A Technology of 3D Elastic Wave Propagation Simulation Using Hybrid Supercomputers *

D.A. Karavaev, B.M. Glinsky, V.V. Kovalevsky

Institute of Computational Mathematics and Mathematical Geophysics of SB RAS

We present a technology of 3D seismic field simulation for high-performance computing systems with GPUs or Intel Xeon Phi coprocessors. This technology covers adaptation of a mathematical modeling method and development of a parallel algorithm. We describe the parallel realization designed for simulation based on using staggered-grids and 3D domain decomposition method. We study the parallel algorithm behavior on computing devices: CPUs, GPUs, Xeon Phi coprocessors. We consider the results of experiments that were carried out on cluster NKS-30T of SSCC and cluster MVS-10P of JSCC for different tests for parallel algorithm.

1. Introduction

Simulation of 3D seismic wave propagation for a large-scale geophysical medium is very important for developing geophysical models, studying the effects of seismic field, comparing the research and model data [1]. Sometimes it is difficult to solve inverse geophysical problem to study the object characteristics and parameters because of complexity of geometry of free surface under study. One of the methods for solving inverse problems is solving the forward problem for a various number of models. Thus, carrying out the simulation, varying elastic parameters and establishing the correspondence with natural geophysical data, one can find a more appropriate geometrical structure and elastic parameters values of a geophysical object under investigation. Now days the most useful and popular numerical modeling methods is based on using finite differences and 3D meshes [2]. The most useful difference methods can be a second or a fourth order of approximation [3–5] and have application in modeling elastic or viscoelastic media [6]. In such a case we can calculate the full seismic field and know the values of elastic wave parameters at each point of mesh. But realistic simulations rely on models with detailed representation and so demand intensive computations with large amounts of 3D data. For example a 3D grid model of 3D isotropic geophysical media is described with the three parameters: density and two velocities of elastic waves. When using the explicit 3D finite difference scheme with the iterative technique one should mind that you need to work with data placed at different time steps. Therefore, we need to place in operating memory of the computing device a large volumes of 3D data that describe 3D mesh model of geophysical object and 3D arrays for variables under study and at different iteration steps. In this context using high performance computing systems can help us to solve forward modeling problem and to deal with a hundreds of GBs of data and to achieve modeling results in a reasonable amount of time. Modern vendors of computing systems offer a so called hybrid systems. The architecture of such systems is difficult enough and differs from CPU based cluster. The hybrid cluster may include computing nodes with several multi-core CPUs and several computing devices that can be either Nvidia GPU cards or Intel Xeon Phi coprocessors. These special computing devices have tens and thousand of computing cores and operating memory that is larger than CPU have. So GPUs and Xeon Phi coprocessors deliver high computing performance. Modern cluster systems that are in the first places in TOP 500 rating have hybrid architecture. Some examples of hybrid clusters that we deal with in Russia are NKS-30T+GPU cluster of the Siberian Supercomputer Center and MVS-10P cluster of the Joint Supercomputer Center. With the use of such computing systems, one can solve large 3D models in parallel manner and accelerate modeling algorithms. Therefore

*This work was supported in part by NS-5666.2014.5 and the RFBR grants No. 13-07-00589, 14-07-00832, 14-05-00867, 15-07-06821, 15-31-20150

to use hybrid systems for computations we need to develop a scalable parallel algorithm and a special program code. And it is a programmable and a researcher's problem. Because we want to solve large-scale 3D problems and use in computations thousands of computing cores of modern architectures. We describe in this paper the technology for the 3D seismic field simulation developed for computations on hybrid supercomputers. This technology includes a modification of a difference method, parallel algorithm, computational schema realized in program codes. There are different program codes realizing methods for the seismic wave propagation modeling on clusters with GPUs [7–11]. In our work we present the results of applying the developed technology for the CPUs, GPUs and the Intel Xeon Phi coprocessors in simulation. It is a new and modern approach for parallel computation. Such systems can allow researchers using OpenMP parallel tools to develop programs for large-scale simulation. Our main goals were to develop the technology and to study the behavior of developed parallel algorithm on different architectures and for different tests for parallel program code. The paper proceeds as follows. Section 2 discusses the fundamentals of seismic wave propagation for 3D case: description of the problem statement and numerical method. Section 3 gives a brief review of hybrid high-performance computing systems related to this work. In Section 4 we describe parallel implementation of algorithm in detail. Section 5 presents computational experiments for different test of a parallel code for a GPU based cluster. In section 6 we discuss the implementation of the simulation code for Xeon Phi based cluster and for the single coprocessor case and for the multi-device case. Section 7 describes related work and concludes the main results.

