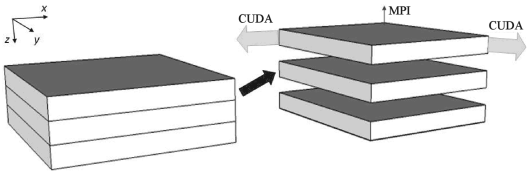

Figure 1. Decomposition of the computational domain

The effective use of graphics cards requires the memory optimization. To reduce the time for the global memory access we made the same optimal arrangement of all three-dimensional arrays used and appropriate distribution of load among threads. All the basic constants used at each time step are specially selected and stored in the constant memory of the graphics card. Although the problem to be solved operates with three-dimensional arrays, we can try to use a faster shared memory of a graphics card [6] for reducing the amount of readings from the GPU global memory. Such optimization will be performed in the future and presumably will bring greater acceleration when computing the displacement, as it uses a lesser amount of data and requires, respectively, a smaller number of copies in the shared memory.

For the computation with GPU we must set the dimension and size of a thread block. The dependence of software performance on the above parameters was studied for calculating the velocities of displacement and stress and for calculating the displacement. Thus, the calculation domain at each GPU is split to a three-dimensional grid of blocks, their size for component x must be a multiple of the length of the warp (the number of physically simultaneously executed threads per GPU). A specific size of each block is empirically selected for each algorithm. Some results of measurements are presented in the table. Here $B_x \times B_y \times B_z$ is thread block size; T_1 is the calculation time of velocities of the displacement and stress for a spatial grid of $500 \times 500 \times 600$ size and 1000 time steps; and T_2 is the calculation time of displacements for a spatial grid of $600 \times 600 \times 600$ size and 1000 time steps.

The dependence of the software speed on the thread block sizes

$B_x \times B_y \times B_z$	T_1, s	T_2, s	$B_x \times B_y \times B_z$	T_1, s	T_2, s
$4 \times 4 \times 4$	789.4		$32 \times 4 \times 4$	290.9	287.5
$8 \times 8 \times 8$	506.3	422.8	$64 \times 2 \times 2$	272.9	260.1
$2 \times 16 \times 2$	1429.2		$64 \times 4 \times 2$	273.8	
$16 \times 2 \times 2$	352.7		$128 \times 2 \times 2$	264.0	258.4
$16 \times 4 \times 4$	358.5	317.3	$128 \times 1 \times 1$	277.2	258.1
$4 \times 32 \times 4$		629.2	$256 \times 1 \times 1$	269.9	257.7
$32 \times 2 \times 2$	290.1		$512 \times 1 \times 1$	300.1	256.9

Such optimization (uncomplicated in the context of the code change) allows accelerating the programs run by several times with the effective use of the graphics accelerator global memory. We expect our conclusions to be generalized to the implementation of all finite difference methods.

Also, let us note that in both above-discussed implementations the source grid coefficients $\lambda_{i,j,k}$, $\mu_{i,j,k}$ and $\rho_{i,j,k}$ are not stored in the GPU memory, but their modifications used in the calculation scheme at each time step are stored for eliminating re-calculation. Thus, the implementation of calculating the displacement and stress velocities requires the allocation of memory to store 26 three-dimensional arrays, and the implementation of calculating the displacement requires only 14 arrays. This reduces the amount of memory used almost at a twofold rate, which is very important for calculations on computer systems that contain a small number of nodes.

5. Comparing the efficiency of the parallel implementations

In order to analyze the efficiency of the parallel implementations of two 3D statements of elastodynamic problems, we have studied their scalabilities and have compared the computing times of the media with equal sizes and parameters.

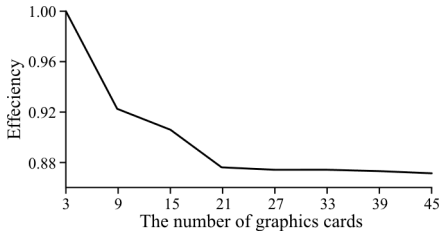

Figure 2. A weak scalability graph for the algorithm of displacement calculation

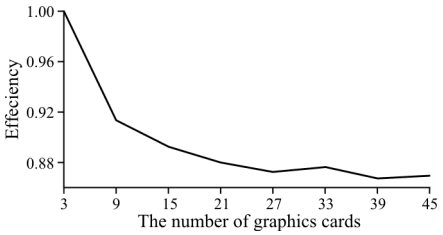

Figure 3. A weak scalability graph for the algorithm of velocities of displacement and stress calculation

By the weak scalability we understand the preservation of the calculation time of one step of the same volume of the problem when the number of graphics cards increases. The results are shown in Figures 2 and 3. In this case the efficiency means the ratio of the calculation time at n nodes of a problem that is n times greater to the calculation time at a single node of the initial problem. As in the case of displacements as well as in the case of the velocities of displacement and stress the efficiency drops to the 87 % level and slightly varies around this value with increasing the number of graphics cards up to 45 (15 nods of NKS-30T+GPU cluster). A similar behavior of the computing time for both algorithms is explained by the same number of data transferred among computing nodes ($3N^2$ for N^3 problem).

For the comparison of the running speed of both approaches we have carried out calculations for the same media with a spatial grid of $1500 \times 700 \times 2100$ size and 1000 time steps. This grid is close to the maximum grid placed in the memory of 45 graphics cards (15 nodes of NKS-30T+GPU cluster) in the calculation of velocities of displacement and stress. In the calculation of displacements on the same spatial grid one uses almost half as much memory size and can carry out at least 8 nodes instead of 15. The measurements results are as follows:

- the velocities of displacement and stress calculation — 183.1 s,
- the displacement calculation with 15 nodes — 174.8 s,
- the displacement calculation with 8 nodes — 247.4 s.

