

Parallel Implementation of 3D Seismic Beam Imaging

Maxim Protasov(✉)

Institute of Petroleum Geology and Geophysics, Novosibirsk 630090, Russia
protasovmi@ipgg.sbras.ru

Abstract. The paper presents the parallel implementation of prestack beam migration of 3D seismic data. The objective is to develop the parallel imaging algorithm for processing large enough volumes of 3D data in production mode, suitable for anisotropic media, and handling 3D irregular seismic data without any preliminary regularization. The paper provides a comparative analysis of the developed migration results with the industrial version of the Kirchhoff migration on synthetic and real data.

Keywords: 3D seismic imaging · Gaussian beams · Anisotropy · Parallelization

1 Introduction

At present, in seismic data processing, prestack depth migration is one of the most computationally expensive procedures. So the migration algorithm should be optimized as much as possible and effectively parallelized. The presented algorithm for three-dimensional anisotropic depth migration in true amplitudes originates from the approach realized in a two-dimensional version [1]. But when implementing the migration of 3D seismic data, the algorithm has undergone significant changes and become closer to the method of three-dimensional Gaussian beam migration, which has become the industry standard [2]. However, this migration option is implemented for regular data in the midpoint domain for each offset [2]. The approach considered in the paper does not require a regularization, and it provides handling irregular data both in the source-receiver and in the midpoint-offset domains.

Similarly, to the two-dimensional predecessor, the algorithm is implemented by extrapolating wavefields from the observation system to a certain fixed image point along anisotropic Gaussian beams [3–5], using individual Gaussian beams instead of decomposing the wavefield into Gaussian beams. The leading algorithmic optimization consists of constructing identical Gaussian beams on the observation surface. It leads to the possibility of dividing migration into two parts: the process of summing data with weights (or decomposing data into beams) and the process of mapping the decomposed data into depth (the migration process itself). Moreover, each of these two parts uses OpenMP technology [6], providing its implementation and optimization. And, the entire migration process is effectively implemented using MPI technology [6]. In general, this

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12949, pp. 611–621, 2021.
https://doi.org/10.1007/978-3-030-86653-2_44

approach made it possible to create a technological version of migration, which, in terms of computational costs, is comparable to the industrial implementation of Kirchhoff migration and superior in quality. The parallel implementation of the algorithm is verified on a representative set of synthetic and real data.

2 Seismic Migration as Asymptotic Inversion

The technique presented here provides an asymptotically correct true-amplitude image of multicomponent seismic data. In this section, the developed migration is described as an asymptotic inversion of seismic data.

2.1 Statement of the Problem

3D heterogeneous anisotropic elastic medium is considered with Lamé's parameters and density decomposing as follows:

$$c_{ijkl}(x,y,z) = c^0_{ijkl}(x,y,z) + c^1_{ijkl}(x,y,z),\ \rho(x,y,z) = \rho_0(x,y,z) + \rho_1(x,y,z). \quad (1)$$

The parameters $c^0_{ijkl}(x,y,z)$ and $\rho_0(x,y,z)$ describe a priori known smooth macro-velocity model/background/propagator while $c^1_{ijkl}(x,y,z)$ and $\rho_1(x,y,z)$ are responsible for its unknown rough/rapid perturbations or reflectors.

Born integral [7] describes the reflected/scattered wavefield on the surfaces:

$$\begin{aligned}&\vec{u}^{obs}(x_r,y_r;x_s,y_s;\omega)\\&= \int \boldsymbol{G}(x_r,y_r,z_r(x_r,y_r);x,y,z)\cdot \boldsymbol{L}_1\langle\vec{u}^0(x,y,z;x_s,y_s,z_s(x_s,y_s);\omega)\rangle dxdydz.\end{aligned} \quad (2)$$

