

Supercomputing Frontiers and Innovations

2024, Vol. 11, No. 4

Scope

- Future generation supercomputer architectures
- Exascale computing
- Parallel programming models, interfaces, languages, libraries, and tools
- Supercomputer applications and algorithms
- Novel approaches to computing targeted to solve intractable problems
- Convergence of high performance computing, machine learning and big data technologies
- Distributed operating systems and virtualization for highly scalable computing
- Management, administration, and monitoring of supercomputer systems
- Mass storage systems, protocols, and allocation
- Power consumption minimization for supercomputing systems
- Resilience, reliability, and fault tolerance for future generation highly parallel computing systems
- Scientific visualization in supercomputing environments
- Education in high performance computing and computational science

Editorial Board

Editors-in-Chief

- **Jack Dongarra**, University of Tennessee, Knoxville, USA
- **Vladimir Voevodin**, Moscow State University, Russia

Editorial Director

- **Leonid Sokolinsky**, South Ural State University, Chelyabinsk, Russia

Associate Editors

- **Pete Beckman**, Argonne National Laboratory, USA
- **Arndt Bode**, Leibniz Supercomputing Centre, Germany
- **Boris Chetverushkin**, Keldysh Institute of Applied Mathematics, RAS, Russia
- **Alok Choudhary**, Northwestern University, Evanston, USA
- **Alexei Khokhlov**, Moscow State University, Russia
- **Thomas Lippert**, Jülich Supercomputing Center, Germany

- **Satoshi Matsuoka**, Tokyo Institute of Technology, Japan
- **Mark Parsons**, EPCC, United Kingdom
- **Thomas Sterling**, CREST, Indiana University, USA
- **Mateo Valero**, Barcelona Supercomputing Center, Spain

Subject Area Editors

- **Artur Andrzejak**, Heidelberg University, Germany
- **Rosa M. Badia**, Barcelona Supercomputing Center, Spain
- **Franck Cappello**, Argonne National Laboratory, USA
- **Barbara Chapman**, University of Houston, USA
- **Yuefan Deng**, Stony Brook University, USA
- **Ian Foster**, Argonne National Laboratory and University of Chicago, USA
- **Geoffrey Fox**, Indiana University, USA
- **William Gropp**, University of Illinois at Urbana-Champaign, USA
- **Erik Hagersten**, Uppsala University, Sweden
- **Michael Heroux**, Sandia National Laboratories, USA
- **Torsten Hoefler**, Swiss Federal Institute of Technology, Switzerland
- **Yutaka Ishikawa**, AICS RIKEN, Japan
- **David Keyes**, King Abdullah University of Science and Technology, Saudi Arabia
- **William Kramer**, University of Illinois at Urbana-Champaign, USA
- **Jesus Labarta**, Barcelona Supercomputing Center, Spain
- **Alexey Lastovetsky**, University College Dublin, Ireland
- **Yutong Lu**, National University of Defense Technology, China
- **Bob Lucas**, University of Southern California, USA
- **Thomas Ludwig**, German Climate Computing Center, Germany
- **Daniel Mallmann**, Jülich Supercomputing Centre, Germany
- **Bernd Mohr**, Jülich Supercomputing Centre, Germany
- **Onur Mutlu**, Carnegie Mellon University, USA
- **Wolfgang Nagel**, TU Dresden ZIH, Germany
- **Alexander Nemukhin**, Moscow State University, Russia
- **Edward Seidel**, National Center for Supercomputing Applications, USA
- **John Shalf**, Lawrence Berkeley National Laboratory, USA
- **Rick Stevens**, Argonne National Laboratory, USA
- **Vladimir Sulimov**, Moscow State University, Russia
- **William Tang**, Princeton University, USA
- **Michela Taufer**, University of Delaware, USA
- **Andrei Tchernykh**, CICESE Research Center, Mexico
- **Alexander Tikhonravov**, Moscow State University, Russia
- **Eugene Tyrtshnikov**, Institute of Numerical Mathematics, RAS, Russia
- **Roman Wyrzykowski**, Czestochowa University of Technology, Poland
- **Mikhail Yakobovskiy**, Keldysh Institute of Applied Mathematics, RAS, Russia

Technical Editors

- **Andrey Goglachev**, South Ural State University, Chelyabinsk, Russia
- **Yana Kraeva**, South Ural State University, Chelyabinsk, Russia
- **Dmitry Nikitenko**, Moscow State University, Moscow, Russia
- **Mikhail Zymbler**, South Ural State University, Chelyabinsk, Russia

Contents

Simulation of Seismic Processes with High-Order Grid-Characteristic Methods V.I. Golubev, A.V. Shevchenko	4
Numerical Modeling of Complex Geometry Thin Composite Structures under Vibrational Testing S.A. Lavrenkov, I.E. Smirnov, D.A. Kravchenko, K.A. Beklemysheva, A.V. Vasyukov	15
Numerical Modeling of Marine Seismology in the Arctic Region During Deposit Dissolution due to Oil Migration E.K. Guseva, V.I. Golubev, I.B. Petrov	29
A Modification of Adaptive Greedy Algorithm for Solving Problems of Fractured Media Geophysics A.V. Favorskaya, N.I. Khokhlov, D.A. Podlesnykh	40
On an Algorithm for Decomposing Multi-Block Structured Meshes for Calculating Dynamic Wave Processes in Complex Structures on Supercomputers with Distributed Memory I.N. Agrelov, N.I. Khokhlov, V.O. Stetsyuk, S.D. Agibalov	54
Parallelization Strategies for Ultrasonic Wave Propagation in Composite Materials Considering Microstructural Details E. Pesnya, A.V. Favorskaya, I.B. Petrov, N.I. Khokhlov	66
Leveraging OpenMP Tasks for Efficient Parallel Modeling of the Elastic Eave Propagation in Multi-mesh Problems N.I. Khokhlov, V.O. Stetsyuk	78
HPC Optimization Algorithm for Assessing Haemodynamic Parameters in Synthetic Patient Cohorts A.V. Rogov, T.M. Gamilov, Y.Yu. Kirichenko, P.Yu. Kopylov, S.S. Simakov	92



This issue 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.

Simulation of Seismic Processes with High-Order Grid-Characteristic Methods

Vasily I. Golubev¹ , Alexey V. Shevchenko^{1,2} 

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

This paper considers simulation of the seismic wave propagation in geological media with different rheological properties. The present work aims to construct a numerical scheme to model porous fluid-saturated medium, for the description of which the Dorovsky model was selected. We employed the grid-characteristic method, which includes choosing the appropriate operator splitting method for a 3D problem, deriving the transformation to the Riemann invariants analytically, and explicitly setting boundary and contact conditions. We simulated two scenarios. Firstly, we compared the wavefields generated by a point-source in the acoustic, linear elastic, and porous fluid-saturated approximations, noting the similarities in the longitudinal wave and differences in other wave types. Secondly, we simulated a part of the marine seismic survey process, including a source in the water layer, governed by the acoustic equations, a water-saturated layer described by the Dorovsky equations, and an explicit contact between these layers. To utilize the modern HPC multi-core and multi-processor systems, the hybrid MPI-OpenMP parallel algorithms were used.

Keywords: Dorovsky model, grid-characteristic method, contact conditions.

Introduction

The task of an accurate modelling of wave processes in deformable media is important in multiple different applications, such as non-destructive testing of composite materials, medical ultrasound imaging, seismic survey process, earthquake simulation, etc. If simple models prove to be insufficient, more complex and accurate ones are employed. It should be noted, that since analytical solutions are not available for most valuable mechanical problems, computer simulations are required.

Let us concentrate on the seismic wave propagation problem. The simplest model is the linear acoustic model [1, 10]. Despite its derivation for fluids, it has gained significant popularity, being a reasonably good choice for simulating only longitudinal waves in geological media. However, other types of waves (transverse waves, complex surface waves) often have to be taken into account, thus requiring more sophisticated models like the isotropic linear elastic model [1]. In other cases, more sophisticated properties need to be considered, for example, anisotropy. Another example of a medium with a complex internal structure is a porous fluid-saturated medium. One of its distinctive physical properties is the existence of the second longitudinal wave with a smaller velocity. Several mathematical models are used to describe it, the most popular one being the Biot model [3]. However, other models like the Dorovsky model [4, 11] are available as well. In our work we choose the Dorovsky model because it is governed by a hyperbolic system of PDEs and it only has three elastic parameters that uniquely define (or are defined by) the three velocities of wave propagation.

Since solving larger equations requires more computational resources, researchers attempt to use simpler models of the medium whenever possible. One option to do that is to combine several models in one simulation, which allows us to conduct highly accurate modelling and save

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

²Ishlinsky Institute for Problems in Mechanics RAS, Moscow, Russian Federation

significant resources at the same time. Usually, each model is set on a separate grid; therefore, special contact conditions between those grids are required. These conditions have to be both physically and mathematically correct to ensure convergence towards the correct solution. Generally, such conditions do not depend upon the chosen numerical method; some examples can be found in the papers [6–8, 10].

Different numerical methods can be used to solve the governing equations. Most popular ones are the finite-difference and finite-element methods, but others can be used, too. In this work we use the grid-characteristic method, which is described in more details in the papers [5, 6, 8].

This paper has the following structure. Firstly, we describe the physical and mathematical models used in this study. Then we present the developed numerical method, including handling internal and boundary points and implementation of contact conditions. Finally, we show the results of several computer simulations.

1. Mathematical Models

Different models can be used for describing the process of seismic wave propagation in geological media. Here we only consider those which can take into account the wave fronts; they are usually represented in the form of systems of partial differential equations with necessary initial and boundary conditions.

1.1. Acoustic Equations

One of the basic models is the linear acoustic model. The equations are as follows [1, 10]:

$$\begin{cases} \rho \vec{v}_t &= -\nabla p + \vec{F}, \\ p_t &= -c^2 \rho \nabla \cdot \vec{v}. \end{cases} \quad (1)$$

Here, $p = p(x, y, z, t)$ is the acoustic pressure, $\vec{v} = \vec{v}(x, y, z, t)$ is the particle velocity (the derivative of the local displacement vector with respect to time). Force vector $\vec{F}(x, y, z, t)$ denotes the right-hand side, the applied external volumetric force. The known parameters (generally, space-dependent) are density ρ [kg/m³] and wave speed c [m/s] ($c = \sqrt{K/\rho}$, where K is the bulk modulus).

Zero initial conditions are used, since unknowns p and \vec{v} are relative to the steady state. One boundary condition is required; for instance, given pressure $P(x, y, z, t)|_{x,y,z \in \partial E}$. Here the computational domain is denoted as E and its boundary – as ∂E . Boundary condition $P \equiv 0$ sets the free surface, which is commonly used on top boundary.

Acoustic equations (1) accurately describe low-amplitude pressure waves (propagating with the speed c) in fluids.

1.2. Equations of Linear Elasticity

The system of PDEs for the linear elastic model is presented below [1, 5]:

$$\begin{cases} \rho \vec{v}_t = (\nabla \cdot \mathbf{T})^\top + \vec{F}, \\ \mathbf{T}_t = \lambda(\nabla \cdot \vec{v})\mathbf{I} + \mu(\nabla \otimes \vec{v} + (\nabla \otimes \vec{v})^\top). \end{cases} \quad (2)$$

Here the unknowns are: the particle velocity $\vec{v} = \vec{v}(x, y, z, t)$ (the derivative of the local displacement vector with respect to time) and the symmetric stress tensor T , which has 6 independent components $T_{xx}, T_{yy}, T_{zz}, T_{xz}, T_{yz}, T_{zx}$. Force $\vec{F} = \vec{F}(x, y, z, t)$ denotes the right-hand side. Density ρ [kg/m³] and two Lamé parameters λ, μ [Pa] are the known possibly space-dependent medium parameters. In the equation (2) \otimes stands for $\vec{a} \otimes \vec{b} = \mathbf{A}_{ij}, A_{ij} = a_i \cdot b_j$ and $\nabla \cdot \vec{v} = \text{div } \vec{v}$.

Here we also use zero initial conditions. The following boundary condition is typically imposed: $T \cdot \vec{n} = \vec{f}(x, y, z, t)$, where \vec{n} is a unit vector normal to the boundary ∂E .

Elastic equations (2) accurately describe body waves: longitudinal wave propagating with the speed $c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and shear wave propagating with the speed $c_s = \sqrt{\frac{\mu}{\rho}}$, surface waves: Rayleigh wave, Love wave.

1.3. Dorovsky Model Equations

The following system of PDEs governs wave propagation in the Dorovsky model [4, 8, 11]:

$$\begin{cases} \vec{u}_t + \frac{1}{\rho_s}(\nabla \cdot \mathbf{h})^\top + \frac{1}{\rho_0}\nabla p = \vec{F}, \\ \vec{v}_t + \frac{1}{\rho_0}\nabla p = \vec{F}, \\ \mathbf{h}_t + \mu(\nabla \otimes \vec{u} + (\nabla \otimes \vec{u})^\top) + \left[\left(\lambda - \frac{\rho_s}{\rho_0} K \right) (\nabla \cdot \vec{u}) - \frac{\rho_f}{\rho_0} K (\nabla \cdot \vec{v}) \right] \mathbf{I} = 0, \\ p_t - (K - \alpha \rho_0 \rho_s)(\nabla \cdot \vec{u}) + \alpha \rho_0 \rho_f (\nabla \cdot \vec{v}) = 0. \end{cases} \quad (3)$$

In each infinitesimal medium volume there is a rigid skeleton and interconnected pores saturated with a fluid (typically water or oil for geophysical applications). The unknown functions in this system are: skeleton velocity (derivative of displacement with respect to time) $\vec{u} = \vec{u}(x, y, z, t)$, fluid velocity (derivative of displacement with respect to time) $\vec{v} = \vec{v}(x, y, z, t)$, minus stress tensor of the rigid skeleton $\mathbf{h} = \mathbf{h}(x, y, z, t)$, pore fluid pressure $p = p(x, y, z, t)$. Force \vec{F} denotes the right-hand side.

