



# On an Algorithm for Decomposing Multi-Block Structured Meshes for Calculating Dynamic Wave Processes in Complex Structures on Supercomputers with Distributed Memory

*Ilia N. Agrelov*<sup>1</sup> , *Nikolay I. Khokhlov*<sup>1,2</sup> , *Vladislav O. Stetsyuk*<sup>1</sup>,  
*Sergey D. Agibalov*<sup>1</sup>

© The Authors 2024. This paper is published with open access at SuperFri.org

The advancement of the oil and gas industry represents a key priority area for the Russian Federation. The Arctic region contains substantial hydrocarbon reserves, but the inherent difficulties in exploring these resources make them particularly challenging to access. The present paper is devoted to the numerical calculation of the dynamic impact propagation on an oil platform using parallel computing methods. To address this issue, a grid-characteristic method was employed. The substantial volume of computation necessitates the utilization of parallel computing techniques, such as Message Passing Interface (MPI). A grid model was constructed based on a real platform, and an algorithm for decomposing the computational domain was developed with the aim of reducing the message time between MPI processes and increasing speedup. A series of test calculations were performed to demonstrate the capabilities of the algorithms. Examples of calculations and the application of the developed method of decomposition are provided. The feasibility of decomposition and parallelization algorithms is currently being investigated. The conducted tests have demonstrated the potential for using the model for real calculations.

*Keywords:* parallel calculation, numerical modeling, wave propagation, multiple grids modeling, parallel algorithm.

## Introduction

The numerical modeling of the propagation of dynamic wave perturbations in solids is a powerful tool that can be employed to solve a diverse array of problems. Such problems include seismic exploration, seismic stability, computational geophysics, and dynamic strength problems. The oil and gas industry represents a substantial component of the Russian Federation's economic structure. The most significant reserves of hydrocarbons are located in the North and the Arctic. The expansion of the oil and gas complex contributes to the enhancement of the country's economic potential, facilitating the growth of modern infrastructure. Russia is engaged in the active development of its Arctic territories, including the exploration of new oil and gas fields. As indicated by publicly available data, approximately 13% of the world's oil reserves and up to 30% of the world's gas reserves are situated within the Arctic region. Despite the considerable potential, the development of the Arctic shelf is significantly constrained by the unique characteristics of its location and the natural environment. This gives rise to risk factors that are absent in temperate latitudes, necessitating the development of new technologies and advances in this direction. One such area of focus is the development and research of oil platforms.

Oil platforms are intricate structures situated on the Arctic shelf. One of the most critical considerations is their resilience to dynamic loads, which can be triggered by a multitude of factors, both anthropogenic and natural. Dynamic disturbances may emerge from seismic loads, collisions with diverse objects (ice, ship, etc.), and explosive impacts (human-made and terror-

---

<sup>1</sup>Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

<sup>2</sup>Scientific Research Institute for System Analysis of the Russian Academy of Sciences, Moscow, Russian Federation

ist). This paper considers the issue of wave disturbance propagation in the structure of an oil platform.

A common approach to addressing such issues is the grid-characteristic method [12, 22]. Due to the long computation time, parallel computing techniques, such as MPI, are required [15, 16]. To reduce the communication time between processes, we developed a grid decomposition method.

Parallel computing is a computational paradigm that involves the concurrent execution of multiple calculations or processes. It is often the case that large tasks can be divided into smaller ones, which can then be solved simultaneously. This has the effect of greatly reducing the time required to complete the solution. The architecture of parallel computing systems can be classified into two principal categories based on the organization of work with memory. These are systems with shared memory and systems with distributed memory. The most renowned technologies for the construction of parallel programs are MPI, OpenMP, CUDA, and OpenCL [27]. This work employs the use of MPI technology. This technology represents one of the most prevalent techniques for the development of parallel programs in distributed memory systems. The abbreviation MPI stands for Message Passing Interface. As the name suggests, the primary mechanism for process interaction in MPI is message passing between processes. MPI is employed in a variety of fields, including ocean and climate modeling [8], plasma modeling [21], ice sheet modeling [19], and density functional theory problems [2].

The application of parallel computing in problem-solving processes necessitates the implementation of effective computational grid decomposition techniques to ensure a balanced distribution of computational workloads and minimize the time required for interprocessor communication. Two principal types of grid decomposition are distinguished: dynamic and static. Static decomposition is conducted prior to the initiation of the computational process, whereas dynamic decomposition can be executed an unlimited number of times during the course of the computation. Dynamic decomposition is employed when the load on the processes may fluctuate over time. Dynamic decomposition methods are implemented in various tools, including Zoltan [9] and DRAMA [3]. In work [5], a dynamic decomposition method utilizing optimal vector field approximation is proposed. This method is employed for the resolution of equations arising in the context of diffusion equation [6]. The method of dynamic decomposition using Voronoi diagrams is described in detail in [20].