Calculations of the displacements and the velocities of displacement and stress with an equal number of nodes take roughly the same amount of time (the calculation of displacement runs a little faster). In the case of calculating the displacements, we can reduce the number of nodes approximately by the factor of 2 with preservation of the computational grid size. In this case, the calculation time increases approximately by 1.4 times.

The conducted numerical experiments show that the approach based on the calculation of displacements is faster and thus allows one to carry out

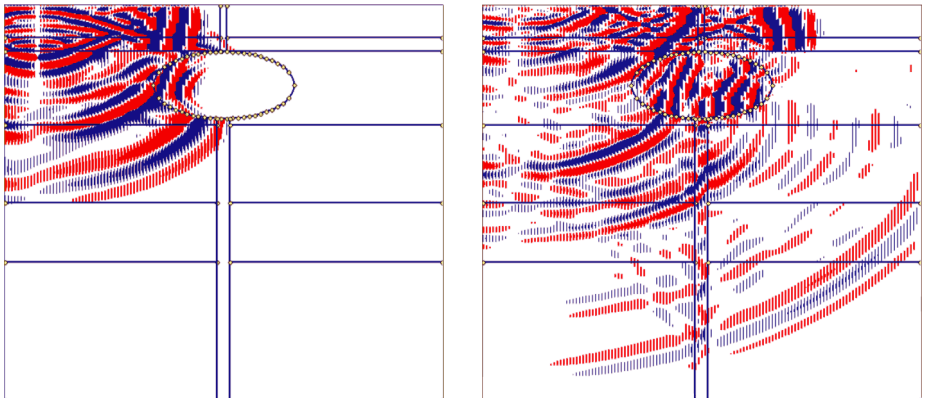

Figure 4. Results of numerical simulation for the rough model of the Elbrus volcano. In the snapshots of the wave field, the component u of the velocities displacement vector is presented in the plane Oxz at different time points

calculations for very large grids, requesting a fewer number of free nodes. This allows a quick access in conditions of a queue on the cluster, while providing a reasonable calculation time (several hours for a full-scale actual problem).

Let us note that for decreasing the number of required graphics accelerators to calculate the velocities of displacement and stress it is possible to create a program implementation of the storage of intermediate calculations in the node memory. This would require the implementation of multiple coping the sub-arrays with the desired unknowns and parameters of the medium from the host memory to the memory device and back at each time step. This would result in a significantly greater increase in the calculation time than when using the mathematical statement in terms of displacements.

To illustrate the efficiency of the software developed we have carried out the numerical simulation of the elastic wave propagation for a rough model of the Elbrus volcano (Figure 4). One can learn more about the geophysical model of the volcano and the results of numerical experiments, in [7, 8].

6. Conclusion

As part of the co-design methodology we have compared of the developed efficient parallel implementations of solutions to the elastodynamic problem written in terms of the velocity of displacement and stress and in terms of displacements for the hybrid supercomputer, equipped with graphics cards. We have studied the characteristic time needed for running the created parallel programs and their scalability.

Based on the results obtained we can give recommendations about the preference of using the approach proposed to calculating the displacements.

Although this approach gives a small gain in time as compared with the approach based on the calculation of the velocities of displacement and stress, it allows one to solve large 3D dynamic problems of the elastodynamic theory by a significantly smaller number of graphics accelerators within a reasonable time.

References

[1] Glinskiy B.M., Kulikov I.M., Snytnikov A.V., et al. Co-design of parallel numerical methods for plasma physics and astrophysics // Supercomputing Frontiers and Innovations. — 2014. — Vol. 1, No. 3. — P. 88–98.

[2] Kulikov I.M. GPUPEGAS: a new GPU-accelerated hydrodynamic code for numerical simulations of interacting Galaxies // The Astrophysical Journal Supplement Series. — 2014. — Vol. 214, No. 12.

[3] Bihn M., Weiland T.A. Stable discretization scheme for the simulation of elastic waves // Proc. 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics (IMACS 1997). Berlin. — 1997. — Vol. 2. — P. 75–80.

[4] Karavaev D.A. Parallel implementation of wave field numerical modeling method in 3D models of inhomogeneous media // Vestnik of Lobachevsky State University of Nizhni Novgorod. — 2009. — No. 6(1). — P. 203–209 (In Russian).

[5] Glinskiy B.M., Karavaev D.A., Kovalevskiy V.V., Martynov V.N. Numerical modeling and experimental research of the "Karabetov Mountain" mud volcano by vibroseismic methods // Numerical Methods and Programming. — 2010. — Vol. 11. — P. 95–104 (In Russian).

[6] Nakata N., Tsuji T., Matsuoka T. Acceleration of computation speed for elastic wave simulation using a Graphic Processing Unit // Exploration Geophysics. — 2011. — Vol. 42. — P. 98–104.

[7] Glinskiy B.M., Martynov V.N., Sapetina A.F. Technology of supercomputer simulation of seismic wave fields in complicated media // Bull. the South Ural State University. Ser.: Computational Mathematics and Software Engineering. — 2015. — Vol. 4, No. 4. — P. 101–116 (In Russian).

[8] Glinskiy B.M., Martynov V.N., Sapetina A.F. 3D Modeling of Seismic Wave Fields in a Medium Specific to Volcanic Structures // Mathematical notes of NEFU. — 2015. — Vol 3(87), No. 22. — P. 84–98 (In Russian).