Parallel Algorithm for One-way Wave Equation Based Migration for Seismic Imaging

Alexander Pleshkevich[1], Dmitry Vishnevsky[2], Vadim Lisitsa[3]✉, and Vadim Levchenko[4]

[1] Central Geophysics Expedition
Moscow, Russia psdm3d@yandex.ru
[2] Institute of Petroleum Geology and Geophysics SB RAS
Novosibirsk, Russia vishnevskydm@ipgg.sbras.ru
[3] Institute of Petroleum Geology and Geophysics SB RAS
Novosibirsk State University
Novosibirsk, Russia, lisitsavv@ipgg.sbras.ru
[4] Institute of Applied Mathematics RAS
Moscow, Russia vadimlevchenko@mail.ru

Abstract. Seismic imaging is the final stage of the seismic processing allowing to reconstruct the internal subsurface structure. This procedure is one of the most time consuming and it requires huge computational resources to get high-quality amplitude-preserving images. In this paper, we present a parallel algorithm of seismic imaging, based on the solution of the one-way wave equation. The algorithm includes parallelization of the data flow, due to the multiple sources/receivers pairs processing. Wavefield extrapolation is performed by pseudo-spectral methods and applied by qFFT - each dataset is processed by a single MPI process. Common-offset vector images are constructed using all the solutions from all datasets thus by all-to-all MPI communications.

Keywords: One-way wave equation · Pseudo-spectral methods · qFFT · CUDA · Nested OMP · MPI

1 Introduction

Modern development of the supercomputers with hybrid architecture, especially use of GPUs, leads to the revision of the numerical methods and algorithms which were computationally inefficient and hard to implement using conventional cluster architecture. Seismic imaging is the procedure to reconstruct the interior structure of the subsurface which is based on the correlation of solutions of direct and adjoint wave propagation problems. Traditionally three types of approaches are considered. The first one is based on the asymptotic solution of scalar or elastic wave equation; i.e. based on the ray theory [1]. The advantage of this technique is the numerical efficiency of the solution construction, possibility to store and combine a high number of ray paths and travel times, thus selective images for different source-receiver offsets and azimuth as well as for different inclination angles of the reflecting surface can be constructed. In addition, ray tracing in models with local velocity decrease or in anisotropic media where triplication of the wavefield may appear is troublesome. The second group of methods is based on the full waveform simulation; i.e. on the solution of the wave equation for all positions of the sources and receivers [2], [15], [14]. This group is antipode to the first one, it is extremely computationally intense but provides complete information about the wavefield. The third group includes methods based on the solution of the one-wave wave equation (OWE) to generate the solution of direct and adjoint problems [4], [5], [13], [3], [17], [7], [12]. This approach is dealing with the pseudo-differential operator possessing solutions propagating in the positive direction with respect to depth. OWE based techniques are more computationally efficient than full-waveform simulations but free from the drawbacks of the ray-based methods. However, use of conventional CPU architecture made these type of approaches unattractive, because they still required rather intense computations, but nowadays with intense part can be done using GPUs, moreover, amount of available memory (either on device and host) makes it possible to solve OWEs for a large number of right-hand sides forming images for different source-receiver offsets and azimuths. In this paper, we present an algorithm implementing the third type of methods. We use the pseudo-spectral method, based on the intense use of FFT procedures, to solve OWE, thus we achieve high performance by implementing CUDA technology and qFFT.

2 Mathematical background

2.1 One-way wave equation

One-way wave equation is a non-local pseudo-differential operator, governing down-going wave propagation:

$$\begin{array}{l} \frac{\partial u}{\partial z} + i\sqrt{\frac{\omega^2}{v^2(x,y,z)} + \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}}u = 0, \\ u(\omega,x,y,0) = u_0(\omega,x,y), \\ u(\omega,x,y,z) \to 0|_{x\to\pm\infty}, \\ u(\omega,x,y,z) \to 0|_{y\to\pm\infty}. \end{array} \tag{1}$$

We consider this equation defined in a domain $D = \{(x,y,z)|\ x \in R,\ y \in R,\ z \in [0, Z_1]\}$. In this notations x and y denote the coordinates in horizontal direction, z is vertical coordinate, u is a wavefield, $\omega \in [\Omega_1, \Omega_2]$ is the temporal frequency, $0 < V_1 \le v(x,y,z) \le V_2 < \infty$ is the wave propagation velocity. Throughout the paper we will assume that the velocity is smoothly varying, because dealing with discontinuous coefficients for wave propagation requires additional study and special treatment of the coefficients [9], [8], [10], [6], [16].