Computational meshes can be classified into two main categories: structural meshes, the nodes of which are ordered, and non-structural meshes, the elements of which are unordered. The decomposition of non-structural grids is frequently reduced to the graph partitioning problem [4, 7]. In works [23, 24] evolutionary algorithms are employed to identify optimal subgrid sizes. Additionally, numerous software packages are available for the decomposition of non-structural grids, including METIS [17], ParMETIS [18], Jostle [26], and [14], among others.

The decomposition of structural meshes is a relatively straightforward process, in comparison to the decomposition of non-structural meshes. The most basic approach is the geometric decomposition algorithm, which involves dividing the computational domain into equal parts along several axes [10]. More advanced algorithms for mesh generation are presented in [1, 28]. In the paper [29], a convolutional neural network is employed for the decomposition of the study area. The majority of decomposition methods partition the meshes into non-overlapping subgrids. However, in work [25] authors propose an alternative approach, whereby the study area is partitioned into overlapping sub-areas.

This paper presents an algorithm for static decomposition of multi-block structured meshes, with the objective of speedup the computation of wave disturbance propagation in an oil platform structure. The greedy decomposition method was employed to address this problem. This method is based on the algorithms described in the articles [11, 13]. The article is organized as follows. Section 1 is dedicated to the derivation of the constitutive system of equations. Section 2 explores a methodology for solving the aforementioned system. Section 3 focuses on the construction of a grid model of an oil platform. Section 4 examines a decomposition method developed to enhance computational speed. Section 5 presents the results of measuring the computational speedup using the developed method, as well as test calculations of disturbance propagation in the oil platform structure from different sources. The conclusion of the study presents a summary of the findings and suggests ways in which further research could be conducted.

## 1. Mathematical Model

This section will examine the mathematical formulation of the problem. In this study, we employ the linear theory of elasticity to describe an elastic solid isotropic body.

We designate the displacement vector at a given point in space, represented by the radius vector  $\vec{x}$  as  $\vec{u}(\vec{x}, t)$ . Additionally, a deformation tensor,  $\varepsilon(\vec{x}, t)$ , is introduced, the components of which are calculated from the displacement vector according to the following rule:

$$\varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad i, j \in \{1, 2, 3\}. \quad (1)$$

The law of motion for each point within the medium can be expressed using Newton's second law in the following form:

$$\rho \frac{\partial^2 u_i}{\partial t^2} - \sum_{j=1}^3 \frac{\partial \sigma_{ij}}{\partial x_j} - f_i = 0, \quad i \in \{1, 2, 3\}, \quad (2)$$

where  $f(\vec{x}, t) = (f_1, f_2, f_3)$  is a density of the field of forces acting on the medium,  $\sigma_{ij}$  is the stress tensor and  $\rho = \rho(\vec{x}, t)$  is the density distribution of the medium material at each point.

In the context of small deformations, the stress and small displacement tensors are related by Hooke's law:

$$\sigma_{ij} = \sum_{k=1}^3 \sum_{l=1}^3 C_{ijkl} \varepsilon_{kl}, \quad i, j \in \{1, 2, 3\}. \quad (3)$$

The fourth-rank tensor  $C_{ijkl}$  is a tensor of elastic constants that defines the relationship between strain and stress tensors. The fourth-rank tensor has 81 components; however, due to its symmetry, as well as the symmetry of the stress and strain tensors, it can be described by only 21 independent constants. Consequently, the tensor  $C_{ijkl}$  can be expressed as a tensor of the second rank

$$C_{\alpha\beta} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix}. \quad (4)$$

In case of an isotropic linear-elastic medium, the equality (3) is greatly simplified. The number of independent variables is reduced to 2, and the elastic constant tensor (4) can be written as

$$C_{\alpha\beta} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}, \quad (5)$$

where  $\lambda$  and  $\mu$  are the so-called elastic Lamé parameters.

Hooke's law (3) for this case can be written as

$$\sigma_{ij} = \lambda \delta_{ij} \sum_{k=1}^3 \varepsilon_{kk} + 2\mu \varepsilon_{ij}, \quad i, j \in \{1, 2, 3\}, \quad (6)$$

where  $\delta_{ij}$  is the Kronecker symbol.

We may define the velocity vector  $\vec{v}$  as the time derivative of the displacement vector  $\vec{u}$ . With this definition, we can write the system of equations as follows:

$$\begin{aligned} \rho \frac{\partial \vec{v}}{\partial t} &= (\nabla \cdot \sigma)^\top + \vec{f}, \\ \frac{\partial \vec{\sigma}}{\partial t} &= \lambda (\nabla \cdot \vec{v}) I + \mu (\nabla \otimes \vec{v} + (\nabla \otimes \vec{v})^\top). \end{aligned} \quad (7)$$

## 2. Grid-Characteristic Method

In order to solve the system of equations derived in the preceding chapter, the grid-characteristic method was employed. To this end, the system was represented in matrix form.