Here $(x_r, y_r, z_r(x_r, y_r))$ is the receiver coordinate, $(x_s, y_s, z_s(x_s, y_s))$ is the source coordinate, ω is the frequency and $\overrightarrow{u}^0$ is the incident wavefield propagating in a smooth background from a volumetric point source, $\boldsymbol{G}$ – Green's matrix for the smooth background and operator $\boldsymbol{L}_1$ introducing by the rough perturbations $c^1_{ijkl}(x,y,z)$ and $\rho_1(x,y,z)$:

$$(\boldsymbol{L}_1\langle\vec{u}^0\rangle)_j = -\sum\nolimits_{i,l,l=1}^{3} \frac{\partial}{\partial x_i}\left(c^1_{ijkl}\frac{\partial u^0_l}{\partial x_k}\right) - \rho_1\omega^2 u^0_l;\ (x_1,x_2,x_3)\equiv(x,y,z). \quad (3)$$

The problem is to reconstruct rough perturbations of elastic parameters $c^1_{ijkl}(x,y,z)$ and density $\rho_1(x,y,z)$ or some of their combinations by resolving integral Eq. (2) with the data $\overrightarrow{u}^{obs}(x_r,y_r;x_s,y_s;\omega)$.

2.2 Asymptotic Inversion: Generalized Coordinates

On the acquisition surface, the beam center coordinates are introduced, i.e., $\overline{p}_{r0} = (x_{r0}, y_{r0})$ – receiver beam center and $\overline{p}_{s0} = (x_{s0}, y_{s0})$ – source beam center. Quasi-pressure (QP) rays are traced from every beam center to every image point $\overline{p}_i = (x_i, y_i, z_i)$ within the model. QP Gaussian beams are constructed along with these rays [3, 4], and

they are denoted by $\overrightarrow{u}_{qp}^{gbr}(x_r, y_r; \overline{p}_{r0}; \overline{p}_i; \omega)$ and $\overrightarrow{u}_{qp}^{gbs}(x_s, y_s; \overline{p}_{s0}; \overline{p}_i; \omega)$. Then on the acquisition surface, integration beam weights are computed: one is the normal derivative of the Gaussian beam at the receivers, another one is the normal derivative of the scalar part of the corresponding beam in the sources:

$$\overrightarrow{T}_{qp}^{gbr}\left(x_r, y_r; \overline{p}_{r0}; \overline{p}_i; \omega\right), T_{qp}^{gbs}\left(x_s, y_s; \overline{p}_{s0}; \overline{p}_i; \omega\right). \tag{4}$$

Both parts of the Born integral (2) are multiplied by the constructed weights (4) and integrated with respect to the source and receiver coordinates. Using saddle point technique for every beam weight (it is the 3D analog of the result described in the paper [1]), one can come to the following identity:

$$\begin{aligned}&\int T_{qp}^{gbs}\left(x_s, y_s; \overline{p}_{s0}; \overline{p}_i; \omega\right)\vec{T}_{qp}^{gbr}\left(x_r, y_r; \overline{p}_{r0}; \overline{p}_i; \omega\right)\vec{u}^{obs}(x_r, y_r; x_s, y_s; \omega)dx_r dy_r dx_s dy_s \\ &= \int \vec{u}_{qp}^{gbr}\left(\overline{p}; \overline{p}_{r0}; \overline{p}_i; \omega\right) \cdot \boldsymbol{L}_1 \langle \vec{u}_{qp}^{gbs}\left(\overline{p}; \overline{p}_{s0}; \overline{p}_i; ; \omega\right)\rangle d\overline{p}\end{aligned} \tag{5}$$

Here is $\overline{p} \equiv (x, y, z)$. The representation of qP beams: $\overrightarrow{u}_{qp}^{gbr(s)} = \overrightarrow{e}_{qp}^{gbr(s)}\varphi_{qp}^{gbr(s)}$.

And computations of beam derivatives (operator $\boldsymbol{L}_1$) retaining terms up to the first order only, together with the microlocal analysis (it is analogous to the research described in the paper [1]) of the right-hand side of (5) gives the following:

$$\begin{aligned}&\int T_{qp}^{gbs}\left(x_s, y_s; \overline{p}_{s0}; \overline{p}_i; \omega\right)\vec{T}_{qp}^{gbr}\left(x_r, y_r; \overline{p}_{r0}; \overline{p}_i; \omega\right)\vec{u}^{obs}(x_r, y_r; x_s, y_s; \omega)dx_r dy_r dx_s dy_s \\ &= \omega^2 \int \exp\left\{i\omega\left(\nabla\tau_s\left(\overline{p}_i; \overline{p}_{s0}\right) + \nabla\tau_r\left(\overline{p}_i; \overline{p}_{r0}\right)\right) \cdot \left(\overline{p}_i - \overline{p}\right)\right\} f\left(\overline{p}; \overline{p}_{r0}; \overline{p}_{s0}\right) d\overline{p}.\end{aligned} \tag{6}$$