The medium is described by the following parameters: $\rho_s = (1 - \beta)\rho_{s0}$, $\rho_f = \beta\rho_{f0}$, $\rho_0 = \rho_s + \rho_f$, where ρ_{s0}, ρ_{f0} are real physical densities of the rigid skeleton and the saturating fluid, respectively; β is the volumetric porosity of the medium defined by $\beta = \frac{V_{pores}}{V_{total}}$.

The remaining medium parameters K, μ, α are the elastic parameters of the saturated medium which are defined based on c_{p1}, c_{p2}, c_s (explained below) and already given ρ_0, ρ_s, ρ_f by the following relations:

$$\mu = \rho_s c_s^2, \quad (4)$$

$$K = \frac{\rho_0 \rho_s}{2\rho_f} \left(c_{p1}^2 + c_{p2}^2 - \frac{8}{3} \frac{\rho_f}{\rho_0} c_s^2 - \sqrt{(c_{p1}^2 - c_{p2}^2)^2 - \frac{64}{9} \frac{\rho_f \rho_s}{\rho_0^2} c_s^4} \right), \quad (5)$$

$$\alpha_3 = \frac{1}{2\rho_0^2} \left(c_{p1}^2 + c_{p2}^2 - \frac{8}{3} \frac{\rho_s}{\rho_0} c_s^2 + \sqrt{(c_{p1}^2 - c_{p2}^2)^2 - \frac{64}{9} \frac{\rho_f \rho_s}{\rho_0^2} c_s^4} \right), \quad (6)$$

$$\alpha = \rho_0 \alpha_3 + K/\rho_0^3. \quad (7)$$

Known possibly space-dependent medium parameters c_{p1}, c_{p2}, c_s are the wave velocities of the first (fast) longitudinal wave, second (slow) longitudinal wave and shear wave, respectively. So, the Dorovsky model describes three kinds of body waves: two longitudinal ones and a shear wave. It also describes surface waves, like the linear elastic model.

We use zero initial conditions and the following boundary condition:

$$\begin{cases} p = P(x, y, z, t), & x, y, z \in \partial E, t > 0, \\ \mathbf{h} \cdot \vec{n} \cdot \vec{n} = H(x, y, z, t), & x, y, z \in \partial E, t > 0, \\ \mathbf{h} \cdot \vec{n} \cdot \vec{\tau} = 0, & x, y, z \in \partial E, t > 0. \end{cases} \quad (8)$$

Here \vec{n} is a unit normal to the boundary, $\vec{\tau}$ is a unit vector tangential to the boundary (two linearly independent vectors in 3D case). Note that setting $P \equiv 0$, $H \equiv 0$ results in the free surface boundary condition in the case of open pores.

2. Numerical Method

Systems of PDEs (1), (2), and (3) are first-order systems that can be easily written in the matrix form

$$\vec{q}_t + A_x \vec{q}_x + A_y \vec{q}_y + A_z \vec{q}_z = \vec{f}, \quad (9)$$

where vector $\vec{q} = \vec{q}(x, y, z, t)$ contains all unknown functions, while known matrices A_x , A_y , A_z are defined by the combination of the appropriate material parameters. Initial conditions $\vec{q}^{(0)} = \vec{0}$. Boundary conditions can be generally written as a linear relation [6, 8]

$$B \cdot \vec{q}(x, y, z, t) = \vec{b}(x, y, z, t), \quad x, y, z \in \partial E, t > 0, \quad (10)$$

and will be discussed in more detail in subsection 2.3.

Below we describe construction of the numerical scheme for the above equations using the grid-characteristic method (GCM).

2.1. Operator Splitting

First of all, we want to simplify this three-dimensional system ($x, y, z \in \mathbb{R}^3$) by consecutive solutions of one-dimensional problems. This can be achieved by a coordinate splitting procedure. For an operator splitting scheme with s stages we need to perform the following procedure (see [9] for details in the 2D case):

```
for i in 1, ..., s:
    perform stepX( $\alpha_i^x \cdot dt$ )
    perform stepY( $\alpha_i^y \cdot dt$ )
    perform stepZ( $\alpha_i^z \cdot dt$ )
```

Here, *perform stepX* means find $\vec{q}^{(n+1)}$ for \vec{q} satisfying the system $\vec{q}_t + A_x \vec{q}_x = \vec{0}$ and $t^{(n+1)} - t^{(n)} = \alpha_i^x \cdot dt$. For stepY and stepZ the similar formulae are applied with the appropriate matrix (A_y or A_z) and spatial derivative ($\frac{\partial}{\partial y}$ or $\frac{\partial}{\partial z}$) with the values from the previously done step as the starting $\vec{q}^{(n)}$.

Thus, an operator splitting scheme is defined by the coefficients α_i^j , $1 \leq i \leq s$, $j \in \{x, y, z\}$. Several examples are given in Tables 1–3. Although the last scheme theoretically demonstrates the highest approximation order, we chose the first scheme for our simulations in this work due to its high efficiency (one stage, $s = 1$) and relatively good performance. Also note that if a high-order scheme is used, one might need smaller dt since $\tau = \alpha_i^j dt$ should satisfy the stability condition, and some $|\alpha_i^j| > 1$. For instance, with the Y7-4 scheme one needs to make 13/3 times more one-dimensional steps for each 3D time step, and the dt value must be decreased approximately $\max_{i,j} |\alpha_i^j| \approx 1.7$ times, increasing the total number of time steps accordingly.

Therefore, a simulation with the high-order Y7-4 scheme takes 7.4 times longer than the one with the XYZ scheme on the same mesh.

The application of the right-hand side can be viewed as another operator in the splitting procedure. However, since in the problems considered right-hand side is usually a point source limited in time, we apply it only once per time step after the homogeneous equations have been solved by the operator splitting technique with respect to spatial coordinates.

Table 1. Coefficients of the XYZ splitting scheme: $s = 1$, 1st order of approximation

i	α_i^X	α_i^Y	α_i^Z
1	1	1	1

Table 2. Coefficients of the 3D Strang splitting scheme [2]: $s = 3$, 2nd order of approximation

i	α_i^X	α_i^Y	α_i^Z
1	0	0	1/2
2	0	1/2	0
3	1	1/2	1/2

Table 3. Coefficients of the Y7-4 splitting scheme [2]: $s = 7$, 4th order of approximation,

$$\phi_1 = \frac{1}{2-\sqrt[3]{2}}, \phi_2 = -\frac{\sqrt[3]{2}}{2-\sqrt[3]{2}}$$

i	α_i^X	α_i^Y	α_i^Z
1	0	0	$\phi_1/2$
2	0	$\phi_1/2$	0
3	ϕ_1	$\phi_1/2$	$(\phi_1 + \phi_2)/2$
4	0	$\phi_2/2$	0
5	ϕ_2	$\phi_2/2$	$(\phi_1 + \phi_2)/2$
6	0	$\phi_1/2$	0
7	ϕ_1	$\phi_1/2$	$\phi_1/2$

2.2. One-Dimensional Case

Consider a one-dimensional system of equations, resulting from the operator splitting procedure defined above:

$$\vec{q}_t + A_x \vec{q}_x = \vec{0}. \tag{11}$$

Since original systems of PDEs (1), (2), and (3) are hyperbolic, matrix A_x has a full set of eigenvectors. It is correct for A_y and A_z too. We combine all eigenvectors as columns of a matrix and denote that matrix as Ω^{-1} – an inverse to a matrix Ω ; we can do that since Ω^{-1} has full rank. Then, if we put all eigenvalues corresponding to eigenvectors in Ω^{-1} into the diagonal of a diagonal matrix Λ , the following relation will be valid: $A_x = \Omega^{-1}\Lambda\Omega$. Let us multiply (11) by the matrix Ω on the left:

$$\Omega \cdot \vec{q}_t + \Omega \cdot \Omega^{-1}\Lambda\Omega \cdot \vec{q}_x = \vec{0}, \tag{12}$$

$$(\Omega\vec{q})_t + \Lambda\Omega\vec{q}_x = \vec{0}. \tag{13}$$

Assuming matrix Ω to be constant and not depending on x , we obtain

$$(\Omega \vec{q})_t + \Lambda (\Omega \vec{q})_x = \vec{0}. \quad (14)$$

Introduce new unknown vector – the so called Riemann invariants $\vec{\omega} = \Omega \vec{q}$, we obtain a system of independents transport equations with respect to each component of $\vec{\omega}$:

$$\vec{\omega}_t + \Lambda \vec{\omega}_x = \vec{0}. \quad (15)$$

For each scalar equation $w_t + aw_x = 0$, where $w = \omega_i$, $a = \lambda_i$, the characteristic property is fulfilled: $w(x_m, t^{(n+1)}) = w(x_m - a\tau, t^{(n)})$, where $\tau = t^{(n+1)} - t^{(n)}$. Value $w_* = w(x_m - a\tau, t^n)$ can be obtained by using some interpolation procedure. Usually, a polynomial interpolation is used; in this work we used a third-order interpolation polynomial built on a four-point stencil in the internal points, resulting in a third-order scheme. This scheme closely resembles the known Rusanov scheme. Note that for positive and for negative λ_i the stencil points used for interpolation are different, which is important to preserve the stability properties of the scheme. Treatment of points near the boundary is discussed in the next subsection.

Finally, when we have computed each $\omega_i^{(n+1)}$ with the appropriate λ_i , we can return to the original unknowns using the formula $\vec{q}^{(n+1)} = \Omega^{-1} \vec{\omega}^{(n+1)}$.

The stability condition is the standard Courant condition $c \cdot dt/h < 1$, where $c = \max_i |\lambda_i|$.

For an efficient implementation we have precomputed matrices Ω and Ω^{-1} analytically. Of course, they depend upon medium properties – parameters of the corresponding PDE. During the simulation at each time step matrix-vector multiplication $\Omega \vec{q}$ and $\Omega^{-1} \vec{\omega}$ is performed quickly and the sparse nature of Ω and Ω^{-1} is taken into account. For larger matrices like the ones resulting from the Dorovsky model (3) utilizing symbolic algebra software like *Wolfram Mathematica* [12] or *Python*'s SymPy library proves to be beneficial.

2.3. Boundary Conditions

The scheme presented above cannot be implemented directly near the boundary because several points in the stencil required for interpolation are located outside the computational domain. In the framework of the grid-characteristic method we extend the computational grid by two ghost points on each side, allowing us to perform the same computations everywhere. Then, in order for the simulation to be accurate, i.e. for the setting of the appropriate boundary conditions, we utilize the following scheme:

1. fill ghost points with values from the closest point on the boundary in order to minimize reflections;
2. perform a special correction to set correct values on the boundary.

The first item approximates zero incoming values from outgoing characteristics – the ones that are outside the computational domain. The second item performs a correction based on the formula [5, 6, 8]:

$$\vec{q}^{(n+1)} = \vec{q}^{in} + \Omega^{out} (B \cdot \Omega^{out})^{-1} \cdot (\vec{b} - B \vec{q}^{in}), \quad (16)$$

where B and \vec{b} are defined in (10), \vec{q}^{in} is \vec{q} after step 1 (with values updated only from those characteristics which stay inside the domain), Ω^{out} is a rectangular matrix, columns of which correspond to the outgoing characteristics. $\Omega^{out} (B \cdot \Omega^{out})^{-1}$ are precomputed analytically.

2.4. Contact Conditions

In many cases the wave propagates through several media with significantly varying properties. Using the most general and accurate mathematical model can be prohibitive for large simulations: for example, for the Dorovsky model (3) we have to store 13 floating-point values per point, which is 3.25 more than for the acoustic model (1), and there are also more operations involved. Also, some numerical discretizations of complex PDEs do not allow setting some of the parameters to zeros to treat special cases. Running one simulation on multiple grids, possibly with different equations on each one, is a popular approach to deal with such issues. In this subsection we present the technique to organize explicit contact between two grids.

Consider two grids representing two media, denoted by lower indices a and b (in this subsection this does not mean differentiation). We cover the computational domain in such a way that the points in grid a which are located exactly on the interface coincide with the points in grid b , which are also located exactly on the interface. Thus, for those physical points we will have two grid nodes: \vec{q}_a in grid a , and \vec{q}_b in grid b . Note that \vec{q}_a and \vec{q}_b can have different sizes and different components from different governing PDEs. In the general form we write the contact condition as

$$C \begin{bmatrix} \vec{q}_a \\ \vec{q}_b \end{bmatrix} = \vec{c}, \quad (17)$$

where C is a rectangular matrix $R \times (n_a + n_b)$, n^a – number of components of \vec{q}_a , n_b – number of components of \vec{q}_b , R – total number of outgoing characteristics for both grids.

The boundary correction (16) can be written in the following form:

$$\vec{u}_i^{(n+1)} = \Phi_i \vec{u}_i^{in} + \vec{f}_i, \quad i \in \{a, b\}. \quad (18)$$

Matrices Φ_i are known, vectors \vec{f}_i are unknown. Substituting (18) into (17), we obtain a system of linear algebraic equations with respect to all components of \vec{f}_i , $i \in \{a, b\}$. The system has a unique solution if the boundary corrector matrices B_i were chosen correctly (for outgoing characteristics).