Let  $\mathbf{q} = [v_1, v_2, v_3, \sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]^\top$ . Then system (7) can be written in the following form:

$$\frac{\partial \mathbf{q}}{\partial t} - \mathbf{A}_1 \frac{\partial}{\partial x_1} \mathbf{q} - \mathbf{A}_2 \frac{\partial}{\partial x_2} \mathbf{q} - \mathbf{A}_3 \frac{\partial}{\partial x_3} \mathbf{q} = 0. \quad (8)$$

We perform splitting by spatial coordinates, i.e., we consider separately 3 systems of the form

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{A}_i \frac{\partial}{\partial x_i} \mathbf{q}. \quad (9)$$

Given that the system is hyperbolic, it follows that all matrices  $\mathbf{A}_i$  possess a complete set of eigenvalues and linearly independent eigenvectors. Consequently, a diagonalization of these matrices can be conducted

$$\mathbf{A}_i = \mathbf{\Omega}_i^{-1} \mathbf{\Lambda} \mathbf{\Omega}_i, \quad (10)$$

where  $\mathbf{\Omega}_i$  consists of columns that are eigenvectors of  $A_i$  and  $\mathbf{\Lambda}$  is a diagonal matrix consisting of eigenvalues.

Then we pass to the new variables  $\omega_i$  by multiplying  $q$  by the matrix of eigenvectors  $\mathbf{\Omega}$

$$\mathbf{w}_i = \mathbf{\Omega}_i \mathbf{q}. \quad (11)$$

Then the system will be written as:

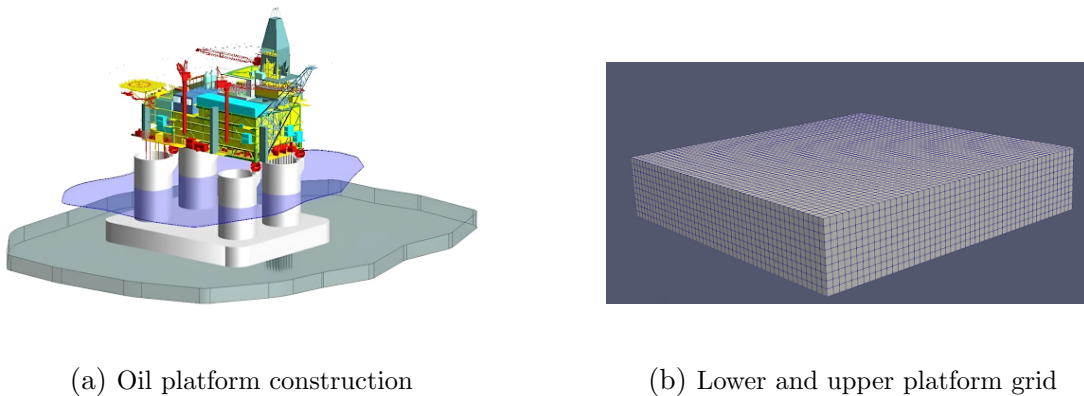
$$\frac{\partial}{\partial t} \mathbf{w}_i + \Lambda_i \mathbf{w}_i = 0. \quad (12)$$

Since the matrix  $\Lambda$  is diagonal, this system consists of independent equations, each of which is a transport equation. In numerical simulations, finite-difference schemes are employed for the solution of these equations. In the present study, the Rusanov scheme of the third order of accuracy was utilized.

### 3. Grid Construction

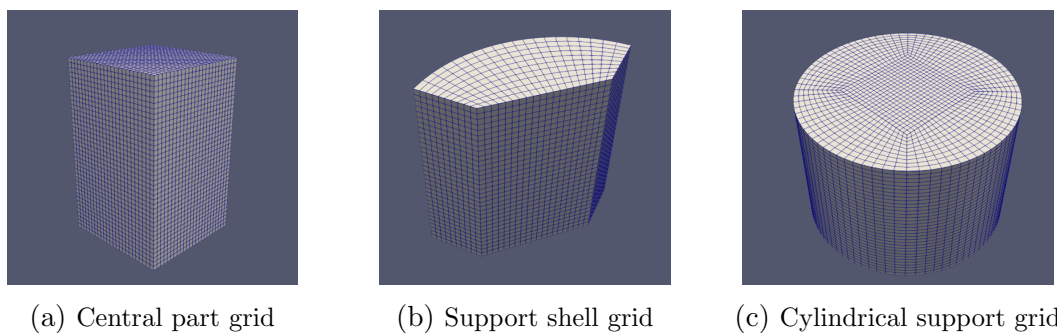
In this section, we will examine the process of constructing the grids. In this work, we have employed the use of structured grids. The utilization of this specific type of grids enables the acceleration of calculations by reducing the required random-access memory.