Our goal is to solve problem (1) inside the domain $D_0 = \{(x,y,z)|\ x \in [X_0, X_1],\ y \in [Y_0, Y_1],\ z \in [0, Z_1]\} \subseteq D$ so that the solution to be accurate for propagation direction deviating from vertical by the angle $\alpha \le 50°$.

Solution of the one-way wave equation can be computed layer-by-layer with respect to depth. In particular, we suggest using the pseudo-spectral method with the sixth order of approximation with respect to the direction of wave propagation (inclination from the vertical direction) [11], [12]. Assume the wavefield at the level $z = z_0$ is known then the solution at the level $z = z_0 + \Delta z$ is computed by the following rule:

$$\begin{gathered}\tilde{u}(\omega, x, y, z_0 + \Delta z) = \alpha_0 u(\omega, x, y, z_0)+ \\ +\alpha_j(\omega, x, y, z_0) u_j(\omega, x, y, z_0 + \Delta z) + \alpha_{j+1}(\omega, x, y, z_0) u_{j+1}(\omega, x, y, z_0 + \Delta z),\end{gathered} \tag{2}$$

where velocity in the particular point (x, y, z) belongs to an interval between two reference velocities; i.e. $v(x, y, z) = v_0 \in (v_j, v_{j+1})$. Functions u_j and u_{j+1} are the solutions of the one-way wave equation with constant velocities v_j and v_{j+1}; computed by the formulae:

$$u_j(\omega, x, y, z_0 + dz) = F^{-1}_{x,y}\left[e^{ik^j_z dz} F_{x,y}\left[u(\omega, x, y, z_0)\right]\right], \tag{3}$$

with

$$k^j_z = sign(\omega)\sqrt{\frac{\omega^2}{v_j^2} - k_x^2 - k_y^2}. \tag{4}$$

Coefficients α_j and α_{j+1} are

$$\alpha_j = \frac{v_0\left[v_{j+1}^2\left(1 - \frac{i\omega dz}{v_{j+1}}\right) - v_0^2\left(1 - \frac{i\omega dz}{v_0}\right)\right]}{v_j\left[v_{j+1}^2\left(1 - \frac{i\omega dz}{v_{j+1}}\right) - v_j^2\left(1 - \frac{i\omega dz}{v_j}\right)\right]} e^{i\frac{\omega dz}{v_0} - i\frac{\omega dz}{v_j}}, \tag{5}$$

$$\alpha_{j+1} = \frac{v_0\left[v_0^2\left(1 - \frac{i\omega dz}{v_0}\right) - v_j^2\left(1 - \frac{i\omega dz}{v_j}\right)\right]}{v_{j+1}\left[v_{j+1}^2\left(1 - \frac{i\omega dz}{v_{j+1}}\right) - v_j^2\left(1 - \frac{i\omega dz}{v_j}\right)\right]} e^{i\frac{\omega dz}{v_0} - i\frac{\omega dz}{v_{j+1}}}, \tag{6}$$

$$\alpha_0 = e^{i\frac{\omega dz}{v_0}} - \alpha_j e^{i\frac{\omega dz}{v_j}} - \alpha_{j+1} e^{i\frac{\omega dz}{v_{j+1}}}. \tag{7}$$

Thus to compute solution at a depth level $z = z_0 + \Delta z$ with variable velocity one needs to update the a set reference solutions and then construct linear combinations in each point at the plane $(x, y, z_0 + \Delta z)$.