Here τ_s, τ_r are travel times from the image point $\overline{p}_i$ to the corresponding acquisition point $\overline{p}_{s0}, \overline{p}_{r0}, f$ is a linearized reflection coefficient. Introducing generalized coordinates $\overline{p}_{01} = (x_{01}, y_{01}), \overline{p}_{02} = (x_{02}, y_{02})$ that depends on beam center coordinates $\overline{p}_{s0}, \overline{p}_{r0}$, the change of variables $(x_{01}, y_{01}, \omega) \to (k_x, k_y, k_z)$ in the right-hand side of (6) is used:

$$\overline{k} = \left(k_x, k_y, k_z\right) = \omega\left(\nabla\tau_s\left(\overline{p}_i; \overline{p}_{01}; \overline{p}_{02}\right) + \nabla\tau_r\left(\overline{p}_i; \overline{p}_{01}; \overline{p}_{02}\right)\right). \tag{7}$$

Multiplication by generalized Beylkin's determinant [8]:

$$BelDet\left(\overline{p}_i; \overline{p}_{01}; \overline{p}_{02}\right) = \begin{vmatrix} \nabla_x(\tau_s+\tau_r) & \nabla_y(\tau_s+\tau_r) & \nabla_z(\tau_s+\tau_r) \\ \frac{\partial\nabla_x(\tau_s+\tau_r)}{\partial x_{01}} & \frac{\partial\nabla_y(\tau_s+\tau_r)}{\partial x_{01}} & \frac{\partial\nabla_z(\tau_s+\tau_r)}{\partial x_{01}} \\ \frac{\partial\nabla_x(\tau_s+\tau_r)}{\partial y_{01}} & \frac{\partial\nabla_y(\tau_s+\tau_r)}{\partial y_{01}} & \frac{\partial\nabla_z(\tau_s+\tau_r)}{\partial y_{01}} \end{vmatrix}, \tag{8}$$

and integration of both parts of the identity (6) with respect to x_{01}, y_{01}, ω gives the final asymptotic inversion result:

$$\begin{aligned}&f\left(\overline{p}_i; \overline{p}_{02}\right) = \int BelDet\left(\overline{p}_i; \overline{p}_{01}; \overline{p}_{02}\right) \cdot T_{qp}^{gbs}\left(x_s, y_s; \overline{p}_{01}; \overline{p}_{02}; \overline{p}_i; \omega\right) \\ &\cdot\vec{T}_{qp}^{gbr}\left(x_r, y_r; \overline{p}_{01}; \overline{p}_{02}; \overline{p}_i; \omega\right) \cdot \vec{u}^{obs}(x_r, y_r; x_s, y_s; \omega)dx_r dy_r dx_s dy_s d\overline{p}_{01} d\omega.\end{aligned} \tag{9}$$

The formula (9) is called imaging condition and provides true amplitude seismic image in generalized acquisition coordinates $\overline{p}_{01}, \overline{p}_{02}$. The last means the imaging condition provides handling any acquisition coordinates. Particularly, asymptotic inversion can utilize beam center receiver coordinates $\overline{p}_{r0} = (x_{r0}, y_{r0})$ while the final result provides averaged linearized reflection coefficient with respect to beam center source coordinates $\overline{p}_{s0} = (x_{s0}, y_{s0})$. Another helpful option uses structural dip and azimuth angle coordinates $\overline{p}_{r0} \equiv (\gamma, \theta)$ [9] in the inversion process and the final image depends on the opening angle and the azimuth of the opening angle $\overline{p}_{s0} \equiv (\beta, az)$ [9].