The lower platform has a geometric shape that is largely similar to that of a rectangular parallelepiped (Fig. 1b). Due to the heterogeneity of the upper platform with the infrastructure (Fig. 1a) and the lack of sufficient information about the dimensions of its parts, it was decided to approximate it by a rectangular parallelepiped as well.



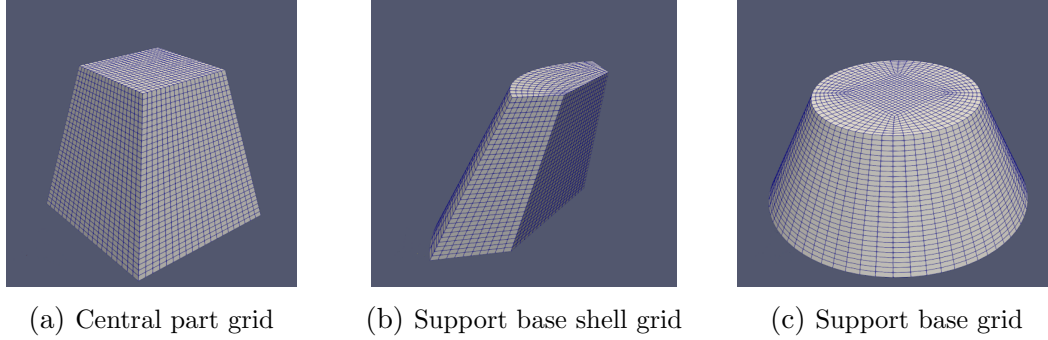
**Figure 1.** Construction of lower and upper platforms grids

In order to construct a rectangular grid to model the cylindrical supports, a block grid approach was employed. The study area was divided into five parts. The central part is a rectangular parallelepiped (Fig. 2a), identical to that used to model the platforms of the structure. The remaining four parts are shells, with one side coinciding node to node with the central part and the other side forming a 90-degree arc of a circle (Fig. 2b).



**Figure 2.** Construction of cylindrical supports grids

It was decided to do the same with the column bases expanding in diameter. The central part is a truncated quadrilateral pyramid, and the geometric configuration of the shells and the resulting appearance of the truncated cone are illustrated in Fig. 3. Furthermore, the upper and lower portions of the columns at the junction of the grids also align node to node at identical radii. Free boundary conditions were used on the boundaries of all grids.



**Figure 3.** Construction of support base grids

## 4. Decomposition Method

When the resulting grid model is parallelized, a considerable number of messages are exchanged between MPI processes. To reduce communication time and enhance the speedup, a grid decomposition method was developed in this work.

Let the model contain  $N$  rectilinear grids and  $M$  curvilinear grids,  $n_i$  is the number of nodes in the  $i$ -th rectilinear grid,  $m_j$  is the number of nodes in the  $j$ -th curvilinear grid and  $P$  is the number of MPI processes. The calculation of curvilinear grids takes more time, thus we introduce an additional coefficient  $a$ . The algorithm consists of the following steps:

1. The number of nodes for each MPI process is calculated

$$M_{opt} = \frac{\sum_{i=1}^N n_i + a \cdot \sum_{j=1}^M m_j}{P}.$$

2. The number of processes allocated to grids of size greater than  $M_{opt}$  is defined as the quotient of their size and  $M_{opt}$  rounded down

$$P_i = \left[ \frac{N_i}{M_{opt}} \right].$$

3. The large grids are divided between the selected processes by a straightforward geometric decomposition method.
4. The remaining grids are distributed among the processes using a greedy algorithm. Initially, the grids are organised into groups, with only one grid in each group. Then at each step two smallest groups are merged into one, continuing until the number of groups is equal to the number of remaining processes.

Figure 4 illustrates an example of the decomposition of the computational domain. In this example, 32 processors were used, with the colour indicating the rank of the process. Figure 4a is a standard decomposition in which each grid is distributed among all MPI processes. Figure 4b is an example of decomposition using the developed algorithm. As can be seen from the figures,

the decomposition algorithm allows for an increase in the granularity of the partitioning of the computational grids.