As discussed in [12] wavefield depth extrapolation by pseudospectral method and its modifications, such as phase-shift plus interpolation approach, can be applied using very coarse discretization in vertical direction. In particular, in our algorithm we use $\Delta z = 50$m. Whereas the images are constructed on a mesh with the step as small as 5m. To interpolate the solution we use the following formulae:

$$\begin{gathered}u(x, y, z_0 + \gamma\Delta z, \omega) = \\ = (1-\gamma) u(x, y, z_0, \omega) e^{\frac{i\omega}{v(x,y,z_0)}\gamma\Delta z} + \gamma u(x, y, z_0 + \Delta z, \omega) e^{\frac{i\omega}{v(x,y,z_0)}(1-\gamma)\Delta z}\end{gathered}$$

here $\gamma \in [0, 1]$ and $v(x, y, z)$ is assumed to be independent of z within the slab $[z_0, z_0 + \Delta z]$.

2.2 Imaging condition

To construct the seismic image at a point (x, y, z) one needs to precompute the Green's functions for source and receiver positions $G(x, y, z, x_s, y_s, \omega)$ and $G(x, y, z, x_r, y_r, \omega)$ respectively, and then convolve them with the wavefield, recorded in the receiver $\phi(x_s, y_s, x_r, y_r, \omega)$; i.e.

$$\begin{gathered} I(x,y,z,x_s,y_s,x_r,y_r) = \\ = \int\limits_{\omega_0}^{\omega_1} \bar{G}(x,y,z,x_s,y_s,\omega)\phi(x_s,y_s,x_r,y_r,\omega)\bar{G}(x,y,z,x_r,y_r,\omega)d\omega, \end{gathered} \tag{8}$$

where ω_0 and ω_1 define the frequency range, and $\bar{G}$ denotes the complex conjugate of the Green's functions.

Seismic data acquisition typically includes dozens to hundreds thousands of sources and receivers, thus different images can be computed; i.e.

$$\begin{gathered} I^k(x,y,z) = \\ = \int_{(x_s,y_s)\in\Omega_s^k} \int_{(x_r,y_r)\in\Omega_r^k} I(x,y,z,x_s,y_s,x_r,y_r)dx_s dy_s dx_r dy_r, \end{gathered} \tag{9}$$

where Ω_r^k and Ω_s^k where can be defined according to the particular needs of seismic processing. Moreover, we use superscript k to denote that usually more than one image is constructed, depending on the processing procedures.

One of the most widely-used seismic images are the common-offset vector images, where the distance between source and receiver positions is fixed whereas parametrization of the image is implemented via mid-points between the sources and receivers. So, that fixed superscript k means that for all points $(x, y, 0)$ one considers pairs of sources and receivers, for which this point is the midpoint and the distance and direction between the pair is fixed, One computes images for all such pairs and then integrates over all points $(x, y, 0)$.

3 Description of the algorithm

3.1 Dataflow

Before proceeding to the description of the computational aspects of the algorithm let us consider the data flow, and estimate the typical amount of data. The input data are seismograms acquired in the field. These data are usually viewed as a 5D cube; i.e. for each source position (2D space), for each receiver position (2D) a single trace is recorded (1D). Typical acquisition system includes dozens to hundreds of thousands of sources and receivers. Let us denote the number of sources and receivers by N_s nad N_r respectively, usually $N_s \approx N_r = 10^4 - 10^5$. The length of the trace record N_t is about 10^3. Thus the volume of the input data varies from 1 TB to 100 TB.

Output data are the set of common-offset vector images. For modern acquisitions, more than 100 images are constructed. Estimate the size of the single image. Typically the domain is about 15 to 20 km in horizontal directions and

5 km in the vertical one. Using the grids steps equal to 25 m in horizontal and 5 m in vertical direction one finishes up with the size of the domain of $0.5 \cdot 10^9$ points, or 2 Gb per a single image.