Thus, we get the following algorithm [5, 6, 8]:

1. run one step of the simulation on both grids for the internal points (and for boundary points using ghost cells), obtain \vec{q}_i^{in} for $i \in \{a, b\}$;
2. solve the small algebraic system (17) with respect to $[\vec{f}_a \ \vec{f}_b]^T$;
3. perform correction according to formula (16) for both grids, computing $\vec{q}_i^{(n+1)}$, $i \in \{a, b\}$.

For the contact between an acoustic medium governed by equations (1) and a fluid-saturated medium governed by the Dorovsky equations (3) the following contact conditions should be implemented [7]:

$$\begin{cases} (-\|\mathbf{h}\| \cdot \vec{n}, \vec{n}) = -P_a, \\ (-\|\mathbf{h}\| \cdot \vec{n}, \vec{\tau}) = 0, \\ (1 - \beta)(\vec{u}, \vec{n}) + \beta(\vec{v}, \vec{n}) = (\vec{V}_a, \vec{n}), \\ \frac{\rho_f}{\beta\rho_0} \cdot p = P_a. \end{cases} \quad (19)$$

Here P_a and \vec{V}_a denote pressure and velocity in the acoustic medium, respectively.

2.5. Parallel Algorithms

This subsection discusses the implemented parallel algorithms for using modern high-performance computing systems. For parallelization in a distributed memory system, MPI technology is used, in a shared memory system – OpenMP technology is used. The utilized grid-characteristic method on structured grids belongs to the class of explicit numerical methods. This allows using classical approaches to parallelization – the principle of geometric parallelism. The computational domain (grid) is divided into parts (blocks) with overlaps, the width of the overlaps is determined by the spatial scheme stencil. At each time step, data is exchanged in the border nodes (in overlapping parts). OpenMP technology (*pragma omp for*) is used to accelerate the basic scheme node loop. Thus, the computational algorithm successfully works at systems with distributed memory, with shared memory and in hybrid systems.

3. Simulation Results

3.1. Homogeneous Space

Our first simulation models wave propagation from a point-source vertical force applied at the centre of cubic computational domain with time dependence in the form of the Ricker impulse with the peak frequency of 30 Hz. The model size was $800 \text{ m} \times 800 \text{ m} \times 800 \text{ m}$. It was covered by a computational grid with step size of 2 m. Time step was 0.8 ms to satisfy the Courant stability condition, 200 time steps were taken. In one simulation, we use the linear elastic model with the parameters $c_p = 2000 \text{ m/s}$, $c_s = 1300 \text{ m/s}$, $\rho = 1450 \text{ kg/m}^3$. In the second case, we used the Dorovsky model with similar parameters: $c_{p1} = 2000 \text{ m/s}$, $c_{p2} = 450 \text{ m/s}$, $c_s = 1300 \text{ m/s}$, $\rho_0 = 1450 \text{ kg/m}^3$, $\rho_f = 100 \text{ kg/m}^3$, $\beta = 10\%$. Result of the simulation is presented in the Fig. 1.

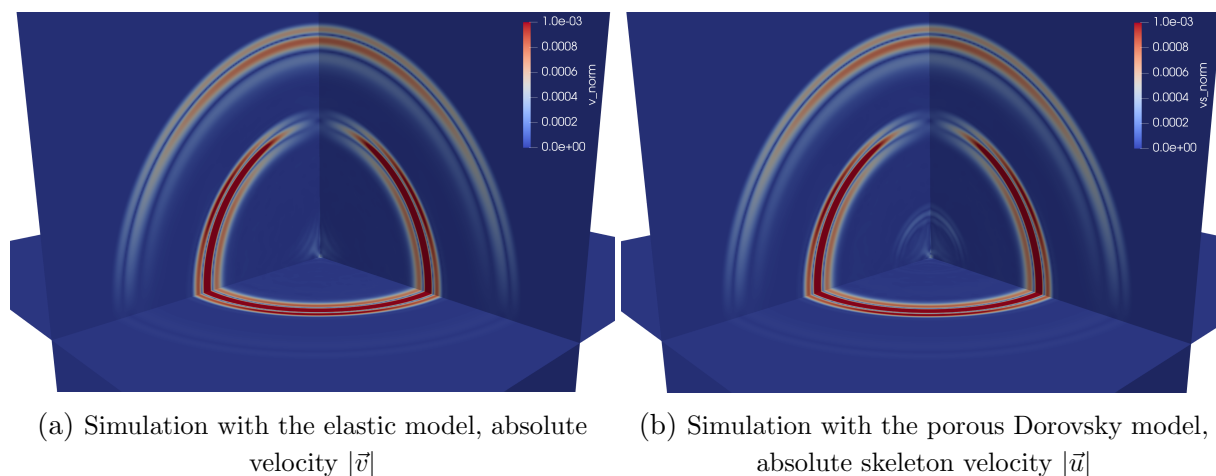


Figure 1. Slices of the absolute velocity. Pressure wave and shear waves are clearly seen in both simulations. The second pressure wave propagates with the low speed $c_{p2} < c_s$ only in the Dorovsky model

3.2. Bottom Sediments

We simulated wave propagation from a point source in a combination of a water layer, described by the acoustic equations (1), and a deformable water-saturated layer of bottom sediments, modelled by the Dorovsky equations (3). The three-dimensional case was considered. We used well-known water parameters: density $\rho = 1000 \text{ kg/m}^3$ and wave velocity $c = 1500 \text{ m/s}$. We used the following parameters for the Dorovsky model: porosity $\beta = 10\%$, physical water density $\rho_{f0} = 1000 \text{ kg/m}^3$, physical skeleton density $\rho_{s0} = 1500 \text{ kg/m}^3$, $\mu = 2.2815e9 \text{ Pa}$, $K = 2.642306867e9 \text{ Pa}$, $\alpha = 2507.905873 \text{ m}^5/(\text{kg} \cdot \text{s}^2)$, corresponding to wave velocities $c_{p1} = 2000 \text{ m/s}$, $c_{p2} = 450 \text{ m/s}$, $c_s = 1300 \text{ m/s}$. The point source was set at the depth of 4 m with the Ricker impulse with the peak frequency of 30 Hz as the time dependence.

The computational domain with the size $700 \text{ m} \times 700 \text{ m} \times 300 \text{ m}$ was covered with two cubic grids with the step of 2 m: one for the water layer (250 m), one for the bottom sediments (50 m). To satisfy the Courant stability condition the time step was set to 0.9 ms. Totally, 250 time steps were taken.

The simulation was carried out on a PC with an Intel(R) Core(TM) i7-10700 CPU 2.9 GHz and 15 GiB RAM. For better utilization of multiple cores on a shared memory system, we used 1 MPI process with 8 OpenMP threads. This 3D simulation took approximately 1.5 minutes.

In Fig. 2 one can see the simulated region consisting of two computational grids in contact with each other. Pressure values are shown in both grids. Longitudinal wave in the water layer and two longitudinal waves in the sediments are clearly seen. The shear wave in the sediments is not visible on the displayed image of pore pressure, but it can be seen in the plot of absolute velocities.

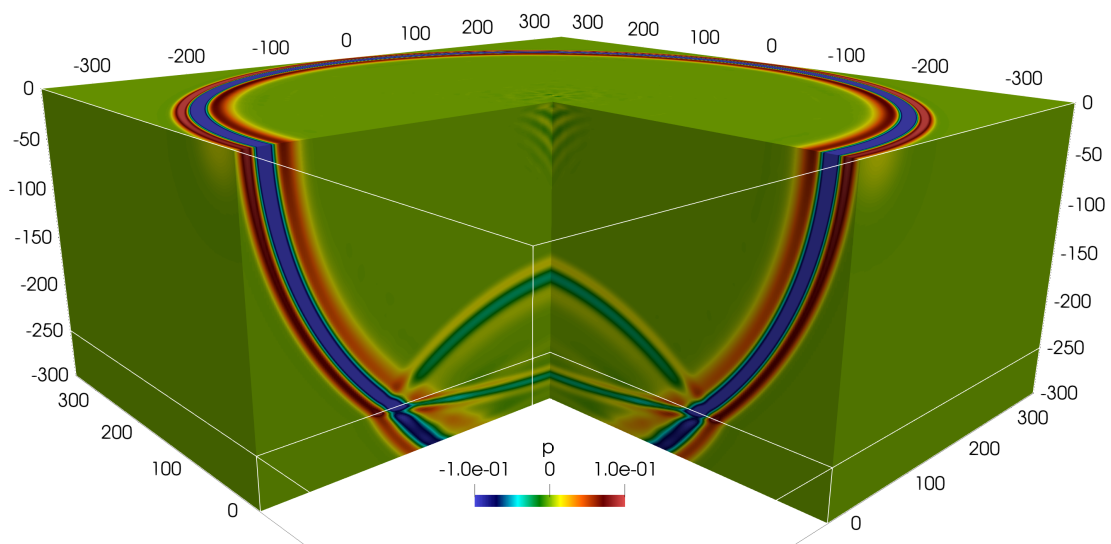


Figure 2. Spatial distribution of the pressure p in the model

Conclusion

The grid-characteristic numerical scheme on rectangular grids for the seismic wave simulation in a porous fluid-saturated medium governed by the Dorovsky model in the 3D case was

constructed. The analytical derivation of matrices Ω and Ω^{-1} makes an efficient implementation possible. The explicit contact conditions between an acoustic medium and a Dorovsky medium have been derived and implemented. We have simulated a problem in a realistic setting of a marine seismic survey process. The usage of the modern MPI and OpenMP technologies is the key for the real-scale computer simulation. Future works involves implementing contact between elastic and porous media and generalization to curvilinear grids in 3D, similar to the work [8] in the 2D case.

Acknowledgements

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 of Kurchatov Institute National Research Centre, <http://ckp.nrcki.ru/>.

The study was supported by the Russian Science Foundation project No. 24-49-02002, <https://rscf.ru/project/24-49-02002/>.






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. Aki, K., Richards, P.: Quantitative Seismology, Theory and Methods. University Science Books, second edn. (2002)
2. Auzinger, W., Koch, O., Thalhammer, M.: Defect-based local error estimators for high-order splitting methods involving three linear operators. Numerical Algorithms 70, 61–91 (2015). <https://doi.org/10.1007/s11075-014-9935-8>
3. Biot, M.: General theory of three dimensional consolidation. J. Appl. Phys. 12, 155–169 (1941). <https://doi.org/10.1007/s11075-014-9935-8>
4. Blokhin, A., Dorovsky, V.: Mathematical modelling in the theory of multivelocitity continuum. Nova Science Publishers (1995)
5. Chelnokov, F.: Numerical modelling of deformational dynamic processes in media with complex structure. Ph.D. thesis, Moscow Institute of Physics and Technology, Moscow, Russia (2005)
6. Favorskaya, A., 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>
7. Golubev, V., Shevchenko, A., Petrov, I.: Taking into account fluid saturation of bottom sediments in marine seismic survey. In: Doklady Mathematics. vol. 100, pp. 488–490. Springer (2019). <https://doi.org/10.1134/S1064562419050107>

8. Golubev, V., Shevchenko, A., Petrov, I.: Simulation of seismic wave propagation in a multicomponent oil deposit model. *International Journal of Applied Mechanics* 12(08), 2050084 (2020). <https://doi.org/10.1142/S1758825120500842>
9. Golubev, V., Shevchenko, A., Petrov, I.: Raising convergence order of grid-characteristic schemes for 2D linear elasticity problems using operator splitting. *Computer Research and Modeling* 14(4), 899–910 (2022). <https://doi.org/10.20537/2076-7633-2022-14-4-899-910>
10. LeVeque, R.: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics, Cambridge University Press (2002). <https://doi.org/10.1017/CB09780511791253>
11. Sorokin, K., Imomnazarov, K.: Numerical solving of the liner two-dimensional dynamic problem in liquid-filled porous media. *Journal of Siberian Federal University. Mathematics & Physics* 3(2), 256–261 (2010). <https://doi.org/10.1016/j.jcp.2012.11.001>
12. Wolfram Research, I.: *Mathematica, Version 14.1*, <https://www.wolfram.com/mathematica>, Champaign, IL, 2024

Numerical Modeling of Complex Geometry Thin Composite Structures under Vibrational Testing

Stepan A. Lavrenkov¹ , Ivan E. Smirnov¹ ,
Dmitrii A. Kravchenko^{1,2} , Katerina A. Beklemysheva¹ ,
Alexey V. Vasyukov¹ 

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

This paper considers the inverse problem of material elastic properties identification from vibrational testing data. The present work aims to describe the approach that uses different kinds of optimizations to allow fast inverse problem solution using single modern multicore CPU or GPU. This includes choosing the model that allows to minimize the computational cost still reproducing the experimental results with good quality. The model for mid-surface symmetric isotropic and composite plates that are moving in vibrational stand is provided. The inverse problem is formulated in terms of the loss function minimization and the solution is computed with stochastic global optimization algorithm and second-order local optimization algorithm, which uses automatic differentiation of the forward problem solver to compute the derivatives. The paper describes parallelization for CPU and GPU and also the approach to reduce RAM usage to fit into single server RAM or single GPU VRAM. The numerical experiments presented in the paper demonstrate the solutions for complex rheologies and geometries: laminated composite plates, isotropic materials with frequency dependent elastic properties, perforated samples.

Keywords: inverse problem, vibrational testing, automatic differentiation, openmp, jar.