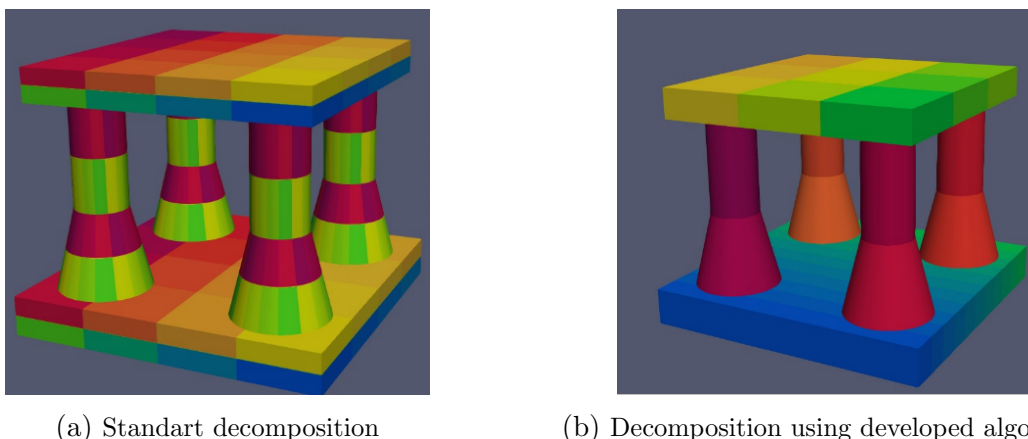


Figure 4. Decomposition of the computational domain. Color indicates the rank of the process

## 5. Results

### 5.1. Speedup

To ascertain the efficacy of the decomposition method, speedup calculations were conducted. The findings are presented in Tab. 1 and Fig. 5. As evidenced, the deployment of the decomposition method does not yield a substantial increase in speedup.

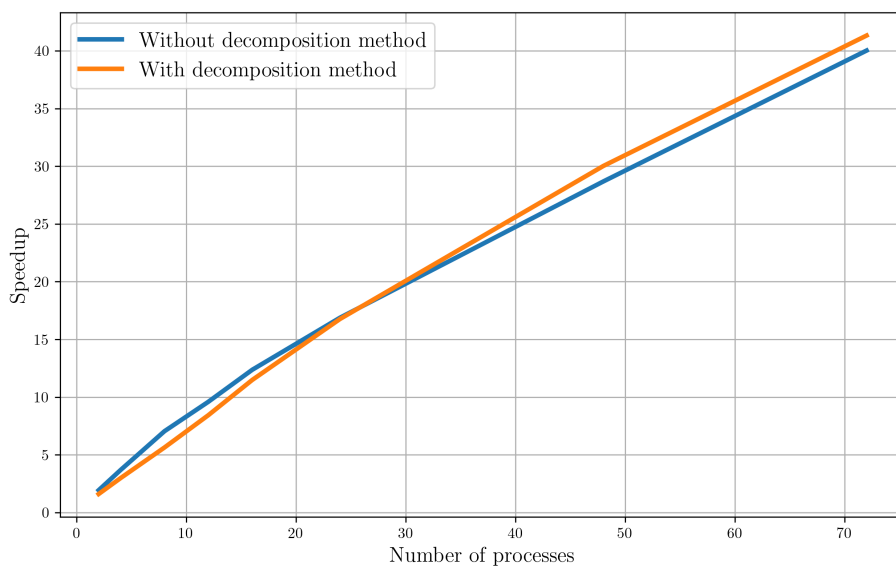


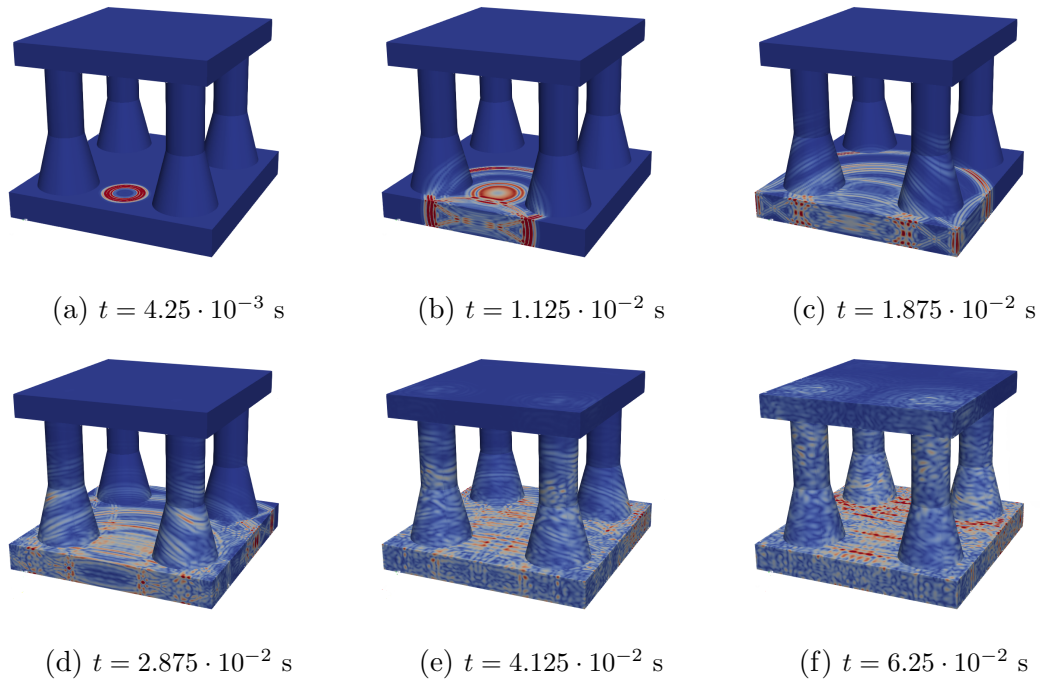
Figure 5. Graph of speedup dependence on the number of processes