2. Problem Statement

Solving the forward geophysical problem is connected with solving system of equations of elastic theory modified for 3D case. In this paper we discuss method for elastic wave propagation simulation applicable for isotropic and elastic materials. Thus, the 3D grid model of a geophysical medium under study is described by three elastic parameters: density, shear wave velocity and longitudinal wave velocity. In the difference scheme, we deal with three material parameters: Lame coefficients and density. We work only with geometries of elastic media that have plane free surfaces, but can have a difficult geometrical structure and different values of elastic parameters at each point of 3D grid. A problem of the 3D elastic wave propagation is described in terms of components of velocities of displacements $\mathbf{u} = (U, V, W)^T$ and components of a stress tensor $\sigma = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz})^T$. The problem is to be solved with appropriate zero initial conditions and zero boundary values. We apply a free-surface condition at the top boundary. We use the Cartesian coordinate system. To numerically solve the simulation problem, we use the difference method [4], based on using staggered grids. The difference scheme is of second order of approximation with respect to time and space. The government equations of difference scheme will be in the form of (1).

$$\rho \frac{\partial \mathbf{u}}{\partial t} = [A]\sigma + \mathbf{F}(t, x, y, z), \frac{\partial \sigma}{\partial t} = [B]\mathbf{u}; \tag{1}$$

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

In the developed programs, we use modified coefficients. This means that coefficients in the difference scheme include all the summations and multiplications and have a presentation that is easier for computations. Thus we have nine components of wave field and eight material coefficients that are assigned to a unit cell at time step. In such a case, we have to deal with large 3D arrays and have to use more operating memory of computing system and can perform computations.

3. Hybrid Clusters

Watching the TOP-500 supercomputer list one can find that some of the powerful computing systems are designed with help of special computing devices and in so-called hybrid manner. These computing devices differ from the common used CPUs and are developed to make extremely fast computations. They are presented in two variants. It can be GPU computing device or Intel Xeon Phi coprocessor. Each of them has their advantages and can be used for large-scale numerical modeling. In case of using GPUs for computations we need to develop program code written with CUDA technology. Sometimes it's difficult for developers to rewrite code to deal with grid of blocks and parallel threads to run a procedure on GPU. Using Intel Xeon Phi for computations is second approach. Developing a program code for such architecture looks like something developing code for SMP and CPU based machine. Because Intel Xeon Phi based is based on x86 architecture but it's not it. Thus you need to recompile your code for the Intel MIC architecture. The work in this paper is oriented on using two supercomputers for large-scale 3D simulation. First is NKS-30T cluster of the Siberian Supercomputer Center. This multi-purpose cluster is useful for solving different tasks by different scientific and educational organizations especially of SB RAS. It is consists of 40 SL390s G7 servers. Each of them has two 6-core Xeon X5670 CPU and three video cards NVIDIA Tesla M 2090 with Fermi architecture. Each video card has 1 GPU with 512 cores and 6GB GDDR5. It is a hybrid part of the cluster. Also cluster has computing nodes based on Xeon CPUs of different series: Intel Xeon E5540, Intel Xeon E5450. The second cluster is MVS-10P. This supercomputer is placed at Joint Supercomputer Center of RAS. JSCC RAS is the most powerful supercomputer center in Russia in science and education. MVS-10P cluster consists of 207 computing nodes. Each of them has two Xeon E5-2690 processors and two Intel Xeon Phi 7110X coprocessors.

Intel Xeon Phi coprocessor is something looks like an SMP machine. Each of above mentioned coprocessors has 60 computing cores with 8 GB RAM and 4 threads per core and 240 threads per machine. Working with such a system one can use coprocessors in several modes: native, offload, symmetrical. Our approach is based on using above mentioned computing devices in offload mode. These means that all the computations take place on devices, CPUs is not used in computations. All the information to perform computations is initialized or is copied on device.