Introduction

This paper considers the inverse problem of material elastic properties identification from vibrational testing data. Inverse problems are typically very computationally intensive. They are commonly solved using large scale clusters to achieve high quality of the solution. However, for the problem we consider, it is desirable to fit into simple hardware setup. If the problem can be solved using a single computer, the solver can be taken close to vibrational testing stands and integrated into engineering procedures. The present work aims to describe the approach that uses different kinds of optimizations to allow fast inverse problem solution using modern multicore CPU or GPU.

Composite materials are widely used in modern engineering projects, sometimes new materials are tailor-made for a certain problem. The experimental status of such materials may lead to lack of reliable data on their elastic properties, that are crucial for engineering estimations and numerical simulations. Moreover, even if the elastic properties of the material are considered to be known, an exact specimen may have different kinds of flaws caused by issues during the material production and further handling. The present work uses the data from the vibrational testing of the specimen to identify its elastic properties. The results of this work will contribute to the methods of nondestructive testing. The choice of the input data is based on the fact that vibrational testing is a routine procedure in many areas. So, if the vibrational data is available, it is desirable to get an additional information from this data. It is worth noting that the target problem of this paper should not be confused with nondestructive health monitoring methods based on known vibration signatures.

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russia

²Keldysh Research Center, Moscow, Russia

Similar problems were considered by different groups of authors. The paper [9] presented an approach that is very close to the one we used, but the authors simulated the response of a beam specimen, and we consider plates. Completely different approaches exist also, the one to mention is Dynamic Mechanical Analysis [2], but it requires small samples of the material of certain shapes, and it is not an option for our use case.

The approach used in this work is based on the results presented earlier by several authors. The work [24] used the experimental stand similar with the one referred to in the present work, the authors measured a mechanical impedance of a triangular aluminum plate in an atmosphere and in a vacuum chamber to study an energy dissipation processes. In [23] the different models for damping in composite materials are overviewed. Several works showed that different materials may have significantly different frequency dependencies of elastic properties [10, 15–17]. The usage of frequency-dependent dynamic moduli in [21] provided better explanation of a dynamic behaviour of a specimen made of two steel plates connected by a polymer layer. In [4] a framework is suggested, which allows to introduce into the model the phenomena of viscous, thermoelastic and viscoelastic damping and the damping due to sound radiation. The exact approach we use was first introduced in [1], the present work extends it and adapts to more complex rheologies and geometries.

The paper is structured as follows. Section 1 covers the choice of the model that allowed to minimize the computational cost still reproducing the experimental results with good quality. Section 2 describes the algorithmic approaches taken that allowed for effective parallelization both on CPU and GPU. The algorithmic part also covers the approach to reduce RAM usage to fit into single server RAM or single GPU VRAM. Section 3 presents the computational results to demonstrate the quality of the solution for complex rheologies and geometries.

1. Mathematical Model

In present work we consider the boundary value problem for the equation of motion of thin plate in the given experimental setup. One edge of the plate is clamped in the vibration stand, which generates sinusoidal oscillations with adjustable frequency, another edge is free. Both edges of the plate have the accelerometers installed on them, which allows for experimental acquisition of frequency-response function of plate oscillations. The forward problem is formulated as follows: *obtain frequency response for the plate with known properties*. Consequently, the inverse problem is *to compute the unknown elastic moduli of the material from experimentally obtained frequency response and geometrical properties of the plate*.

1.1. Forward Problem

The equations of motion for the plate was derived using the following assumptions:

- the plate is thin, so its height is much less than its length and width;
- the Kirchhoff-Love kinematic hypothesis is applicable: straight lines normal to the mid-surface remain straight and normal to the mid-surface after deformation, the thickness of the plate does not change during a deformation;
- the plate is symmetric with respect to the mid-surface and has the same height for every in-plane point;
- if the plate consists of several laminae, each lamina has constant thickness.

The resulting equations of motion are [19]:

$$2e\rho\ddot{w}(x, y, t) - \frac{2e^3}{3}\rho\Delta\ddot{w}(x, y, t) + \nabla \cdot \nabla \cdot \mathbf{M}(x, y, t) = 0, \quad (x, y) \in \Omega,$$

where w is the transverse movement of the plate, $e = \frac{1}{2}h$ is half of the plate thickness, Ω is a non-empty connected open subset of \mathbb{R}^2 , which represents mid-surface of a plate, ρ is the density of the material, \mathbf{M} is the tensor of moments:

$$\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix} = \int_{-e}^e \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix} z^2 dz.$$

For a linear elastic material:

$$\begin{pmatrix} M_{xx} \\ M_{yy} \\ M_{xy} \end{pmatrix} = \begin{pmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{pmatrix} \begin{pmatrix} \frac{\partial^2}{\partial x^2} w \\ \frac{\partial^2}{\partial y^2} w \\ \frac{\partial^2}{\partial x \partial y} w \end{pmatrix} = D \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix} \begin{pmatrix} \frac{\partial^2}{\partial x^2} w \\ \frac{\partial^2}{\partial y^2} w \\ \frac{\partial^2}{\partial x \partial y} w \end{pmatrix},$$

where $D = \frac{Eh^3}{12(1-\nu^2)}$ is flexural rigidity, $\nu = (\frac{E}{2G} - 1)$ is Poisson's ratio, E is Young's modulus, G is shear modulus.

In case of laminated plates the classical laminate theory (CLT) [8, 19] is used to determine D_{ij} components. In present work, orthotropic laminae are considered. Consider the k -th orthotropic lamina with axes $(x_1, y_1)_k$ which are associated with the transverse and normal directions to the reinforcing fibers. These axes are rotated by angle φ_k counterclockwise about the plate axes (x, y) . In $(x_1, y_1)_k$ axes Hooke's law is formulated as:

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_6 \end{pmatrix}_k = \begin{pmatrix} Q_{11} & Q_{12} & Q_{16} \\ Q_{12} & Q_{22} & Q_{26} \\ Q_{16} & Q_{26} & Q_{66} \end{pmatrix}_k \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_6 \end{pmatrix}_k, \quad (1)$$

where Q_{ij} are components of tensor \mathbf{Q}_k which depend on engineering constants $E_1, E_2, G_{12}, \nu_{12}$:

$$\nu_{21} = \frac{E_2\nu_{12}}{E_1}, \quad Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}, \quad Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}}, \quad Q_{12} = \nu_{12}Q_{22},$$

$$Q_{66} = G_{12}, \quad Q_{16} = Q_{26} = 0.$$

The equation (1) in (x, y) axes:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}_k = \begin{pmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{pmatrix}_k \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{pmatrix}_k. \quad (2)$$

Here $\overline{\mathbf{Q}}_k = \mathbf{T}(\varphi_k) \cdot \mathbf{Q}_k \cdot \mathbf{T}^T(\varphi_k)$, rotational tensor \mathbf{T} in (x, y) axes:

$$\mathbf{T}(\varphi_k) = \begin{pmatrix} \cos^2 \varphi_k & \sin^2 \varphi_k & -2 \sin \varphi_k \cos \varphi_k \\ \sin^2 \varphi_k & \cos^2 \varphi_k & 2 \sin \varphi_k \cos \varphi_k \\ \sin \varphi_k \cos \varphi_k & -\sin \varphi_k \cos \varphi_k & \cos^2 \varphi_k - \sin^2 \varphi_k \end{pmatrix}.$$

The D_{ij} components for the laminate with N laminae are then calculated as

$$\mathbf{D} = \begin{pmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{pmatrix} = \sum_{k=1}^N \int_{z_{k-1}}^{z_k} \begin{pmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{12} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{16} & \overline{Q}_{26} & \overline{Q}_{66} \end{pmatrix}_k z^2 dz. \quad (3)$$

In the equation above, z corresponds to a coordinate along axis, which is normal to the mid-surface. Also, the following rule is applied: $z_0 = -e$, $z_N = e$, $z_k - z_{k-1} = h_k$ - thickness of k -th lamina. Each lamina is orthotropic and has constant thickness across the plate so the general formula is:

$$D_{ij} = \frac{1}{3} \sum_{k=1}^N (\overline{Q}_{ij})_k (z_k^3 - z_{k-1}^3).$$

Considering the oscillations as harmonic $w = u(x, y, \omega)e^{i\omega t}$ and introducing the damping via complex dynamic moduli [12] with loss factor β such as $\mathcal{D}_{ij} = D_{ij}(1 + i\beta)$, we arrive at the equation of motion in frequency domain (4) with boundary conditions (5) on the clamped end, which oscillates with amplitude g_0 , and (6) on the free end [20].

$$-2e\rho\omega^2 \left(u - \frac{e^2}{3} \Delta u \right) + \nabla \cdot \nabla \cdot (\mathcal{D} \cdot \nabla \otimes \nabla u) = 0, \quad (4)$$

$$\begin{cases} u = g_0 \\ \nabla u \cdot \mathbf{n} = 0 \end{cases} \quad (x, y) \in \Gamma_c, \quad (5)$$

$$\begin{cases} -\frac{2}{3}e^3\rho\omega^2 \nabla u \cdot \mathbf{n} + \nabla \cdot \mathbf{M} \cdot \mathbf{n} + \nabla(\mathbf{M} \cdot \mathbf{n} \cdot \boldsymbol{\tau}) \cdot \boldsymbol{\tau} = 0 \\ \mathbf{M} \cdot \mathbf{n} \cdot \boldsymbol{\tau} = 0 \end{cases} \quad (x, y) \in \Gamma_f. \quad (6)$$

1.2. Inverse Problem

Let θ be the vector of unknown model parameters (for an isotropic material $\theta = (E, G, \beta)^\top$, for an orthotropic one $\theta = (E_1, E_2, G_{12}, \nu_{12}, \beta)^\top$, and $\{(\omega_k, u_k^{exp})\}_{k=1}^N$ - discrete frequency response acquired in the experiment. Then, solving the inverse problem is equal to finding the minimum of a loss function (7)

$$L(\theta_s) = \min_{\theta \in \Theta} L(\theta) = \min_{\theta \in \Theta} \left[\frac{1}{N_{exp}} \sum_{m=1}^{N_{exp}} (\log(|u(\omega_m, \theta)|) - \log(|u_{exp}(\omega_m)|)) \right], \quad (7)$$

where $u(\theta, \omega_k)$ is a solution of a forward problem with parameters θ and frequency ω_k , Θ is the admissible set of parameters, θ_s is a solution vector.

2. Numerical Method and Algorithms

2.1. Forward Problem

The system of equations (4)–(6) is solved numerically using FEM with Morley elements [13]. The matrices obtained after spatial discretization are not dependent on frequency ω and tensor \mathcal{D} . This makes the following optimization possible: matrices are calculated once and stored in memory, so the global matrix of linear system is obtained via linear combination with coeffi-

cients, which depend on ω and values in tensor \mathcal{D} . So, forward problems for different vectors of unknowns θ can be solved independently using parallel computations.

The algorithm described above was implemented in Python using FreeFem++ [7] for FEM matrices operations and jax [3] for automatic differentiation of forward problem solver and JIT compilation for CPU and GPU targets. In present, work two implementations were used. The first one is GPU-accelerated solver which is based on dense matrix operations provided by jax. The second solver uses CPU to operate on sparse matrices. After FEM discretization the sparsity pattern of the resulting linear system is determined so all constant matrices are stored as 1D arrays. This solver uses UMFPACK [6] to solve the resulting sparse linear system and OpenMP [5] to implement parallel computations for linear systems with different frequencies ω .

The computation of forward problem with CPU solver takes around 1.5 seconds on a single core of Intel i7-11370H CPU for a rectangular isotropic plate with reasonably dense spatial mesh and 1000 frequencies. The computation requires around 1.5 GB RAM. The efficiency of the parallel version is about 70%. This allows to achieve forward problem computation time less than 0.5 seconds when using single multi core CPU. These results allow to solve the inverse problem on a single machine as well.

2.2. Inverse Problem

The minimization of loss function (7) is implemented with global and local optimization algorithms. Global optimization uses differential evolution algorithm [22] to find the approximate minimum when the bounds on each parameter are given. Then, local optimization based on trust-region method [14] is used to improve the solution from global optimization. The local optimization uses gradients and Hessian matrices calculated via automatic differentiation. Also, the data compression algorithm [11] is used to decrease the amount of points in discrete frequency range which results in lower RAM consumption.

3. Results

3.1. Composite Laminates

Three numerical experiments were conducted using the algorithm for laminates described above. Plates with lamina from carbon-fiber reinforced polymers (CFRP) were chosen, the reference parameters were taken from [18]. The modeling covered the plate under investigation and also the accelerometer with mass 1.7 g and radius 3.8 mm. The results of numerical simulation may provide a valuable insight on how the experimental procedure could be improved.

3.1.1. Laminate with plies of woven fabric

In this case, the material has reinforcing fibers in two orthogonal directions with plain weave. This symmetry allows for the following simplification: two engineering constants are equal $E_1 = E_2$ and the number of unknown elastic moduli is reduced by one. The parameters of the plate are listed in Tab. 1.

Using these parameters the forward problem was solved in frequency range from 0 to 2500 Hz. The result of this calculation was considered to be the reference data. Then, the parameters of the specimen were considered unknown, and the inverse problem was solved in respect to the reference data. The results are presented in Fig. 1 and in Tab. 2. These results show that the

inverse problem can be solved with high accuracy for such parameter configuration even for really large bounds for global optimization.