In the standard industrial technology of seismic processing, the data is usually processed in common midpoint and offset domains. To get the image in these terms, one should use the following representation of generalized acquisition coordinates:

$$\begin{aligned} \overline{p}_{01} &\equiv (x_m, y_m) = ((x_{s0} + x_{r0})/2, (y_{s0} + y_{r0})/2), \\ \overline{p}_{02} &\equiv \left(h_x, h_y\right) = (x_{r0} - x_{s0}, y_{r0} - y_{s0}) \end{aligned} \tag{10}$$

In this case, the inversion is done with respect to midpoints, while the final image is got with respect to offsets.

3 Implementation of Migration Operator

The software implementation utilizes the division of the whole imaging procedures by two blocks (Fig. 1). The first one computes beam attributes and stores them to the disk, and the second one provides seismic data mapping via beam attributes usage. The division into two parts makes it possible to implement offset-midpoint imaging condition (9) effectively. This way provides an opportunity to perform beam data decomposition before their mapping into the image domain. Therefore, computational cost decreases significantly because the summation of data occurs once for all image points, in contrast to the straightforward realization without beam data decomposition when there is a separate summation for each image point. Taken together, this approach provides an opportunity to implement a technological version handling industrial volumes of 3D seismic data within a reasonable computation time.

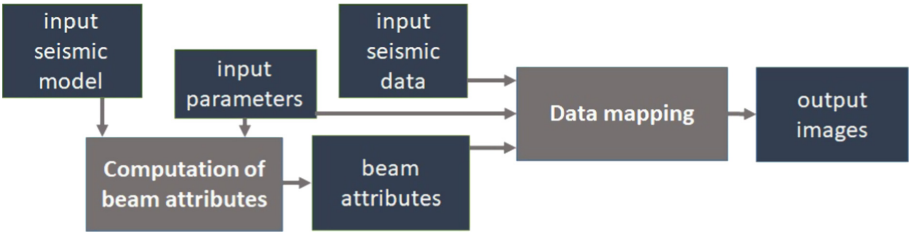

Fig. 1. General upper-level diagram for the realization of the offset-midpoint imaging procedure. In all the diagrams and schemes (Fig. 1, 2, 3, 4, and 5), dark blue color means data, and brown color implies a process (Color figure online).

3.1 Parallelization Scheme

Since the attribute computation block works separately from the migration itself, then the parallelization is performed independently for each such block also. Scheme for the attributes computation utilizes two-level parallelism. The first level uses MPI parallelization done by beam centers. Every node extracts and loads the corresponding part of the model and computes beam attributes for the beam centers portion (Fig. 2). The second level on every node provides OpenMP parallelization by the number of rays, i.e., on beams, where every CPU core provides a corresponding bunch of rays (Fig. 2).

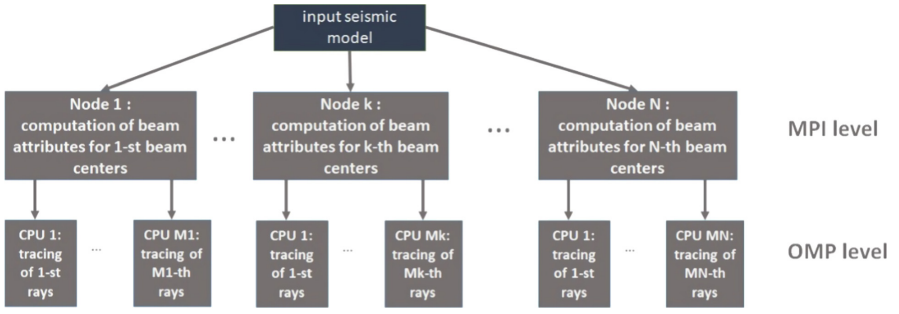

Fig. 2. Parallelization scheme of beam attributes computation procedure.

After the beam attributes computation procedure finishes, the seismic data mapping block starts processing. Its realization also utilizes MPI and OpenMP technologies. MPI realization is done for every offset with respect to the beam center midpoints (Fig. 3).