We have adopted numerical method, designed parallel algorithm and parallel scheme for computations to work with hybrid high-performance systems and to conduct calculations on it. Using this information, we have developed program codes for such systems that solve problem of elastic wave propagation in 3D case.

4. Parallel Implementation

We consider hybrid parallel implementation, using both GPUs and Intel Xeon Phi coprocessors for computation. Our technology includes a multiple device realization. Firstly it was developed for a multi-GPU system and was oriented for a NKS-30T+GPU cluster. Knowing the fact that only two of three GPUs placed on idle computing node allow P2P exchanges we designed our parallel schema not using this option. So we use computing devices in so called offload mode. All the calculations are performed on devices and the CPUs is used for managing among devices and data exchanges.

To solve large-scale problems and to compute fine-grain grid 3D models we apply 3D topology that has data distribution character Fig. 1. We divide initial large model into smaller 3D subdomains in such a case that to place data in device operating memory. All the computations are performed with help of GPUs or Intel Xeon Phi coprocessors. Along the spatial directions the decomposition is performed using MPI tools, and further decomposition for every subdomain is performed with OpenMP tools or CUDA. The OpenMP tools is used for parallel implementation of the code on MIC architecture. To manage them and perform data exchanges between neighbor computing devices we use CPUs. Under our realization the one computing device is managed by one CPU core that has one MPI process run on it. For a GPU based realization we apply CUDA technology for parallel computations. It allows creating a grid of threads and parallelizing computing cycles depending from the number of thread running on GPU cores. For the NKS-30T cluster we can create up to 1024 parallel threads and use up to 512 cores. In GPU program code we work on cards only with global memory. According to parallel algorithm we need to exchange data in the ghost zones between the neighboring subdomains. This fact especially important if we work with devices in offload mode and perform data exchanges at every iteration step. As direct communication between Xeon Phi coprocessors is not available and we cannot realize similar communication operations among all GPU cards we realize a special approach. Also we need data exchanges between devices placed on different computing nodes of cluster. To use all the possibilities of MPI and CUDA and OpenMP we create two types of computing procedures. Such modification allows us to overlap the procedures for the communication and the computation to reduce the total computing time.

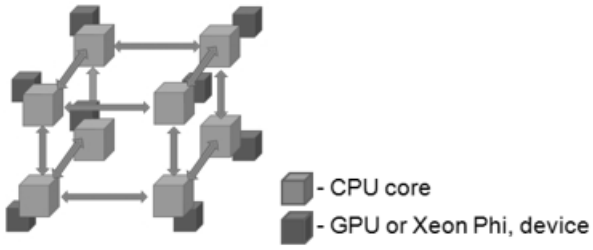

Fig. 1: Schematic illustration of the 3D domain decomposition.

In our realization we make calculations at first for grid points on the sides of subdomains for ghost point exchange. Then we copy data from Xeon Phi's into CPU and run exchange procedures with use of MPI functions and special designed buffers. We do the computation for the remaining internal grid points of 3D subdomains and the communication procedures simultaneously. We verify whether all the exchanges have been done and then copy data from buffers placed at CPUs into buffers at Xeon Phi coprocessors and after that into 3D arrays. Then we proceed to the next time step. Therefore, we overlap the communication and computation by using the non-blocking MPI Send/Receive procedures for data transfer Fig. 2. We do the data transfer between the CPUs at nodes concurrent with the computations at devices.

Using Intel Xeon Phi may be more easily in comparison with CUDA. Because in program code we use OpenMP for parallel computations for Intel Xeon Phi instead of CUDA technology. We developed program codes using the above mentioned technology for 3D seismic field numerical modeling. As we have mentioned above we realized two types of procedures. One is for 2D data and other is for 3D data.

All the program codes were written using C++ language. When using the developed technology for parallel computations we can perform calculation on a single device or on cluster. In further section we compare the behavior of parallel algorithm for systems with different types of computing devices: CPUs, GPUs, Xeon Phi coprocessors. We developed program codes with MPI for CPUs, MPI with CUDA for GPUs and MPI with OpenMP.