Table 1. Woven fabric sample parameters

Parameter	Value
Dimensions, [mm]	$100 \times 20 \times 1.5$
Ply orientation, [degrees]	(0, 30, 0)
Density ρ , [$\text{kg} \cdot \text{m}^{-3}$]	1400
E_1, E_2 , [GPa]	61.0
G_{12} , [GPa]	4.9
ν_{12} , [1]	0.04
Loss modulus β , [1]	0.02

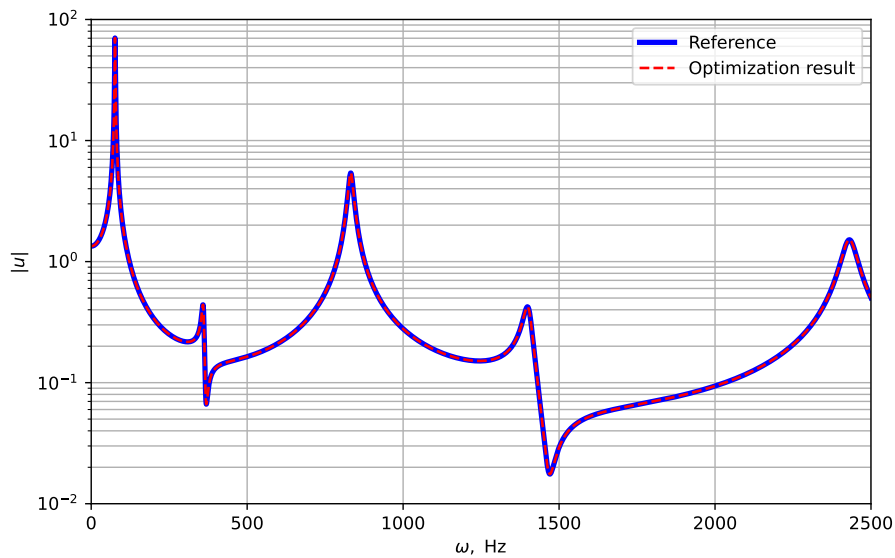


Figure 1. Woven fabric sample: frequency responses for reference parameter values and values from the inverse problem solution

Table 2. Results for plate with woven plies

Parameter	E_1 , GPa	G_{12} , GPa	ν_{12}
Reference value	61.0	4.9	0.04
Parameter bounds for global optimization	3–130	1–40	0.01–1.0
Optimization result	60.99	4.899	0.0404
Relative error, %	$-2.2 \cdot 10^{-4}$	-0.014	0.9

3.1.2. Thin anisotropic plate

In this experiment, the material consists of thin unidirectional plies with the parameters presented in Tab. 3. The forward problem was solved in frequency range from 0 to 2700 Hz. The inverse problem was solved for two different frequency ranges: 0–1700 Hz and 0–2700 Hz. The

results are presented in Fig. 2 and in Tab. 4. The results show that the quality of solution may benefit from narrowing the frequency range in case when there are many peaks present in AFC.

Table 3. Thin sample parameters

Parameter	Value
Dimensions, [mm]	$100 \times 20 \times 0.45$
Ply orientation, [degrees]	(0, 35, 0)
Density ρ , [$\text{kg} \cdot \text{m}^{-3}$]	1100
E_1 , [GPa]	67.4
E_2 , [GPa]	17.7
G_{12} , [GPa]	11.34
ν_{12} , [1]	0.47
Loss modulus β , [1]	0.02

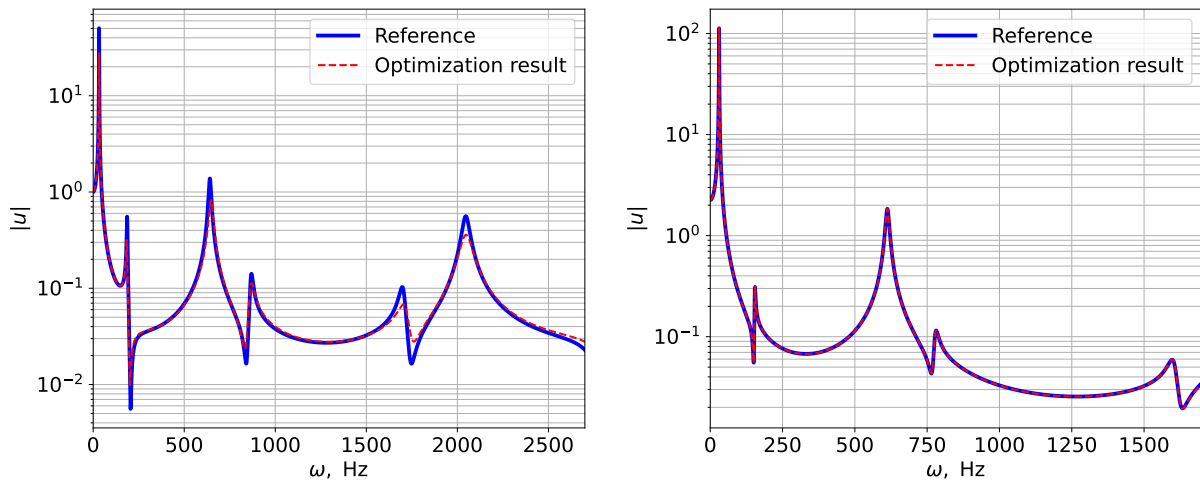


Figure 2. Thin plate: results for wide (left) and narrow (right) frequency ranges

Table 4. Results for thin plate with unidirectional plies in wide/narrow frequency range

Parameter	E_1 , GPa	E_2 , GPa	G_{12} , GPa	ν_{12}
Reference value	67.4	17.7	11.3	0.47
Parameter bounds for global optimization	30–200	1–50	1–50	0.01–2.0
Optimization result	72.2/68.6	17.8/18.1	11.4/11.3	0.45/0.471
Relative error, %	7.2/1.8	0.7/2.4	0.3/0.007	–3.5/0.2

3.1.3. Thick anisotropic plate

In this experiment, the laminate consists of 11 unidirectional plies with the parameters presented in Tab. 5. The forward problem was solved in frequency range from 0 to 2500 Hz. The results are presented in Fig. 3 and in Tab. 6. The relative error of inverse problem solution is higher in comparison to thin plate case, which means that accuracy may decrease for plates with many laminae.

Table 5. Parameters for thick plate

Parameter	Value
Dimensions, [mm]	$100 \times 20 \times 1.4$
Ply orientation, [degrees]	(0, 35, 45, 90, 45, 0, 45, 90, 45, 35, 0)
Density ρ , [$\text{kg} \cdot \text{m}^{-3}$]	1400
E_1 , [GPa]	62
E_2 , [GPa]	45
G_{12} , [GPa]	21
ν_{12} , [1]	0.42
Loss modulus β , [1]	0.02

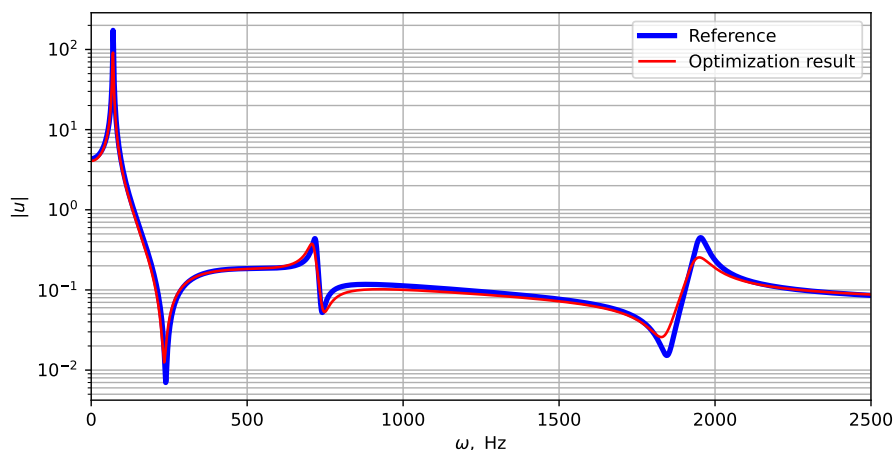

Figure 3. Thick plate: inverse problem solution and reference AFC

Table 6. Results for thick plate

Parameter	E_1 , GPa	E_2 , GPa	G_{12} , GPa	ν_{12}
Reference value	62.0	45.0	21.0	0.42
Bounds for optimization	3–150	10–70	1–50	0.01–2.0
Optimization result	63.33	47.55	21.3	0.352
Relative error, %	2.2	5.7	1.4	–16.2

3.2. Isotropic Samples

Several numerical experiments were performed for isotropic metallic samples, since real experimental data was available for such specimens. These experiments aim to demonstrate that the presented approach can work for real life samples taking into account relatively complex geometry and rheology.

3.2.1. Frequency dependence

As it was mentioned in Introduction, there are evidences that different materials possess significantly different frequency dependence of elastic moduli. There is no common theory in this area that covers all the materials and frequency ranges. Different authors typically use empirical formulae derived from experimental data. The present work used two most common approaches. The first one is linear increase or decrease of moduli as given by (8). The results

that can be achieved with this approximation are presented in Fig. 4. The change of moduli on 2000 Hz were 30 % of the initial values.

$$\begin{aligned} E &= E_0 (1 \pm \alpha\omega), \\ \nu &= \nu_0 (1 \pm \beta\omega). \end{aligned} \tag{8}$$

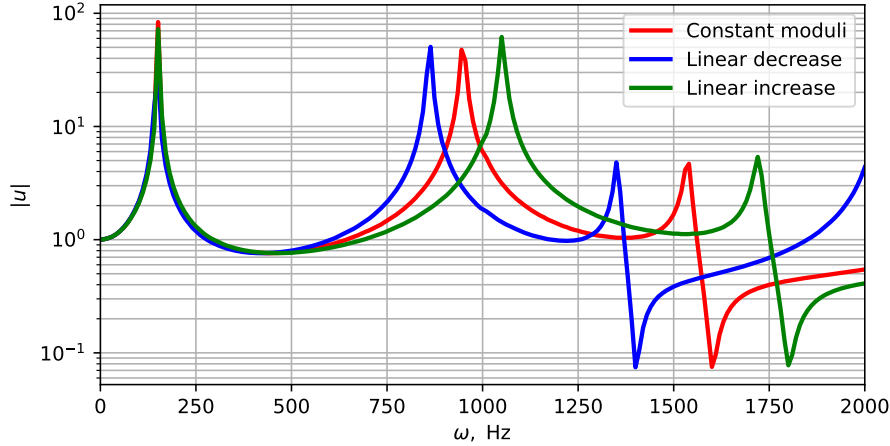


Figure 4. Synthetic calculations with frequency dependent elastic moduli

The second commonly used approximation is expressed as (9). When using the second approximation, $\{\alpha_E, \beta_E, \alpha_\nu, \beta_\nu\}$ become additional sought coefficients of inverse problem. This makes the inverse problem harder but allows to capture more complex material behaviour.

$$\begin{aligned} E &= E_0 \left(1 + \alpha_E e^{\beta_E \omega}\right), \\ \nu &= \nu_0 \left(1 + \alpha_\nu e^{\beta_\nu \omega}\right). \end{aligned} \tag{9}$$

To test this approximation, the inverse problem was solved twice using the real experimental data for a steel plate. The first solution was obtained without taking frequency dependence into account. The second solution was calculated using the approximation (9). The frequency response results are presented in Fig. 5. The final parameters for the solution without frequency dependence are: $\{E = 192.47 \text{ GPa}, \nu = 0.23, \beta = 0.0091\}$. The results for the solution with frequency dependence are presented in Tab. 7. One can see that the frequency dependence for the specimen under testing can be considered neglectable in the frequency range used in the experiment. However, this experiment demonstrated that the solution of the inverse problem remains stable when we introduce frequency dependence of moduli.

Table 7. Results for frequency dependent elastic moduli

Parameter	α_E	β_E	α_ν	β_ν	E_0	ν_0	β
Value	$-2.57 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$	$-4.0 \cdot 10^{-4}$	10^{-3}	192.47	0.23	0.0091

3.2.2. Perforated plates

Another numerical experiment performed was about testing the possibility to represent highly perforated specimens using effective parameters. The motivation behind this experiment is the fact that calculations of perforated plates with real geometry require a large amount

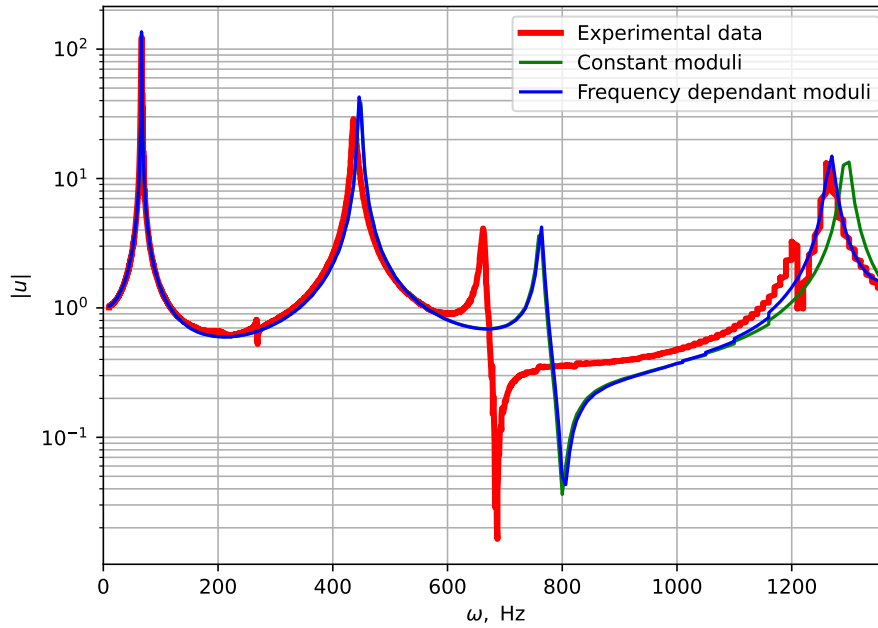


Figure 5. Inverse problem solution with frequency dependence of elastic moduli

of time and memory to represent the holes in details. An example of such perforated plate with generated mesh on it is displayed in Fig. 6. A possible way to solve this problem is to determine effective parameters of solid plate that has the same frequency response function as the perforated one.