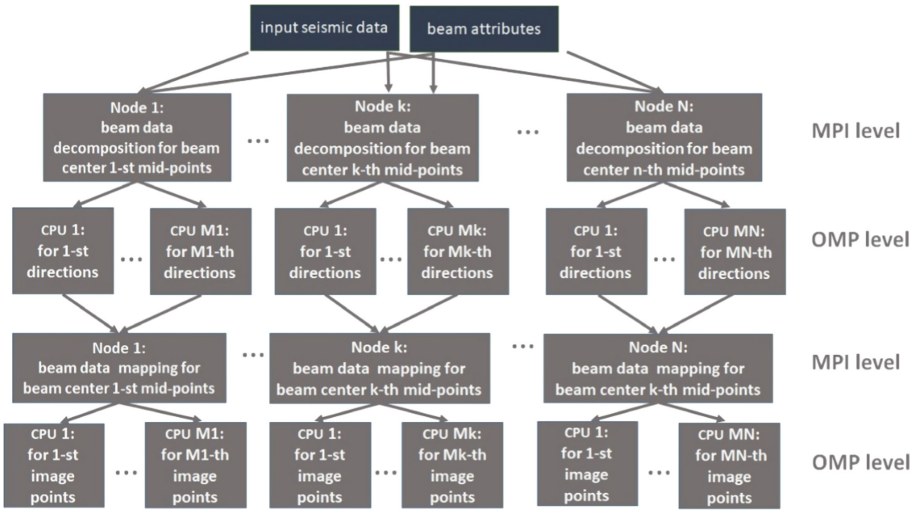

Fig. 3. Parallelization scheme of data mapping procedure.

Then the algorithm contains two steps on every MPI node. The first step provides OpenMP realization of beam decomposition with respect to directions. Every node accumulates OpenMP results, and then the second step again utilizes OpenMP technology for beam data mapping into the image domain with respect to portions of image points (Fig. 3). Every MPI process works independently, so there are no connections between MPI nodes. But the optimization implementation requires uniform data distribution between nodes which is realized based on data coordinates analysis.

3.2 The Functionality of the Algorithm

The straightforward implementation of the imaging condition (9) requires high computational costs for the migration of modern production volumes of data. Simultaneously, the realization of such a migration in structural angles automatically calculates Common Image Gathers (CIGs) depending on the opening angle and opening azimuth. Of course, this is a useful functionality required for amplitude analysis and amplitude inversion depending on the medium's parameters (for example, impedances). On the other hand, the effective implementation of the imaging condition (9) makes it possible to calculate the common image point's seismograms depending on the offsets. Such seismograms are certainly necessary, for example, for velocity analysis, but angle domain seismograms are also needed, for instance, for amplitude analysis. Therefore, within the framework of the migration algorithm based on the imaging condition (9), calculating angle domain gathers is implemented. In short, inside the migration process for each image point, a specific function provides the correspondence between the angle and the offset by searching for a stationary point. After migration, resorting and interpolation to a regular grid specified by the user takes place (Fig. 4). In this case, the additional computational time for sorting and interpolation occurs already after the migration stage.

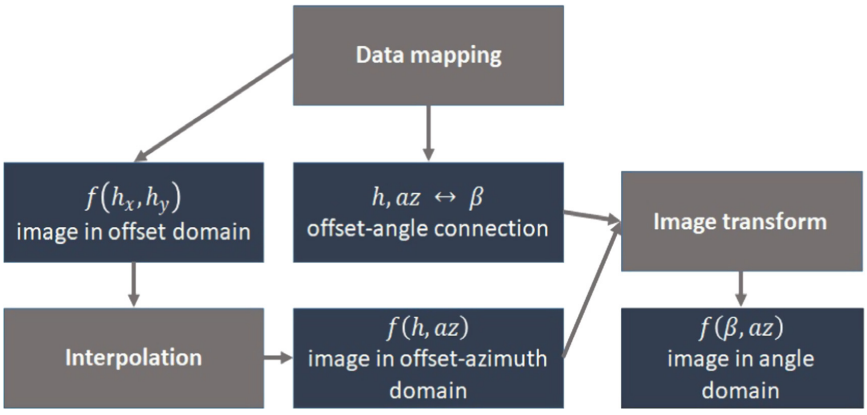

Fig. 4. Scheme for functionality realization of angle domain common image gathers computation.