One of the principal aspects of the seismic imaging is that each source/receiver is used to compute all the output images. Thus even if data can be split into several datasets, each dataset will be used to compute all images, and the best hypothetical algorithm would be able to process whole data and provide the complete set of images. However, this is not affordable by modern computers, and we suggest algorithm which operates independently with several datasets, intensively use MPI communications, and provide several images corresponding to these datasets. A sketch of the data flow is provided in figure 1.

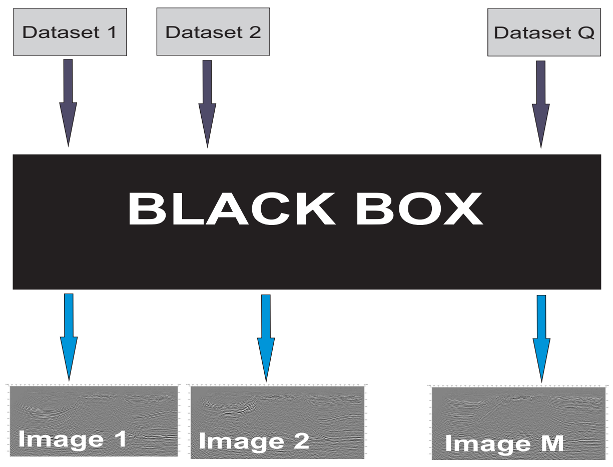

Fig. 1. A sketch of the data flow in the seismic imaging algorithm

3.2 Hypothetical sequential version

To simplify the description the parallel version of the algorithm for seismic imaging, let us consider a simple sequential version first. Assume a whole input seismic data, Green's functions for all sources and receivers positions, and all constructed seismic images can be stored in RAM. In this case, the algorithm will have the following structure. The outer loop is the integration with respect to the time frequencies. For each time-frequency we use a loop with respect to depth; i.e. we compute the Green's functions for all sources and receivers positions, combine them and multiply by the corresponding signal. Thus at fixed depth level we construct all possible images $I^k(x, y, z_l, \omega_m)$. Next we interpolate images within the slab $[z_{l-1}, z_l]$. Repeating this procedure for all frequencies and all depth layers we obtain all common-offset vector images at the same time.

However, as it follows from the estimates, presented in the previous section, the sequential version of the algorithm cannot be implemented using modern computers. Because, it requires storage of all fixed-depth 2D cross-section of the Green's functions Let us remind that the number of the sources and receivers is about 10^5, the size of single 2D cross-section of the Green's function is about 10^6 points, which leads to the estimate of 8 Mb per Green's function (single precision complex numbers). Thus storing all Green's functions would require 800 Gb, which is unacceptable. To overcome this and design parallel algorithm we suggest consider relatively small datasets, which fits the devise memory and process each dataset in parallel.

3.3 Datasets construction

Construction of the datasets can be explained on the physical aspects of the solved problem. It can be viewed as a 2D domain decomposition of the acquisition system. As mentioned above the imaging domain can be parametrized by the coordinates of the mid-points, so we apply the 2D domain decomposition and consider all the sources and receivers corresponding to the mid-points belonging to a single subdomain as a single dataset. Note, that the size of subdomains is restricted by the amount of the memory per single GPU because the main computations are done on GPUs. Assume a single have N allocated sources and receivers, thus N 2D cross-sections of the Green's functions have to be computed and stored. Due to the typical geometry of the acquisition system, the size of the subdomain rarely exceeds 512 by 512 grid points. Thus the amount of memory, needed for computations is $\frac{1}{4}8N = 2N$ Mb. Thus we can simultaneously operate from 2000 to 4000 Green's functions using modern GPUs. Additionally, for each subdomain, we need to construct a set of 2D slices of the images at each depth layer. The number of images hardly exceed 100, thus we can neglect this amount. A single 3D image for the whole domain should be stored in RAM, which requires, as estimated above, up to 4-8 Gb of RAM and it is independent on the datasets.