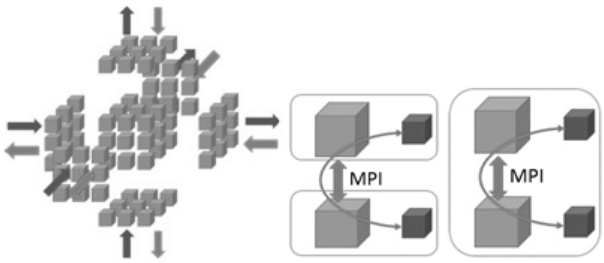

Fig. 2: Realization of a parallel computations.

5. Studying the Behavior of the Parallel Algorithm on GPU Cluster

When developing parallel algorithm one should investigate the behavior of parallel algorithm and a program code on different tests: scalability test and speed-up test. In the first case, we watch the algorithm behavior under consideration that computing area is scaling proportional to number of computing devices, but the time for computations should not vary strongly. In the second case, we fix the number of points in a 3D mesh and we watch the speedup of computations when varying the number of computing devices. In this section we also compare the work of program codes for simulation written to use CPUs or GPUs. The results are obtained on the NKS-30T+GPU cluster of SSCC. The computational time of staggered-grid difference method depends on only the grid size of a 3D mesh and it is independent on geology structure complexity. To compare the computation time for GPU accelerators and multi-core CPUs, we performed the following tests in which the computation time was recorded. The programs were compiled without any optimization options. The performance of the developed codes is shown in Fig. 3 and Fig. 4.The time of calculations with GPUs is ten times greater (gpu(x10)) in the figures.

The results of scalability tests presented at figure show the well-done program behavior. When we scale the model size the program shows a good behavior. So when we do a scaling of mesh and when we increase the number of devices the time of calculations is almost the same on test with CPUs and on test with GPUs. From Fig. 3 we can see that the program was effectively parallelized and the ratio of results on GPU to results on CPU is 65 times in average.

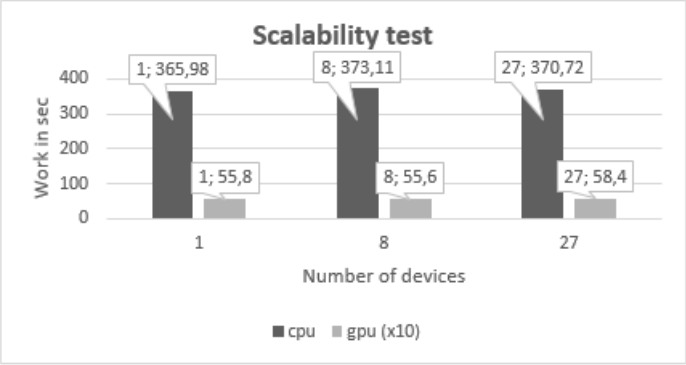

Fig. 3: A scalability test.

The results of the speed-up tests are presented in the Fig. 4. Figure 4 reveals that the ration of CPU/GPU is about x39 on one device and x31 on the other 27 devices. The ratio of 8 to 27 devices for GPU is about x3.3.

A mesh of size 308^3 was used for scalability test. Each GPU accelerator contains a portion of the whole mesh corresponding to 308^3. All the computations were performed using 11 iterations. To run CUDA threads we use 512 cores of GPU and up to 1024 threads. For 2D calculations, we use grid blocs size of 32x32 and size of 8x8x8 for 3D calculations.

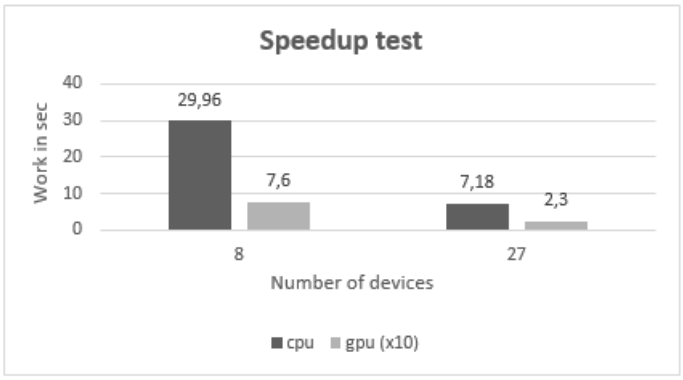

Fig. 4: A speedup test.