**Table 1.** Comparison of speedup with and without decomposition algorithm

Number of processes	Speedup without decomposition method	Speedup with decomposition method
2	1.94	1.61
4	3.69	2.99
8	7.05	5.64
12	9.60	8.44
16	12.37	11.49
24	16.88	16.77
48	28.68	30.02
72	40.02	41.32

## 5.2. Calculation of Disturbance Propagation

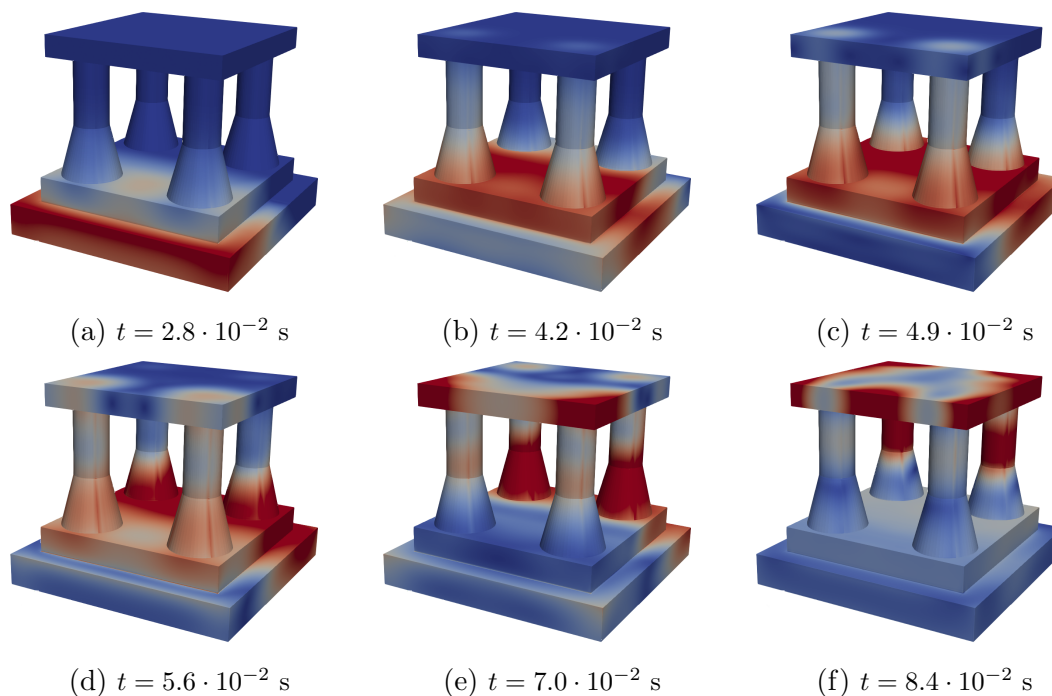
In order to verify the performance of the program, a series of test calculations with different sources of disturbances were performed. Figure 6 presents a numerical simulation of disturbance propagation within an oil structure. The simulation employs an explosion at the base of the platform between supports as the source of disturbance, the amplitude change over time is described by a Ricker pulse with a frequency of 100 Hz.



**Figure 6.** Wave disturbance propagation in the oil platform structure following an explosion at the platform base at various time points

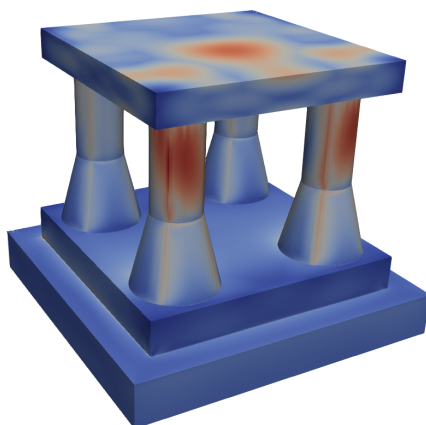
Additionally, one of the important directions of research in this area is to assess the structural resilience to seismic events. For this purpose, we incorporated a representation of the Earth, a rectangular parallelepiped of larger dimensions, into the computational domain. A plane wave emanating from under the ground will act as a perturbation. The pulse profile aligns with the Ricker pulse at a frequency of 10 Hz. The result of the modeling is illustrated in Fig. 7.