Division of the data leads to the significant loss of the algorithm scaling if compared with the hypothetical sequential one. This happens because some of the sources/receivers belong to several datasets, thus we need to compute the Green's functions for these sources/receivers several times, whereas in the hypothetical algorithm we would do it only ones. In our, numerical experiments, we achieve the number of the Green's functions recomputations as low as 2.3 in average. Examples of the datasets for synthetic SEG Salt marine data and for onshore field data are provided in figure. 2

3.4 Parallel algorithm MPI+OMP+CUDA

The parallel algorithm is similar to the sequential one, described above, but each MPI process operates with a single dataset. The block-scheme of the algorithm is provided in figure 3. So, the input data for a single MPI process is a single dataset. We perform the integration with respect to time-frequency in the outer

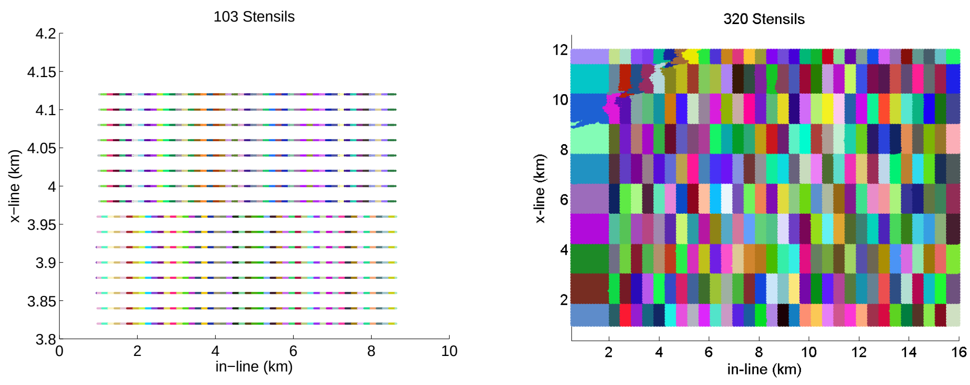

Fig. 2. Examples of the datasets (coordinates of the mid-points) for SEG Salt model (left), and real onshore seismic data (right).

loop. Then we use the loop with respect to the depth layers. We use GPU to compute the set of 2D cross-sections of the Green's functions (let us assume we proceed from layer z_j to z_{j+1}). Computations are based on the intense use of 2D qFFT. The set of images at the layer z_{j+1} are also computed by GPU. After that 2D slices of images are passed to RAM. While GPU constructs the solutions and images at the layer z_{j+1}, CPU sorts the 2D cross-sections of the images, use MPI send/recv to exchange the 2D cross-sections of images with other MPI processes, so that single-offset vector images for all datasets are accumulated by a single MPI process. After that, we use CPU to interpolate the accumulated image within the slab $[z_{j-1}, z_j]$ and sum it up to the final image. So, the results of one step with respect to the depth for a single MPI process are: set of 2D cross-sections of the Green's functions (all offsets, single dataset) at layer $z = z_{j+1}$; full set of 2D cross-sections of the images (all offsets, single dataset) at layer $z = z_{j+1}$; a single common offset image for all datasets on a fine grid within a slab $[z_{j-1}, z_j]$.

4 Numerical Experiments

4.1 Weak Scaling

Structure of the algorithm makes it difficult to estimate the strong scaling in its classical meaning; i.e. fixing a size of the problem and increase the number of MPI processes. However, we can estimate weak scaling, because increasing the number of MPI processes we process a larger number of datasets. We consider the SEG Salt model, for which we compute 17 images, having 2600 datasets. We perform simulations using 17, 34, 51, and 68 MPI processes. This means that a single node deals with one dataset at a time, but only 17 of them compute images. Measured times are provided in table 1. The loss of the efficiency is caused by the MPI exchanges, which are implemented with enforced synchronization.