6. Studying the Behavior of the Parallel Algorithm on Xeon Phi Cluster

In this section, we present the results of the parallel algorithm behavior for a hybrid cluster architecture with Intel Xeon Phi coprocessor. We have carried out experiments on one computing device to choose appropriate options for large 3D models. We made a comparison of programs running on different computing devices for calculations when using only CPUs or only Xeon Phi coprocessors.

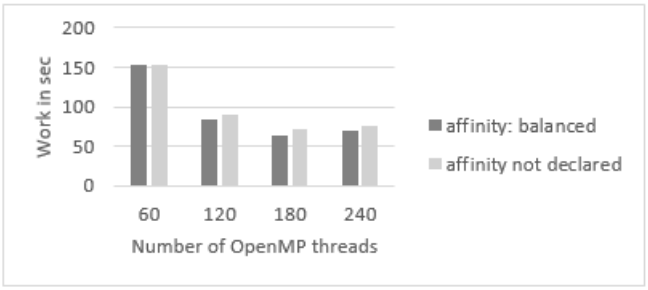

Fig. 5: A test with affinity option and the number of parallel threads on one device.

When tuning the application on one device, we have carried out experiments with affinity option and with different number of threads per core. We worked with 3D mesh with 308^3 grid points and 11 iterations. The affinity option has been taken in two versions: «not declared» or «balanced». The results of such a research is presented at Fig. 5. The most appropriate is the «balanced» option and using 60 cores with a 3 threads per core.

The results of scalability test presented at Fig. 6 show the well-done program behavior. When we scale a 3D model as great as 2-fold along each spatial coordinate and the number of devices as great as 8-fold, the program shows a good behavior. In these tests, we use a 3D subdomain size with 308^3 grid points and 11 iteration steps for one device. In this case we have used 2 x 2 x 2 grid of computing devices. From Fig. 6 we can see that the program was effectively parallelized and the ratio of CPU/Xeon Phi is about x5.7.

The results of the speed-up tests presented in the Fig. 7 show the program behavior with 308^3 grid points and 11 iterations for all devices. It is shown that ration of CPU/Xeon Phi is about x5.7 on one coprocessor and x3,6 other eight coprocessors. The ratio of 1 to 8 devices for Xeon Phi is about x7.7 on eight devices.

The obtained results give experience in applying the developed technology for using Intel Xeon Phi coprocessors in computations. Maybe obtained values showing the acceleration are not

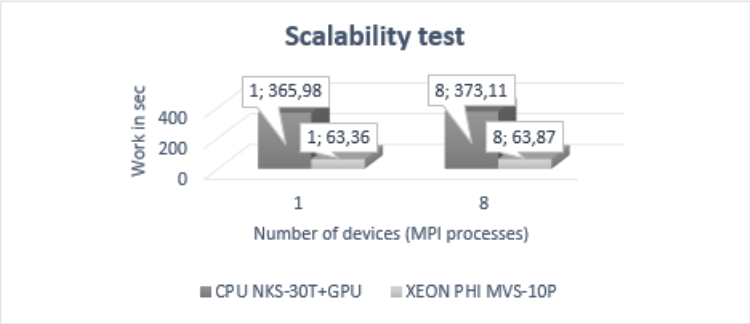

Fig. 6: A scalability test.

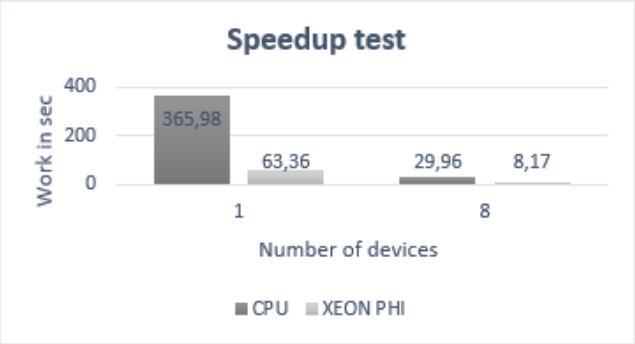

Fig. 7: A speedup test.

big enough but we showed it in first experiments. The investigation and modification of program code and algorithm may be done with use of vector operations and with study of other tuning parameters.