**Figure 7.** Wave disturbance propagation in the oil platform structure during an earthquake at various time points

In order to assess the strength characteristics of the oil platform, it is also important to consider the distribution of stress intensity within the structure. When this value exceeds a certain threshold value (which depends on the material in question), the body in question will transition from an elastic state to a plastic state. This results in deformations that are no longer reversible. Figure 8 shows that the highest intensity is observed in the upper parts of the supports, which is opposite from the source. Additionally, a surge of stress intensity also appears in the central part of the tops.



**Figure 8.** Investigation of stress distribution in the oil platform structure. The places of the highest intensity of stresses are highlighted by color

## Conclusion

In this study, a grid model of an oil platform was constructed to simulate the propagation of wave disturbances within its structure. Additionally, a greedy grid decomposition method was developed to increase the computational grid's granularity and accelerate the computational process.

In order to evaluate the strength characteristics of the structure, calculations were performed to assess the impact of potential disturbance sources, such as explosion and seismic loading. During the test calculations, no defects were detected, indicating that this grid construction approach is applicable to the problem. Tests have shown that the use of the developed method of grid decomposition does not result in a significant speedup in comparison with the default decomposition. One potential explanation for this observation is the unequal distribution of load among the processes.

The next step will be to conduct a detailed analysis of the obtained results, focusing on identifying the factors that may limit the speedup potential.

## Acknowledgements

The research was supported by the Russian Science Foundation grant no. 20-71-10028, <https://rscf.ru/project/20-71-10028/>. This work has been carried out using computing resources of the federal collective usage center Complex for Simulation and Data Processing for Mega-science Facilities at NRC “Kurchatov Institute”, <http://ckp.nrcki.ru/>.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Ali, Z., Tucker, P.G., Shahpar, S.: Optimal mesh topology generation for CFD. *Computer Methods in Applied Mechanics and Engineering* 317, 431–457 (2017). <https://doi.org/10.1016/j.cma.2016.12.001>
2. Andrade, X., Alberdi-Rodriguez, J., Strubbe, D.A., *et al.*: Time-dependent density-functional theory in massively parallel computer architectures: the octopus project. *Journal of Physics: Condensed Matter* 24(23), 233202 (2012). <https://doi.org/10.1088/0953-8984/24/23/233202>
3. Basermann, A., Clinckemallie, J., Coupeuz, T., *et al.*: Dynamic load-balancing of finite element applications with the drama library. *Applied Mathematical Modelling* 25(2), 83–98 (2000). [https://doi.org/10.1016/S0307-904X\(00\)00043-3](https://doi.org/10.1016/S0307-904X(00)00043-3)
4. Buluç, A., Meyerhenke, H., Safro, I., *et al.*: *Recent Advances in Graph Partitioning*, pp. 117–158. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-49487-6\\_4](https://doi.org/10.1007/978-3-319-49487-6_4)
5. Cacace, S., Cristiani, E., Falcone, M., Picarelli, A.: A patchy dynamic programming scheme for a class of Hamilton–Jacobi–Bellman equations. *SIAM Journal on Scientific Computing* 34(5), A2625–A2649 (2012). <https://doi.org/10.1137/110841576>