Implementing the imaging condition in the structural angles (9) makes it possible to calculate selective images or common image point seismograms depending on the angles,

which are essential for the diffraction image construction. It was also required to have this option when implementing the imaging condition (9) in midpoint offset coordinates. There were two calculation options. The first option is carried out by splitting the image into angular seismograms during migration. Of course, in this case, selective images will not be as focused as in the implementation of the imaging condition (9) in structural angles. However, in many cases, they can still be quite helpful. This option requires significantly more memory (RAM) but has a very weak effect on computations' cost. On the other hand, such images are calculated by processing the total image in the Fourier domain, where the component corresponding to the calculated angles is selected (Fig. 5). A reasonable combination of these methods makes it possible to construct diffraction images of sufficiently good quality.

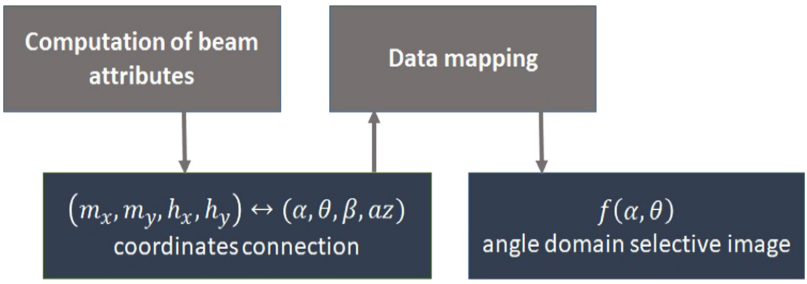

Fig. 5. Scheme for functionality realization of angle domain selective images computation.

The imaging condition (9) in terms of midpoint-offset coordinates makes it possible to implement a technological version of migration. In contrast, the migration algorithm based on the imaging condition (9) in structural angles requires much more computational resources. Conducted specific calculations for these implementations on identical data and equipment show that the computational time difference reaches two orders of magnitude (Table 1). This difference leads to the fact that structural angle realization is acceptable only for small amounts of data in a relatively small image area. On the other hand, midpoint-offset realization provides similar functionality as the structural angle version. Simultaneously, it allows handling large enough data set within reasonable computational resources, and therefore it is suitable for industrial processing.

Table 1. Computational time comparison for midpoint-offset realization and structural angle realization.

Data type	Data size	Computational time (1 CPU) structural angle realization	Computational time (1 CPU) midpoint-offset realization
2D data set	2 GB	40 h	1 h
3D data set	26 GB	4000 h	37 h

4 Numerical Examples

This section provides numerical results and a comparative analysis of the developed migration results with the industrial version of the Kirchhoff migration on 3D synthetic data and 3D marine seismic data.

4.1 Synthetic Data Set

Today, the need to test any developed algorithm on synthetics before using it on real data has become the standard. Amount of synthetic data are created for realistic models. One of the well-known 3D datasets is the free SEG-salt model data [10]. The model is 3D and contains a salt body (Fig. 6a, 7a). The dataset simulates an offshore seismic survey. For this model and data, two migration tests are performed. One of them provides Kirchhoff industrial migration test for the SEG-salt smoothed model. In this case, the model is smoothed to obtain the horizontal interface's best quality image at a depth of about 3500 m. Comparison of two-dimensional slices of the SEG-salt model (Fig. 6a, 7a) and similar slices of a three-dimensional image obtained by Kirchhoff migration (Fig. 6c, 7c) show that the position of this boundary is restored quite well in many places. The same results demonstrate a rather good restoration of the lower interface of the salt body. However, at the same time, the upper boundary is blurred and defocused. Another test provides Gaussian beam migration, using a special algorithm for a more accurate description of the ray-tracing through a salt body. Comparison of 2D slices of the SEG-salt model (Fig. 6a, 7a), similar slices of a 3D Kirchhoff migration image (Fig. 6c, 7c), and the corresponding Gaussian beam migration results (Fig. 6b, 7b) are presented. They show that the lower horizontal interface at a depth of 3500 m is reconstructed much better using Gaussian beams than with Kirchhoff migration with "optimal" model smoothing. Obviously, in the subsalt zone at a depth of between 2000 and 3000 m, the Gaussian beam imaging procedure reconstructs inclined boundary, while on the result of Kirchhoff migration, it is practically absent. Moreover, the salt boundary is more reliably determined by the Gaussian beam imaging algorithm (Fig. 6b, 7b) than by the Kirchhoff migration procedure (Fig. 6c, 7c), where migration artifacts complicate the image.