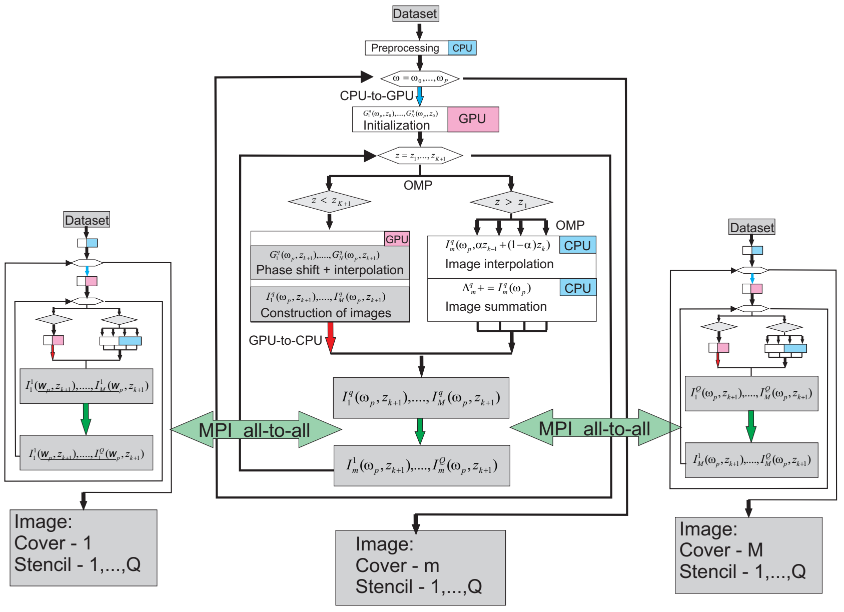

Fig. 3. A block-scheme of the parallel algorithm

Moreover, this algorithm allows using non-blocking procedures Isen/Irecv which can be performed during computations, however, in the presented version of the algorithm, it is not implemented.

Table 1. Computation time for weak scaling estimation.

	17 proc	34 proc	51 proc	68 proc
Time (hours)	1.91	2.22	2.64	3.02
W scaling (%)	-	86	72	63

4.2 Images Construction for SEG Salt model

We use the algorithm to compute the image for the SEG Salt model with pre-computed seismic data. The size of the model is 7980 m InLine (x-direction) and 7920 m CrLine (y-direction). The depth of imaging is 4000 m. We use the grid with the steps 20 by 20 meters in horizontal directions. Computations are done on a grid with step 50 m in the vertical direction, whereas the image is constructed on a grid with the step of 5 m in the vertical direction. We compute 17 common-offset vector images using 2600 datasets. We perform simulations using 51 nodes. Total computation time, was slightly less than the estimate, coming from table 1. So, the total computation time is 5032 core-hours. We provide the 2D cross-sections of the model and the image in figure 4

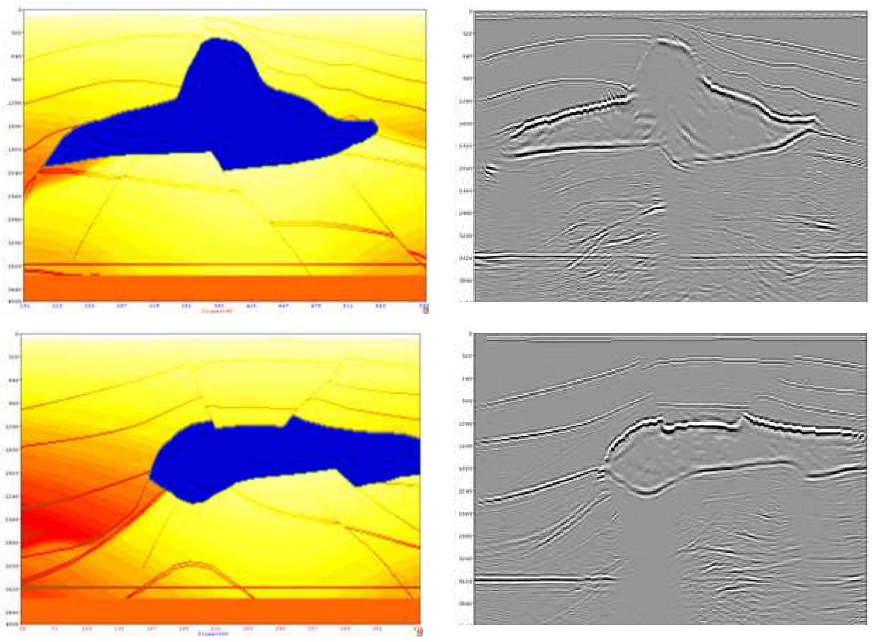

Fig. 4. Cross-sections of the SEG Salt model (left) and corresponding image (right), along a cross-line (upper row) and in-line (lower raw).