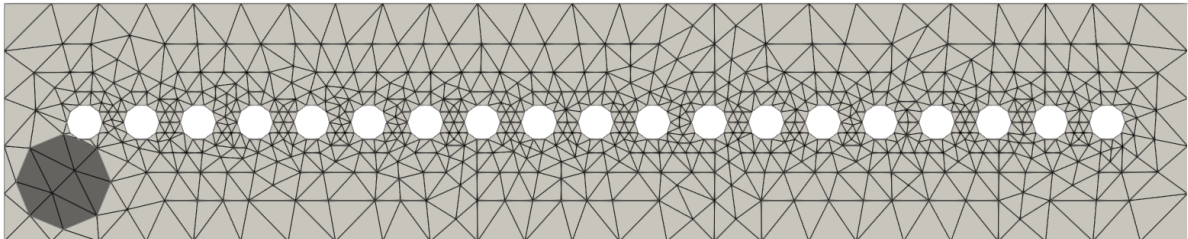


Figure 6. An example of perforated plate with mesh on it

During this experiment, the plate was taken perforated by one central line of holes along length, total linear size of holes was kept constant but their radius was varied. An effective density of a solid plate was calculated as $\rho = \rho_0 (1 - \pi r^2 N / xy)$, where ρ_0 is real material density, r – radius of each hole, N – number of holes, xy – square of plate. An inverse problem was solved for each value of radius to determine effective elastic moduli. The results are presented in Fig. 7, the values are presented relative to the moduli on the solid plate without perforation.

This approach was tested for the inverse problem solution. The results are presented in Fig. 8. Red line represents real experimental vibrational data for 2 mm thick aluminum plate with 19 holes of 1.5 mm radius. Green line is the result of solving the direct problem for a solid plate with an effective density and elastic moduli calculated from the dependence presented in Fig. 7. Blue line is the result of solving the inverse problem for a solid plate using these effective density and elastic moduli as an initial guess.

One can see that the effective parameters estimated from these simple relations do not deliver good quality solution. It is an expected result since the relations in Fig. 7 were derived

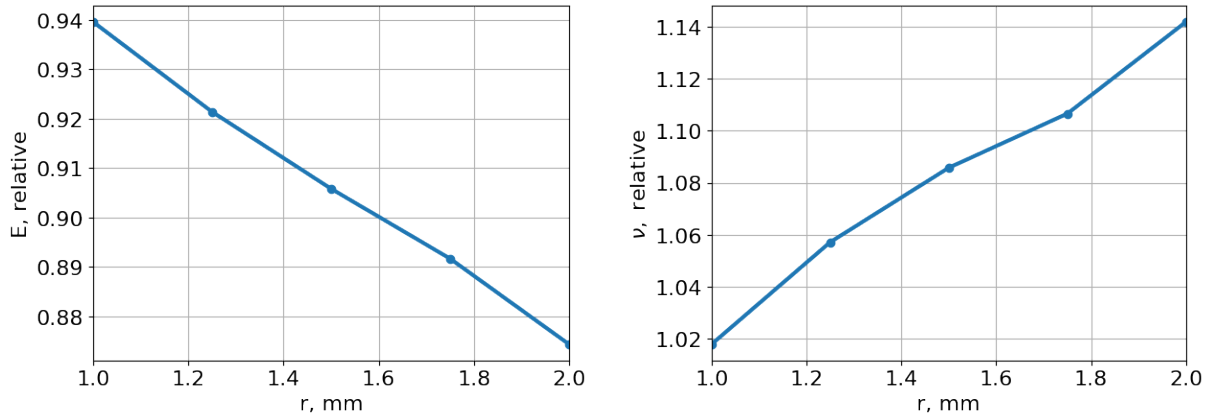


Figure 7. Effective elastic moduli of perforated plate with respect to holes radius: Young's module (left) and Poisson coefficient (right)

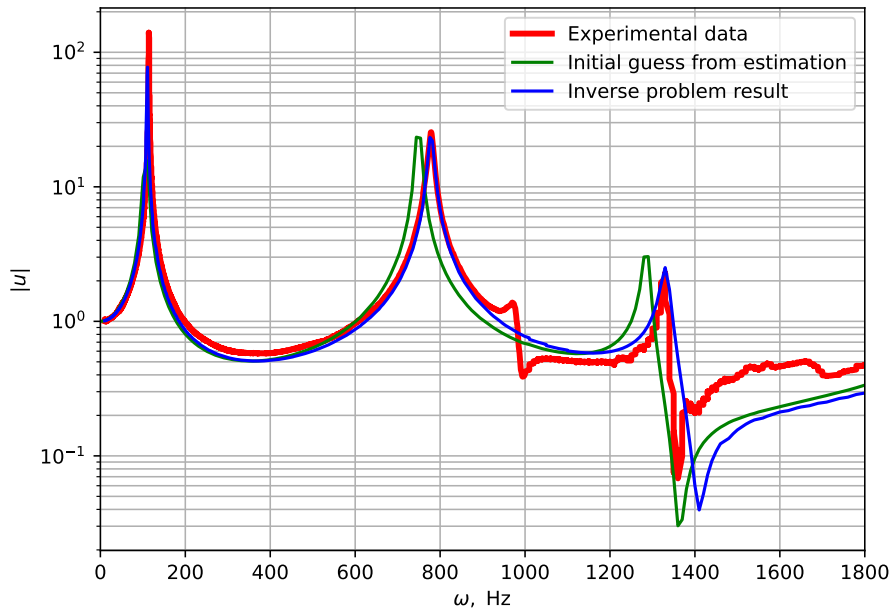


Figure 8. Using perforated plate effective parameters for inverse problem solution

from a sample of certain geometry and rheology, one can not expect that all the specimens from other materials will behave exactly the same. However, these effective parameters can be used as a good initial approximation that reduces the time to final solution.

Conclusion

The present work considers the inverse problem of material elastic properties identification from vibrational testing data. The major focus of the paper is to describe the approach that uses different kinds of optimizations to allow fast inverse problem solution. The methods presented allow to achieve forward problem computation time less than 0.5 seconds using single modern multicore CPU or GPU. This fact makes it possible to solve the inverse problem on a single machine as well. The numerical experiments presented in the paper demonstrate the applicability of the approach for complex rheologies and geometries: multilayered laminated composite plates, isotropic materials with frequency dependent elastic properties, perforated samples.

Several directions, outlined in the present paper, require further research. The inverse problem for thick laminates was solved, but the quality of the solution may be improved. The concept of frequency dependency of elastic moduli should be studied further to become applicable for a larger scope of materials. The approach for perforated specimens effective properties should be refined in future work using the experimental data for different materials.

Acknowledgements

The study was supported by the Russian Science Foundation project № 22-11-00142.

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. Aksenov, V., Vasyukov, A., Beklemysheva, K.: Acquiring elastic properties of thin composite structure from vibrational testing data. *Journal of Inverse and Ill-posed Problems* 32(3), 467–484 (2024). <https://doi.org/10.1515/jiip-2022-0081>
2. Ayyagari, S., Al-Haik, M.: Enhancing the viscoelastic performance of carbon fiber composites by incorporating CNTs and ZnO nanofillers. *Applied Sciences* 9(11), 2281 (2019). <https://doi.org/10.3390/app9112281>
3. Bradbury, J., Frostig, R., Hawkins, P., *et al.*: JAX: composable transformations of Python+NumPy programs. <http://github.com/google/jax> (2018), accessed: 2024-05-30
4. Chaigne, A., Lambourg, C.: Time-domain simulation of damped impacted plates. I. Theory and experiments. *The Journal of the Acoustical Society of America* 109(4), 1422–1432 (2001). <https://doi.org/10.1121/1.1354200>
5. Dagum, L., Menon, R.: OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering* 5(1), 46–55 (1998). <https://doi.org/10.1109/99.660313>
6. Davis, T.A.: Algorithm 832: UMFPACK V4.3 – an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.* 30(2), 196–199 (2004). <https://doi.org/10.1145/992200.992206>
7. Hecht, F.: New development in FreeFem++. *J. Numer. Math.* 20(3-4), 251–266 (2012). <https://doi.org/10.1515/jnum-2012-0013>
8. Jones, R.M.: *Mechanics of composite materials*. Taylor & Francis, Philadelphia, PA (1999). <https://doi.org/10.1201/9781498711067>
9. Kim, S.Y., Lee, D.H.: Identification of fractional-derivative-model parameters of viscoelastic materials from measured FRFs. *Journal of sound and vibration* 324(3-5), 570–586 (2009). <https://doi.org/10.1016/j.jsv.2009.02.040>

10. Koblar, D., Boltežar, M.: Evaluation of the frequency-dependent Youngs modulus and damping factor of rubber from experiment and their implementation in a finite-element analysis. *Experimental Techniques* 40, 235–244 (2016). <https://doi.org/10.1007/s40799-016-0027-7>
11. Lavrenkov, S., Vasyukov, A.: Experimental data compression for GPU-based solution of inverse coefficient problem for vibrational testing data. In: Balandin, D., Barkalov, K., Meyerov, I. (eds.) *Mathematical Modeling and Supercomputer Technologies, MMST 2023*, Nizhny Novgorod, Russia, November 13-14, 2023. Proceedings. Communications in Computer and Information Science, vol. 1914, pp. 302–309. Springer Nature Switzerland (2024). https://doi.org/10.1007/978-3-031-52470-7_24
12. Menard, K.P., Menard, N.: *Dynamic Mechanical Analysis*. CRC Press, Boca Raton, FL (2020). <https://doi.org/10.1201/9780429190308>
13. Morley, L.S.D.: The triangular equilibrium element in the solution of plate bending problems. *Aeronautical Quarterly* 19(2), 149–169 (1968). <https://doi.org/10.1017/S0001925900004546>
14. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer New York, New York, NY (2006). <https://doi.org/10.1007/978-0-387-40065-5>
15. Paimushin, V.N., Firsov, V.A., Gazizullin, R.K., Shishkin, V.M.: Theoretical and experimental method for determining the frequency-dependent dynamic modulus of elasticity and damping characteristics of a titanium alloy OT-4. *Journal of Physics: Conference Series* 1158(3), 032044 (2019). <https://doi.org/10.1088/1742-6596/1158/3/032044>
16. Paimushin, V.N., Firsov, V.A., Gyunal, I., Shishkin, V.M.: Accounting for the frequency-dependent dynamic elastic modulus of duralumin in deformation problems. *Journal of Applied Mechanics and Technical Physics* 58, 517–528 (2017). <https://doi.org/10.1134/S0021894417030178>
17. Pritz, T.: Frequency dependences of complex moduli and complex Poisson's ratio of real solid materials. *Journal of Sound and Vibration* 214(1), 83–104 (1998). <https://doi.org/10.1006/jsvi.1998.1534>
18. Rahmani, H., Najaf, S.H.M., Ashori, A., Golriz, M.: Elastic properties of carbon fibre-reinforced epoxy composites. *Polymers and Polymer Composites* 23(7), 475–482 (2015). <https://doi.org/10.1177/096739111502300706>
19. Reddy, J.N.: *Mechanics of Laminated Composite Plates and Shells*. CRC Press, Boca Raton, FL (2003). <https://doi.org/10.1201/b12409>
20. Reddy, J.N.: *Theory and Analysis of Elastic Plates and Shells*. CRC Press, Boca Raton, FL (2006). <https://doi.org/10.1201/9780849384165>
21. Ruzek, M., Guyader, J.L., Pezerat, C.: Experimental identification of the bending equation of beams from the vibration shape measurements. *Journal of Sound and Vibration* 332(16), 3623–3635 (2013). <https://doi.org/10.1016/j.jsv.2013.02.017>

22. Storn, R., Prince, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997). <https://doi.org/10.1023/A:1008202821328>
23. Treviso, A., Van Genechten, B., Mundo, D., Tournour, M.: Damping in composite materials: Properties and models. *Composites Part B: Engineering* 78, 144–152 (2015). <https://doi.org/10.1016/j.compositesb.2015.03.081>
24. Tuan, P.H., Wen, C.P., Chiang, P.Y., *et al.*: Exploring the resonant vibration of thin plates: reconstruction of Chladni patterns and determination of resonant wave numbers. *The Journal of the Acoustical Society of America* 137(4), 2113–2123 (2015). <https://doi.org/10.1121/1.4916704>

Numerical Modeling of Marine Seismology in the Arctic Region During Deposit Dissolution due to Oil Migration

*Evgeniya K. Guseva*¹ , *Vasily I. Golubev*^{1,2} , *Igor B. Petrov*^{1,2} 

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

As the Arctic region requires costly and difficult to access surveys of the ground, numerical modeling proves to be an effective way to study occurring processes in the area. Moreover, the simulations of the seismic exploration can help identify the main signs of oil migration which is crucial for the risk assessment of the deposit development. Therefore, the main goal of this work is to conduct the forward modeling of the seismology in the offshore areas of the region in order to determine the indicators of such processes. The present study, in particular, is aimed at recreating of basic features of the region such as a layered ground with the gradual change in material parameters and inclusion of a permafrost sheet. Furthermore, the boundaries between layers are considered to be curvilinear. This complex problem was effectively solved using the grid-characteristic method which allows for the accelerating of the calculations using OpenMP. As a result of the computations, the reconstructed wave phenomena is analyzed based on the obtained wave patterns and synthesized seismograms. The change in the responses from the oil reservoir in the process of draining is identified which can further help interpret real measurements.