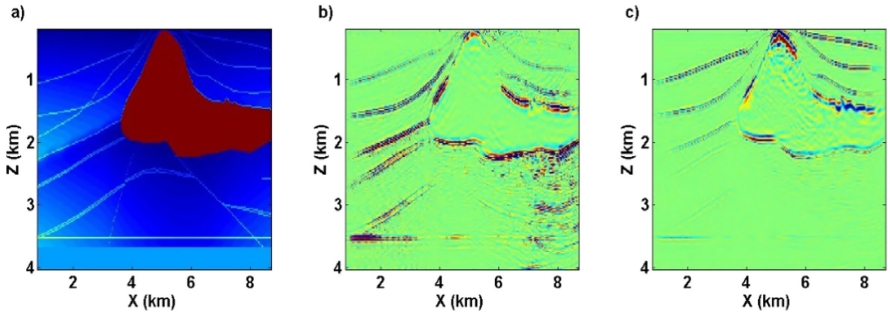

Fig. 6. 2D sections of 3D volumes for the coordinate y = 7800 m: a) SEG-salt model; b) beam imaging result; c) Kirchhoff migration image.

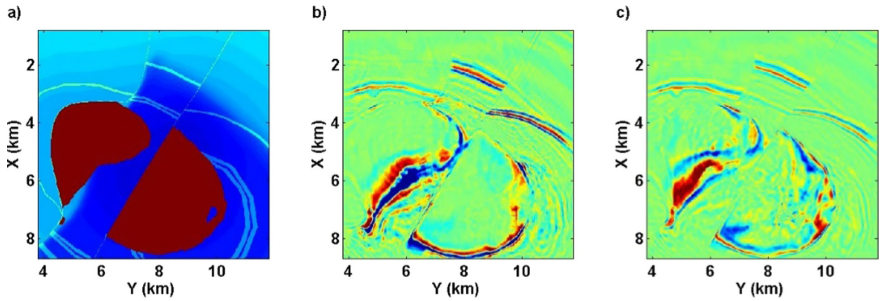

Fig. 7. 2D sections of 3D volumes at a depth of 2000 m: a) SEG-salt model; b) beam imaging result; c) Kirchhoff migration image.

4.2 Real Data Set

The developed anisotropic Gaussian beam migration is applied to the real data. For the data at the velocity model building stage, the anisotropic depth model is constructed. The industrial anisotropic Kirchhoff migration is used for the model and the data also. Gaussian beam migration results (Fig. 8a, 9a) show sufficiently focused reflected events, which indicates that the anisotropy during migration is taken into account correctly. Comparing the images obtained by beam migration (Fig. 8a, 9a) and Kirchhoff migration (Fig. 8b, 9b) provides the resolution of the Gaussian beam imaging exceeds the Kirchhoff migration resolution. The seismic horizons in the image obtained from the beam migration are traced better than those obtained by the Kirchhoff migration. The discussed images were obtained for the area 300 km^2 and the data size almost 600 Gigabytes. The computational resources contain ten nodes with 20 CPU cores on each computation node. Such data and resources provide approximately 23 h for running the beam imaging procedure (Table 2). The computation time of Kirchhoff migration is about 20 h in the same environment (Table 2). So computational times are similar, but the image quality of the beam migration is better.

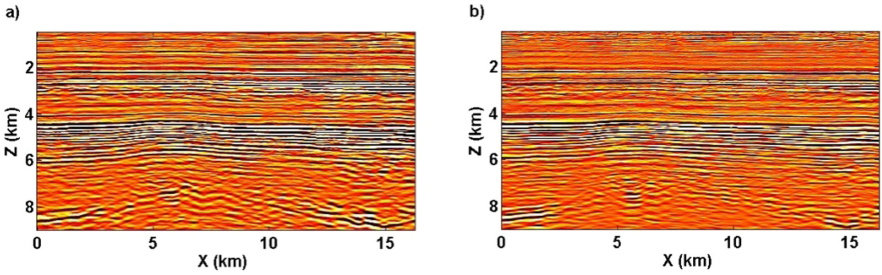

Fig. 8. 2D sections of 3D images for the coordinate y = 8.5 km: a) beam migration result; b) Kirchhoff migration result.