4.3 Real-data example

The second numerical experiment is done to construct the seismic image using real onshore seismic data. The size of the model is 15525 m InLine (x-direction) and 11250 m CrLine (y direction), depth is 5000 m. We use the grid with the steps 25 by 25 by 50 m for computations and 25 by 25 by 5 m for image construction. We compute 40 common-offset vector images, using 320 datasets. The geometry of the datasets is provided in figure 2. We use 40 nodes for simulations; i.e. equal to the number of images. One simulation takes almost 40 hours, thus total wall-clock time is about 320 hours, which leads to the 11520 node-hours to construct the set of 40 images. A slalom-line section of the obtained 3D seismic image is presented in figure 5.

5 Conclusion

We presented an original algorithm of seismic migration, based on the solution of one-way wave equation. The algorithm combines MPI, OMP, and CUDA

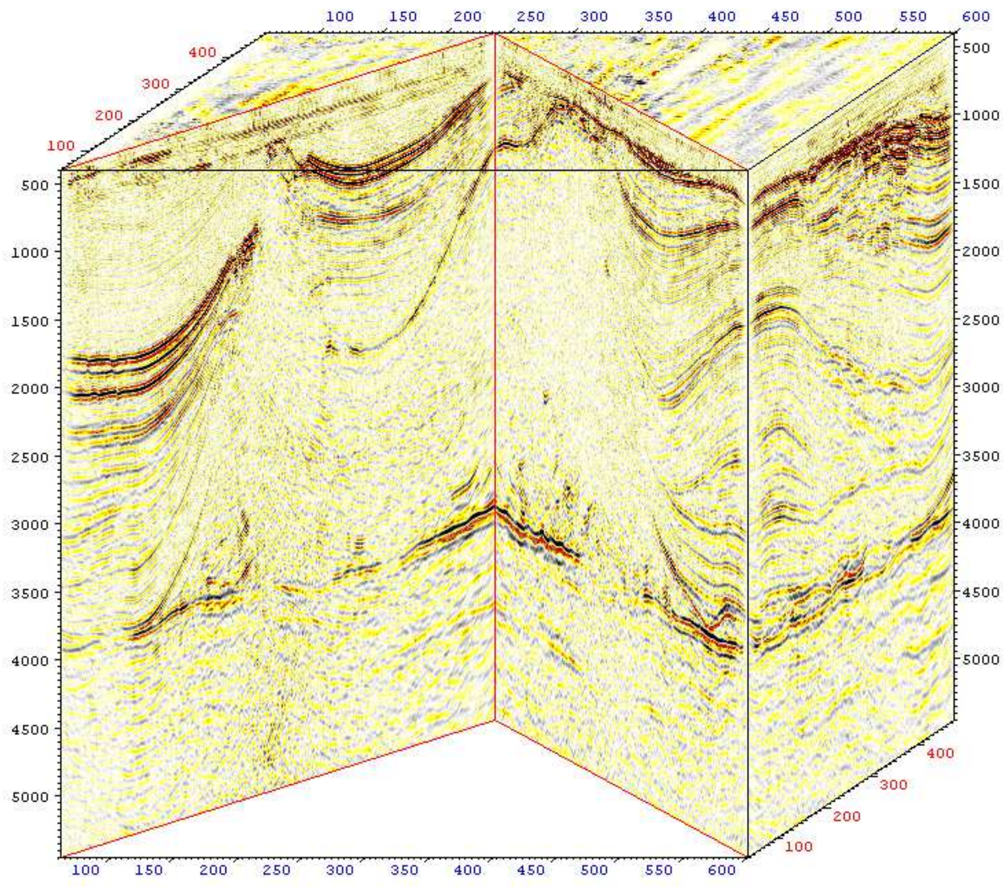

Fig. 5. Slalom-line section of the obtained 3D seismic image.

technologies. Dataflow is parallelized via MPI, so that each node deals with a single dataset. For computations of the Green's functions and the images for the dataset are performed by GPU. After that, the images are passed between the nodes using MPI. Additional, computations and I/O are implemented via OMP technology. As the result, we are able to compute the common-offset vector images by means of solving one-way wave equation, which is more accurate and informative than images constructed by conventional ray-based techniques.