Keywords: numerical modeling, oil migration, marine seismology, grid-characteristic method, Arctic region.

Introduction

Throughout the years, numerical modeling proved to be a great way to investigate natural and anthropogenous processes, especially in the places of costly and difficult to access in situ measurements such as the Arctic region [11, 15]. The area draws a lot of attention of the researchers due to the significant mineral reserves and promising oil development. This proves the necessity for conducting full-scale surveys of the subsurface areas of ground. For this purpose, measurements are commonly conducted using seismology [23]. Harsh environmental conditions incentive thorough investigation of the area before the preparations for the development of the deposit can begin. Thus, it is essential to identify potential dissolution of the oil-bearing formation which is still a poorly understood mechanism [1, 3]. As a result, numerical modeling can help determine the indicators of this process in the obtained measurements which is the purpose of this work.

However, this task is complicated by the series of unique qualities of the Arctic region which are mainly defined by cold temperatures. As a result, one of the main characteristics of the region is the presence of permafrost sheets, formed when the temperature constantly stays less than 0°C for a long period (from a couple to thousands of years) [20]. The layered ground is often inhomogeneous which causes the inconstancy of its mechanical characteristics and the curvilinear boundaries between the sheets. In order to account for the gradual change of the parameters, in this work, the obtained in situ data [14, 16, 26] is used to determine their dependencies on the depth. As a result, the computational domain was introduced consisting of the sea, water floor, permafrost sheet and bottom dense clay layer which contained the oil reservoir that decreased in size due to drain. This distinguishes the study from the previous works that use simpler settings with constant parameters [2, 8] and linear boundaries between the layers [9, 18].

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

²Scientific Research Institute for System Analysis of the National Research Centre “Kurchatov Institute”, Moscow, Russian Federation

The computer modeling was conducted using a software package RECT written in C++ and dedicated to the solution of hyperbolic equations by applying the grid-characteristic method [5] on structured grids. This method allows for effective solution of dynamic problems in objects of complex shapes [6] and with different rheologies [7]. Moreover, the software supports several means of parallelization, such as OpenMP [13], MPI [19], and CUDA [12] with the former one used to accelerate calculations in this work. The package is developed by the Computational Physics Department and the Informatics and Computational Mathematics Department of the Moscow Institute of Physics and Technology. The article is organized as follows. Section 1 is devoted to explanation of numerical setting including the governing system of equations in Section 1.1, computational method in Section 1.2 and domain in Section 1.3. Obtained results are discussed in Section 2, they consist of the wave patterns analyzed in Section 2.1 and seismograms in Section 2.2. Conclusion summarizes the study and points directions for further work.

1. Numerical Setting

1.1. Governing System of Equations

In order to represent all layers of the computational domain, isotropic linear elasticity is used. In the current work, the simulations are conducted in two-dimensional case. Three-dimensional calculations require a lot of time and computational resources while showing results similar to the plane problem [24]. The difference between these settings can be the result of the strong asymmetry which is not considered in this work. Thus, it is still possible to determine the influence of the deposit dissolution on the seismic measurements. The governing system of equations can be written in the following form:

$$\begin{aligned}
 \rho \frac{\partial v_x}{\partial t} &= \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + f_x, \\
 \rho \frac{\partial v_y}{\partial t} &= \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + f_y, \\
 \frac{\partial \sigma_{xx}}{\partial t} &= (\lambda + 2\mu) \frac{\partial v_x}{\partial x} + \lambda \frac{\partial v_y}{\partial y}, \\
 \frac{\partial \sigma_{yy}}{\partial t} &= \lambda \frac{\partial v_x}{\partial x} + (\lambda + 2\mu) \frac{\partial v_y}{\partial y}, \\
 \frac{\partial \sigma_{xy}}{\partial t} &= \mu \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right).
 \end{aligned} \tag{1}$$

Here the velocity vector of medium points $\mathbf{v} = (v_x, v_y)^\top$ and the stress tensor σ are unknown and calculated after each time step. The material parameters are density ρ and Lamé parameters λ and μ , the equations also allow for introduction of external forces $\mathbf{f} = (f_x, f_y)^\top$. This model reconstructs pressure and shear waves with velocities $c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $c_s = \sqrt{\frac{\mu}{\rho}}$. These velocities and density fully define these equations.

1.2. Grid-Characteristic Method

For the solution of the hyperbolic problem (1), the grid-characteristic method [5] is used. According to the procedure, the initial system is rewritten with introduction of the new vector of unknowns $\mathbf{u} = (v_x, v_y, \sigma_{xx}, \sigma_{yy}, \sigma_{xy})^\top$:

$$\frac{\partial \mathbf{u}}{\partial t} + A_x \frac{\partial \mathbf{u}}{\partial x} + A_y \frac{\partial \mathbf{u}}{\partial y} = \mathbf{g}. \tag{2}$$

Here matrices A_i are introduced:

$$A_x = - \begin{pmatrix} 0 & 0 & 1/\rho & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\rho \\ \lambda + 2\mu & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \end{pmatrix}, A_y = - \begin{pmatrix} 0 & 0 & 0 & 0 & 1/\rho \\ 0 & 0 & 0 & 1/\rho & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & \lambda + 2\mu & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3)$$

Next, coordinate-wise splitting and splitting on physical processes are performed. At first, the step along the horizontal axis x is calculated, next, the step along the vertical axis y was made. Finally, external forces were taken into account. For the solution of each of these equations, the new variables, Riemann invariants $\mathbf{r}_i = \{r_i^j\}_{j=1}^5$, were introduced. As the system (2) is hyperbolic, matrices A_i , $i = \{x, y\}$ can be presented in the form $A_i = R_i \Lambda_i L_i$, where L_i is the eigenvector matrix, $R_i = L_i^{-1}$, and $\Lambda_i = \text{diag}\{0, -c_s, c_s, -c_p, c_p\}$ is the eigenvalue matrix. Thus, (2) can be rewritten as the system of one-dimensional transport equations with $\mathbf{r}_i = L_i \mathbf{u}$:

$$\frac{\partial \mathbf{r}_i}{\partial t} + \Lambda_i \frac{\partial \mathbf{r}_i}{\partial i} = 0. \quad (4)$$

For the solution of each equation in (4), third order Rusanov scheme was applied [21], monotonized using the grid-characteristic monotonicity criterion [10]:

$$[r_i^j]_m^{n+1} = [r_i^j]_m^n + \frac{c_i^j}{2} ([r_i^j]_{m-1}^n - [r_i^j]_{m+1}^n) + \frac{c_i^{j2}}{2} ([r_i^j]_{m-1}^n - 2[r_i^j]_m^n + [r_i^j]_{m+1}^n) - \frac{c_i^j(1 - c_i^{j2})}{6} ([r_i^j]_{m-2}^n - 3[r_i^j]_{m-1}^n + 3[r_i^j]_m^n - [r_i^j]_{m+1}^n).$$

Here the Courant number is introduced $c_i^j = \frac{\lambda_i^j \tau}{h_i}$, and $j \in [1, 5]$, λ_i^j is the corresponding eigenvalue, h_i is the spacial step. Condition $c_i^j < 1$ guaranteed the stability of the calculations. One of the numerical problems which is common of the dynamic processes in the layered media is the appearance of non-physical oscillations due to high gradients in the mechanical properties. To deal with this issue, we introduce the grid-characteristic criterion in order to monotone the solution. When $[r_i^j]_m^{n+1} \leq \min [r_i^j]_1^n, [r_i^j]_2^n$, the following correction was performed: $[r_i^j]_m^{n+1} = \min [w_i^j]_1^n, [r_i^j]_2^n$. In case $[r_i^j]_m^{n+1} \geq \max [r_i^j]_1^n, [r_i^j]_2^n$, another formulae was used: $[r_i^j]_m^{n+1} = \max \{[r_i^j]_1^n, [r_i^j]_2^n\}$. Here $[r_i^j]_1^n, [r_i^j]_2^n$ are the grid function values at the time $t = t_n$ in the nearest points to the characteristic which originates from the point (t_n, x_m) . The introduced scheme is explicit which made it possible to employ parallelism using the OpenMP technology. This allowed for the significant acceleration of the calculations.

1.3. Computational Domain

In order to conduct simulations, the computational domain in a two-dimensional case was created. It consisted of the around 100 m deep sea layer, and about 100 m wide seabed, 300 m wide permafrost and a deepest dense soil sheet beneath it. The bottom layer contained an oil deposit, which was gradually draining due to oil migration [1]. The layout for the problem is presented in Fig. 1. The widths of the layers correspond to the mean values, registered in this geographical region. The boundaries between all areas were significantly curvilinear. The dissolution of the reservoir was represented as the gradual change of its size.

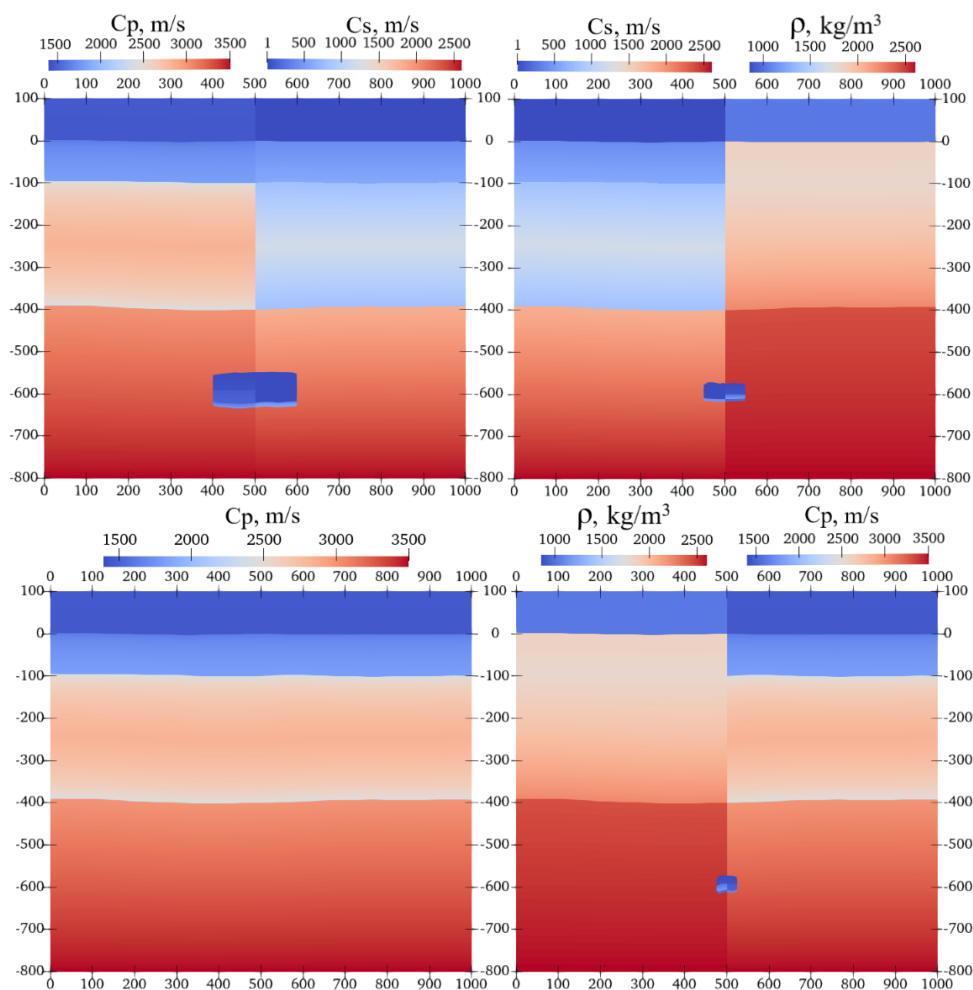


Figure 1. Layout of the computational domain and elastic parameters of layers

1.3.1. Material parameters

To construct an adequate geological model, the ground was considered to be a vertically inhomogeneous medium. The dependencies of the elastic parameters on depth obtained from the experimental data [16] were used as a basis for formulation of the gradient corresponding to each layer. All parameters remained within their characteristic values as noted in [16, 26]. The final relations between the elastic parameters and the depth are presented in Figs. 2–3. In the water layer, the change in the velocity of the pressure waves is associated with the temperature gradient, often observed in the northern seas [14]. Thus, the pressure wave velocity changed according to the similar trend. Moreover, in this layer, the density was equal to 1024 kg/m^3 , and the shear wave velocity was set to 1 m/s .

Under the water, the transition from the thawed soil of the seabed to the frozen soil of the permafrost layer was reproduced based on a gradual decrease in temperature. However, as the temperature rises and stabilizes, the permafrost ends with a transition to a denser clay layer, which was reflected in the density change. For the velocities of the pressure waves, a standard power-law change in parameters at shallow depths was set. For greater depths, a linear increase was used. Additionally, the shear wave velocity and density were changed linearly. The drying of the deposit was modeled by a successive size reduction of the oil reservoir. Inside it, a small

transition was considered from the upper oil layer to the water base and the thawed bottom of the reservoir.