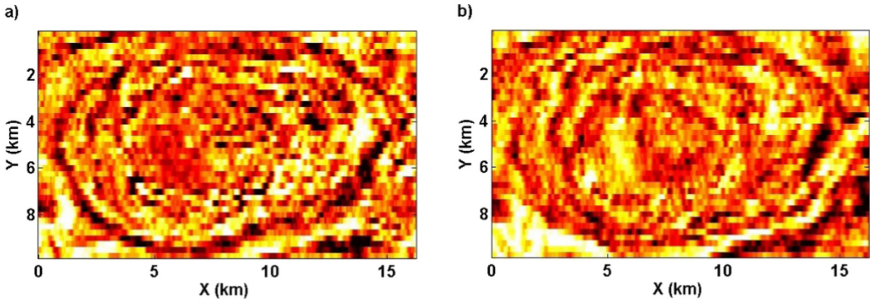

Fig. 9. 2D sections of 3D images at a depth of 6 km: a) beam migration result; b) Kirchhoff migration result.

Table 2. Computational time comparison for beam and Kirchhoff migrations computed on the real marine data set.

Migration	Data size	Computational resources	Computational time
Beam	592 GB	10 Nodes × 20 CPU cores	≈23 h
Kirchhoff	592 GB	10 Nodes × 20 CPU cores	≈20 h

5 Conclusions

The paper presents the parallel implementation of the 3D seismic beam imaging algorithm for anisotropic media. The algorithm can be applied to irregular data either in source-receiver or midpoint-offset domains. Acquisition domain realization provides similar functionality as the structural angle version, but it allows handling large enough data sets and is suitable for industrial processing. The implementation of beam imaging utilizes MPI and OpenMP technologies, effectively providing two-level parallelization with division into beam attribute and data mapping blocks with intrinsic separate data decomposition. The computational time comparison of the developed beam imaging and industrial Kirchhoff migration shows similar behavior. However, the migration of freely distributed synthetic data SEG-Salt and real data proves the developed algorithm's advantages. The obtained stacked images are cleaner and more coherent in comparison with the results of Kirchhoff migration.

Acknowledgments. The reported study was funded by RFBR and GACR, project number 20-55-26003.

References

1. Protasov, M.I.: 2-D Gaussian beam imaging of multicomponent seismic data in anisotropic media. Geophys. J. Int. **203**, 2021–2031 (2015)
2. Hill, N.R.: Prestack Gaussian-beam depth migration. Geophysics **66**, 1240–1250 (2001)

3. Nomofilov, V.E.: Asymptotic solutions of a system of equations of the second order which are concentrated in a neighborhood of a ray. Zap. Nauch. Sem. LOMI **104**, 170–179 (1981)
4. Cerveny, V.: Seismic Ray Theory. Cambridge University Press, Cambridge (2001)
5. Popov, M.M.: Ray theory and Gaussian beam for geophysicists. EDUFBA, SALVADOR-BAHIA (2002)
6. Parallel data processing homepage. https://parallel.ru/vvv/. Accessed 3 Mar 2021
7. Devaney, A.J.: Geophysical diffraction tomography. IEEE Trans. Geosci. Remote Sens. **22**, 3–13 (1984)
8. Beylkin, G.: Imaging of discontinuous in the inverse scattering problem by inversion of causual generalized Radon transform. J. Math. Phys. **26**(1), 99–108 (1985)
9. Protasov, M.I., Tcheverda, V.A., Pravduhin, A.P., Isakov, N.G.: 3D anisotropic imaging of 3D seismic data on the basis of Gaussian beams. Seismic Technol. **1**, 35–48 (2017)
10. SEG/EAGE 3D modeling Salt Model Phase-C Homepage. https://wiki.seg.org/wiki/SEG/EAGE_3D_modeling_Salt_Model_Phase-C_1996. Accessed 2 May 2016