Acknowledgements

This research was initiated and sponsored by Central Geophysical Expedition JSC of Rosgeo. V. Lisitsa and D. Vishnevsky are also thankful to Russian Foundation for Basic Research for partial financial support of this work, grans no. 18-05-00031, 18-01-00579, 16-05-00800. The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University and cluster NKS-30T+GPU of the Siberian supercomputer center.

References

1. Claerbout J.F.: Imaging the Earth's Interior. Blackwell Scientific Publication. 1985.
2. Etgen J., Gray S. H., Zhang Yu.: An overview of depth imaging in exploration geophysics. Geophysics, 74(6), WCA5–WCA17, (2009).
3. Fu L.-Y.: Wavefield interpolation in the Fourier wavefield extrapolation. Geophysics, 69(1), 257–264, (2004).
4. Gazdag J.: Wave equation migration with the phase-shift method. Geophysics, 43(7), 1342–1351, (1978).
5. Gazdag J, Sguazzero P.: Migration of seismic data by phase shift plus interpolation. Geophysics, 49(2), 124–131, (1984).
6. Lisitsa V., Podgornova O., Tcheverda V.: On the interface error analysis for finite difference wave simulation. Computational Geosciences, 14(4), 769–778, (2010).
7. Liu H.W., Liu H., Tong X.-L., Liu Q.: A Fourier integral algorithm and its GPU/CPU collaborative implementation for one-way wave equation migration. Computers & Geosciences, 45, 139–148, (2012).
8. Moczo P., Kristek J., Vavrycuk V., Archuleta R.J., Halada L.: 3D heterogeneous staggered-grid finite-differece modeling of seismic motion with volume harmonic and arithmetic averaging of elastic moduli and densities. Bulletin of the Seismological Society of America, 92(8), 3042–3066, (2002).
9. Moczo P., Kristek J., Galis M.: The finite-difference modelling of earthquake motion: Waves and Ruptures. Cambridge University Press. 2014.
10. Muir F., Dellinger J., Etgen J., Nichols D.: Modeling elastic fields across irregular boundaries. Geophysics, 57(9), 1189–1193, (1992).
11. Pleshkevich A., Vishnevsky D., Lisitsa V.: Explicit additive pseudospectral schemes of wavefield continuation with high-order approximation. SEG Technical Program Expanded Abstracts, 36(1), 5546–5550, (2017).
12. Pleshkevich A.L., Vishnevsky D.M., Licitsa V.V.: Development of pseudospectral amplitude-preserving 3D depth migration. Russian Geophysics, (S), 94–101, (2017).

13. Stoffa P. L., Fokkema J. T., de Luna Freire R. M., Kessinger W. P.: Split-step Fourier migration. Geophysics, 55(4), 410–421, (1990).
14. Virieux J., Calandra H., Plessix R.-E.: A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging. Geophysical Prospecting, 59(5), 794–813, (2011).
15. Virieux J., Operto S., Ben-Hadj-Ali H., Brossier R., Etienne V., Sourbier F., Giraud L., Haidar A.: Seismic wave modeling for seismic imaging. The Leading Edge, 28(5), 538–544, (2009).
16. Vishnevsky D., Lisitsa V., Tcheverda V., Reshetova G.: Numerical study of the interface errors of finite-difference simulations of seismic waves. Geophysics, 79(4), T219–T232, (2014).
17. Zhang J.-H., Wang S.-Q., Yao Z.-X.: Accelerating 3d Fourier migration with graphics processing units. Geophysics, 74(6), WCA129–WCA139, (2009).