6. Cacace, S., Falcone, M., *et al.*: A dynamic domain decomposition for the eikonal-diffusion equation. *Discret Contin Dyn S* 9(1), 109–123 (2016). <https://doi.org/10.3934/dcdss.2016.9.109>
7. Çatalyürek, Ü., Devine, K., Faraj, M., *et al.*: More recent advances in (hyper) graph partitioning. *ACM Computing Surveys* 55(12), 1–38 (2023). <https://doi.org/10.1145/3571808>
8. Chaplygin, A.V., Gusev, A.V.: Shallow water model using a hybrid MPI/OpenMP parallel programming. *Probl Inf* 1(50), 65–82 (2021). <https://doi.org/10.24411/2073-0667-2021-10006>
9. Devine, K., Boman, E., Heaphy, R., *et al.*: Zoltan data management services for parallel dynamic applications. *Computing in Science & Engineering* 4(2), 90–96 (2002). <https://doi.org/10.1109/5992.988653>
10. Farrashkhalvat, M., Miles, J.P.: *Basic Structured Grid Generation: With an introduction to unstructured grid generation*. Elsevier (2003). <https://doi.org/10.1016/B978-0-7506-5058-8.X5000-X>
11. Favorskaya, A., Khokhlov, N., Sagan, V., Podlesnykh, D.: Parallel computations by the grid-characteristic method on Chimera computational grids in 3D problems of railway non-destructive testing. In: *Russian Supercomputing Days*, pp. 199–213. Springer (2022). [https://doi.org/10.1007/978-3-031-22941-1\\_14](https://doi.org/10.1007/978-3-031-22941-1_14)
12. Favorskaya, A.V., Zhdanov, M.S., Khokhlov, N.I., Petrov, I.B.: Modelling the wave phenomena in acoustic and elastic media with sharp variations of physical properties using the grid-characteristic method. *Geophysical Prospecting* 66(8), 1485–1502 (2018). <https://doi.org/10.1111/1365-2478.12639>
13. Fofanov, V., Khokhlov, N.: Optimization of load balancing algorithms in parallel modeling of objects using a large number of grids. In: *Supercomputing: 6th Russian Supercomputing Days, RuSCDays 2020, Moscow, Russia, September 21–22, 2020, Revised Selected Papers 6. Communications in Computer and Information Science*, vol. 1331, pp. 63–73. Springer (2020). [https://doi.org/10.1007/978-3-030-64616-5\\_6](https://doi.org/10.1007/978-3-030-64616-5_6)
14. Golovchenko, E.N., Yakobovskiy, M.V.: Parallel partitioning tool GridSpiderPar for large mesh decomposition. *Numerical Methods and Programming* 16(4), 507–517 (2015). <https://doi.org/10.26089/NumMet.v16r448>
15. Ivanov, A.M., Khokhlov, N.I.: Efficient inter-process communication in parallel implementation of grid-characteristic method. In: *Smart Modeling for Engineering Systems: Proceedings of the Conference 50 Years of the Development of Grid-Characteristic Method*. pp. 91–102. Springer (2019). [https://doi.org/10.1007/978-3-030-06228-6\\_9](https://doi.org/10.1007/978-3-030-06228-6_9)
16. Ivanov, A.M., Khokhlov, N.I., *et al.*: Parallel implementation of the grid-characteristic method in the case of explicit contact boundaries. *Computer research and modeling* 10(5), 667–678 (2018). <https://doi.org/10.20537/2076-7633-2018-10-5-667-678>
17. Karypis, G., Kumar, V.: *A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*. University of

- Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN 38, 7–1 (1998)
18. Karypis, G., Schloegel, K., Kumar, V.: Parmetis: Parallel graph partitioning and sparse matrix ordering library (1997)
  19. Larour, E., Seroussi, H., Morlighem, M., Rignot, E.: Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM). *Journal of Geophysical Research: Earth Surface* 117(F1) (2012). <https://doi.org/10.1029/2011JF002140>
  20. Muratov, R.V., Ryabov, P.N., Dyachkov, S.A.: Dynamic domain decomposition method based on weighted Voronoi diagrams. *Computer Physics Communications* 290, 108790 (2023). <https://doi.org/10.1016/j.cpc.2023.108790>
  21. Palmroth, M., Ganse, U., Pfau-Kempf, Y., *et al.*: Vlasov methods in space physics and astrophysics. *Living reviews in computational astrophysics* 4(1), 1 (2018). <https://doi.org/10.1007/s41115-018-0003-2>
  22. Petrov, I.B., Favorskaya, A.V., Khokhlov, N.I.: Grid-characteristic method on embedded hierarchical grids and its application in the study of seismic waves. *Computational Mathematics and Mathematical Physics* 57, 1771–1777 (2017). <https://doi.org/10.1134/S0965542517110112>
  23. Rao, A.R.M.: Distributed evolutionary multi-objective mesh-partitioning algorithm for parallel finite element computations. *Computers & Structures* 87(23-24), 1461–1473 (2009). <https://doi.org/10.1016/j.compstruc.2009.05.006>
  24. Rao, A.R.M.: Parallel mesh-partitioning algorithms for generating shape optimised partitions using evolutionary computing. *Advances in Engineering Software* 40(2), 141–157 (2009). <https://doi.org/10.1016/j.advengsoft.2008.03.017>
  25. Rao, A.R.M., Rao, T.A., Dattaguru, B.: A new parallel overlapped domain decomposition method for nonlinear dynamic finite element analysis. *Computers & Structures* 81(26-27), 2441–2454 (2003). [https://doi.org/10.1016/S0045-7949\(03\)00312-2](https://doi.org/10.1016/S0045-7949(03)00312-2)
  26. Walshaw, C., Cross, M.: JOSTLE: parallel multilevel graph-partitioning software—an overview. *Mesh partitioning techniques and domain decomposition techniques* 10, 27–58 (2007)
  27. Xia, Z.: Review of architecture and model for parallel programming. *Applied and Computational Engineering* 8, 455–461 (2023). <https://doi.org/10.54254/2755-2721/8/20230225>
  28. Yang, X., Shan, J.L., Yu, F., *et al.*: Boundary constrained quadrilateral mesh generation based on domain decomposition and templates. *Computers & Structures* 295, 107275 (2024). <https://doi.org/10.1016/j.compstruc.2024.107275>
  29. Zhou, Y., Cai, X., Zhao, Q., *et al.*: Quadrilateral mesh generation method based on convolutional neural network. *Information* 14(5), 273 (2023). <https://doi.org/10.3390/info14050273>