7. Conclusion

We proposed the technology for realistic 3D full seismic field simulation in elastic medium for isotropic case. This technology covers solving different problems as one large problem. It include developing parallel algorithm, choosing the domain decomposition approach to place large volumes of 3D data into computing nodes memory, developing parallel computation schema oriented on high performance computations dealing with great amount of computing cores of hybrid clusters. We presented the results of the research into developing a scalable parallel algorithm and software for hybrid systems with different architectures of computing devices. We proposed new software for simulation on supercomputers with Intel Xeon Phi coprocessors. We described the parallel implementation of the difference method based on using computing devices as accelerators in offload mode. We have carried out computing experiments and investigated the behavior of the scalable parallel algorithm for different tests. We presented the first results of applying the developed technology to use Intel Xeon Phi coprocessors in 3D simulation of seismic field. The results of the research done are important and can be of practical use in the field of developing scalable parallel algorithms for exzaflops supercomputers [12] of the future and modeling its behavior on a greater number of computing cores in simulation systems. It is shown that the 3D difference method with staggered grids can be well parallelized for Nvidia GPU and Intel MIC architecture. We can use the discussed computing devices to simulate big size models. Based on the above-mentioned results, we conclude that carrying out computations on Intel Xeon Phi coprocessors for the large-scale seismic field simulation is a promising approach.

References

1. Glinsky B.M., Karavaev D.A., Kovalevsky V.V., Martynov V.N: Numerical modeling and experimental research into «Karabetova Mountain» mud volcano using vibrosesmic methods // Vichislitelnie metody I Programmirovanie. Vol. 11, pp. 95–104 (in Russian)(2010)

2. R. W. Graves: Simulating seismic wave propagation in 3D elastic media using staggered grid finite differences. // Bull. Seism. soc. Am., vol. 86, pp. 1091—1106 (1996)

3. A.R. Levander Fourth-order finite-difference P-SV seismograms. // Geophysics, vol. 53, issue 11, pp. 1425—1436 (1988)

4. Virieux J. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. // Geophysics, Volume 51,Number 4, pp. 889–901 (1986)

5. Moczo, P., Kristek, J. and Halada, L.: 3D fourth-order staggered-grid finite difference schemes stability and grid dispersion // Bull. Seism. Soc. Am., Vol. 90, No. 3, pp. 587–603 (2000)

6. J.O. A, Robertsson, J.O. Blanch, and W.W. Symes Viscoelastic finite-difference modeling // Geophysics, vol. 59, issue 9, pp. 1444—1456 (1994)

7. Dimitri Komatitsch Fluid-solid coupling on a cluster of GPU graphics cards for seismic wave propagation // Comptes Rendus Mecanique, Volume 339, Issues 2-3, pp. 125–135 (2011)

8. Dimitri Komatitsch / Dimitri Komatitsch, Gordon Erlebacher, Dominik Goddeke, David Michea / High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster // Journal of Computational Physics, Volume 229, Issue 20, pp. 7692–7714 (2010)

9. D. Michea, D. Komatitisch Accelerating a three-dimensional finite-difference Wave Propogation Code Using GPU Grapics Cards // Geophys. J. Int. 182(1), pp. 389–402 (2010)

10. T. Okamoto,H. Takenaka, T. Nakamura, and T. Aoki."Accelerating large-scale simulation of seismic wave propagation by multi-GPUs and three-dimensional domain decomposition", Earth Planets and Space, 62, pp.939–942, (2010)

11. Micikevicius, P. 3D finite-difference computation on GPUs using CUDA // in GPGPU-2: Proc. 2nd Workshop on General Purpose Processing on Graphics Processing Units, Washington DC, USA, pp. 79—84 (2009).

12. Chernykh I., Glinskiy B., Kulikov I., Marchenko M., Rodionov A., Podkorytov D., Karavaev D. Using Simulation System AGNES for Modeling Execution of Parallel Algorithms on Supercomputers // Computers, Automatic Control, Signal Processing and Systems Science. The 2014 Int. Conf. on Applied Mathematics and Computational Methods in Engineering, pp. 66–70 (2014)