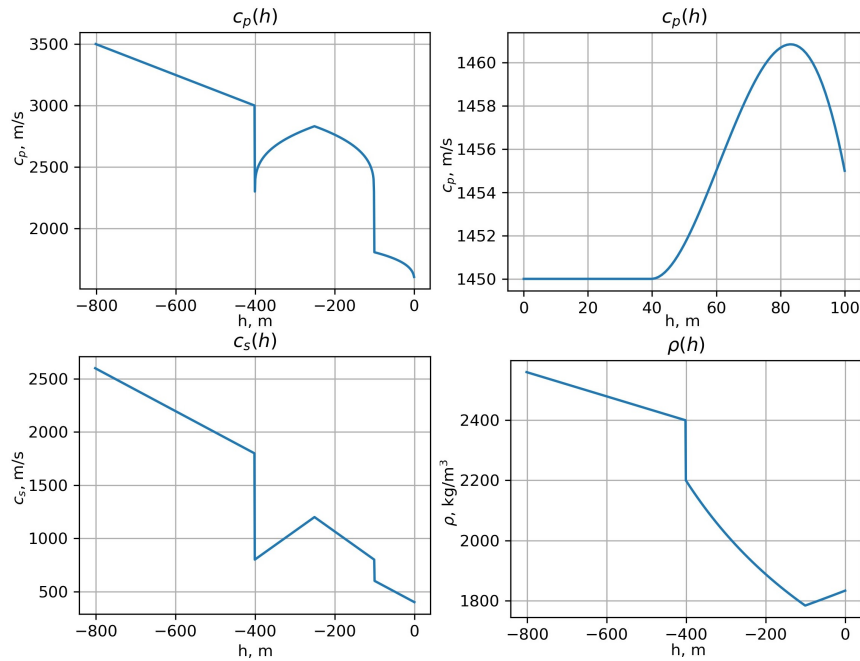


Figure 2. Elastic parameters in water and soil without a reservoir along the depth

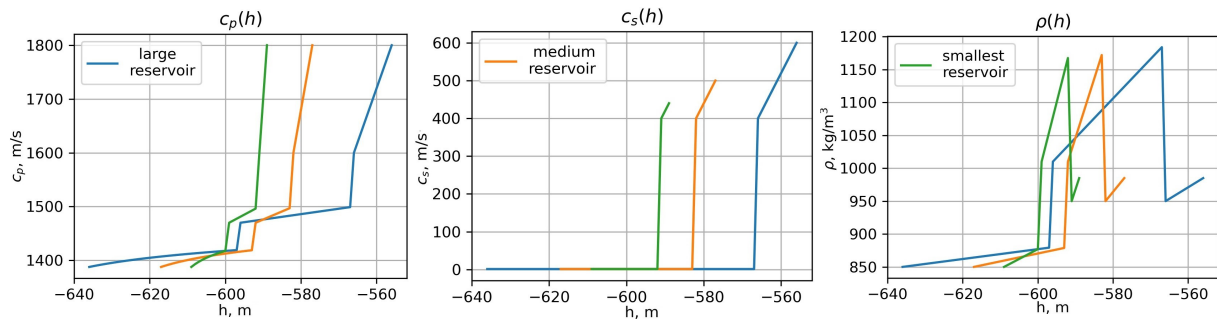


Figure 3. Elastic parameters in a reservoir along the depth: oil layer on the surface of water and wet soil

1.3.2. Grid parameters

There are two basic approaches to creation of computational grids. The first one consists of using explicit contact conditions and creating grids for each area of the domain. According to the second approach, one rectangular grid is created and in each node corresponding material parameters are set. Both approaches allow for reproducing of all wave fronts and results in similar wave patterns [8, 25], however, both also have downfalls.

The second method is characterized by an increase in the amplitude of individual waves. There may also be a lack of smoothness in the incoming signals due to the step-wise nature of the sections of geological media. For the first method, it is necessary to create curvilinear grids, which, if constructed incorrectly, can distort the shape of the wave fronts. Additional numerical errors can also be introduced by nonphysical boundaries within layers with inclusions

when constructing structured grids. Such effects lead to the inability to compute the difference between results obtained in different setups. This is a significant drawback as this technique is often used for analyzing processes in dynamics, and is also employed in this work.

Moreover, the gradual nature of elastic parameters of the layers incentivizes the usage of the second approach. Thus, in order to save time, computational resources and make it easier to set material parameters, one rectangular grid of 1000×900 cells was created. In order to account for the curvilinear boundaries, the dependencies in Figs. 2–3 were interpolated on the grid for each layer separately. Therefore, in each node, the elastic parameters of the corresponding areas were set. To fulfill the Courant stability condition, the time step was set equal to 0.1 ms. The total calculation time was 1 s. The top boundary of the domain was considered to be free, on the other sides, characteristic absorbing conditions were placed. The seismic point source was described as the 30 Hz Ricker pulse. Two different signal generator's positions were compared: on the water surface and on the seabed. The receivers were placed on the water surface and on the seabed with the 2 m horizontal spacing.

2. Simulation Results

As a result of the calculations, spatial wave fields and synthetic seismograms were produced (see Figs. 4–6).

2.1. Wave Patterns

The wave pattern analysis shows that the wave fronts generated by both sources are similar to each other. However, it is noticeable that when the source position is set on the sea surface, the amplitude of some waves decreases. This is related to the fact that, as water is an acoustic medium, it partially absorbs some waves. Moreover, this is also the reason why there are no surface waves, such as Rayleigh waves [17], observed on the upper boundary. However, when the source is located on the seabed, surface Stoneley (also called Scholte) waves [22] appear and propagate along the water floor.

As the position of the source at the seabed is closer to the reservoirs, the initial wave reaches the deposit earlier. Nonetheless, in both cases, while propagating through the stratified medium, this wave undergoes multiple reflections. Moreover, the gradient of elastic parameters seem to result in appearance of additional oscillations in the wave. The narrow top layers create a wave channel which leads to the formation of seismic ghost reflections [4] which can conceal the initial wave rebounded from the deposit.

These reflections from the reservoirs are the crucial point of the seismology which help identify the internal processes in the ground such as oil migration. It is worth noting that all such reflections, as shown in Fig. 4, have a similar rounded shape which originate from the borders of the layer. As the deposit dissolves, the shape transforms from the oval to the spherical one. Moreover, when the size of the reservoir decreases, its upper boundary moves farther from the day surface which is indicative of the change in the travel time that can be expected in the considered setting. Overall, curvilinear boundaries account for asymmetry in the resulting pictures.

Additionally, in Fig. 4, several computational artifacts are present. The point signal leaves a noticeable cross-shaped trace in the wave patterns, especially in case of the generator on the water surface. Moreover, non-physical reflections from the sides of the domain can appear due

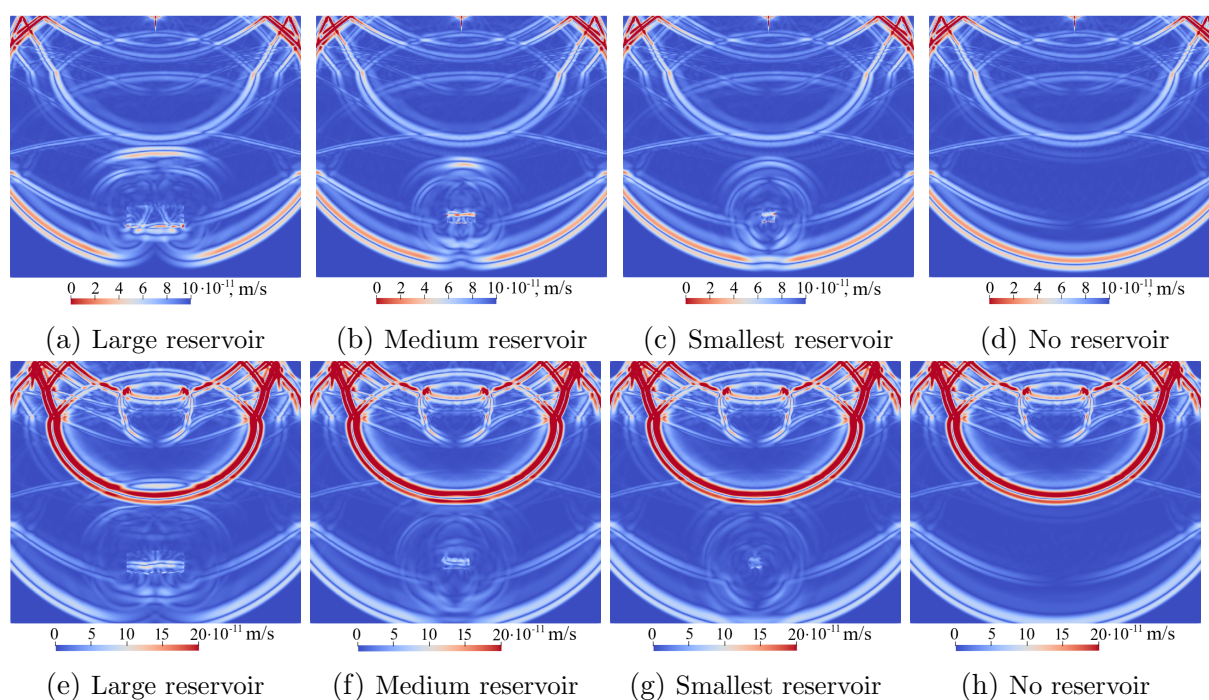


Figure 4. Wave fields. The top row is generated for the source in the water ($T = 0.35$ s), the bottom row – for the source on the seabed ($T = 0.3$ s)

to the well known problem of the absorbing boundary conditions. However, these effects do not influence the results significantly.

2.2. Synthesized Seismograms

All of the effects noted in the wave patterns can also be clearly identified in synthetic seismograms (Figs. 5–6). When the position of the sources and the receivers coincide (first and last rows in Fig. 5), the seismograms show the presence of the initial impulse above other signals. Additionally, in case of the source in the seabed (first row), the noted Stoneley surface waves are also present. However, when the signal generator is moved to the water surface, the signs of these waves (third row) can also be seen, although they are not that prominent.

The second most distinguished signals are the first reflections of the initial wave from the water surface and the seabed. These waves are followed by multiple responses formed by the waves' circulation between these boundaries. Thus, seismic ghost reflections [4] produce a lot of noise in the seismograms and make the resulting seismograms almost indistinguishable between each other as the oil deposit drains. Moreover, in case of the placement of the sources and the signal receivers on the same surface, surface waves further conceal the rest of the picture. Therefore, the most convenient position of the receivers should not coincide with the position of the sources.

Additionally, the discovered non-physical reflections from the sides of the domain can also hinder the analysis of the calculations. As a result, the difference between the settings can be noted only about 0.5 s after the calculations in the central parts of each seismogram. Therefore, they require further processing as it is hard to identify oil migration. To solve this problem, the differences between the settings with and without reservoirs were calculated. In practice, in order to trace down the occurring processes, this procedure can be performed after consecutive

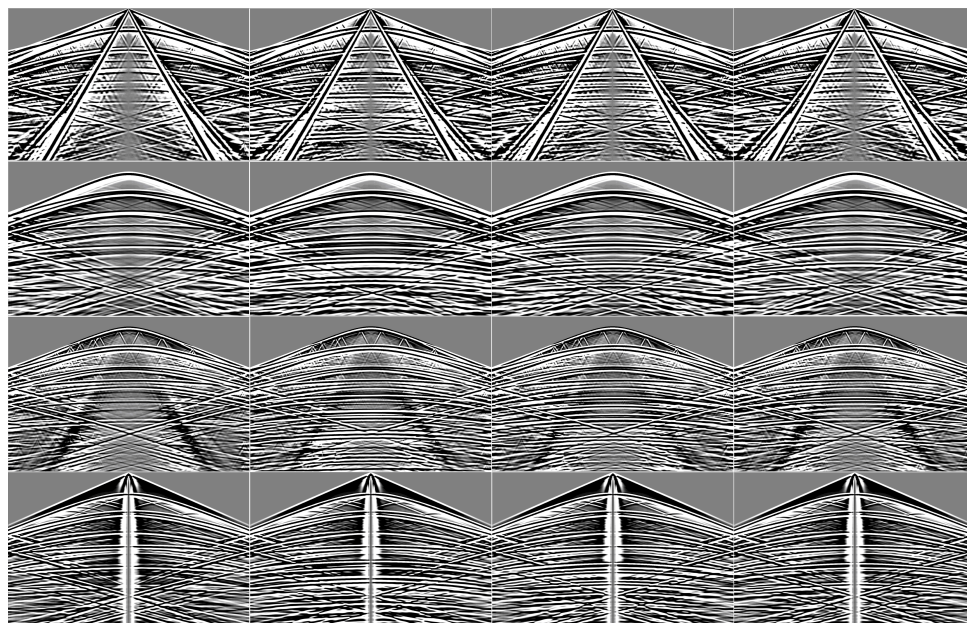


Figure 5. Synthetic seismograms. The top two rows represent the source in the seabed, the bottom two rows – the source on the water. The first and the third rows obtained from the receivers on the seabed, the second and the fourth – from the water surface. The first column corresponds to the case without reservoir, the second one – with the larger reservoir, the third one – with the medium one, the fourth one – with the smallest one

measurements or when the general geometry of the seabed is well-known beforehand. As part of numerical analysis, this technique allows us to investigate the signal from the reservoir separately from other responses.

The resulting seismograms are presented in Fig. 6. The picture shows that the signals of different reservoirs are very similar. In accordance with the wave patterns, the travel time decreases with the placement of sources closer to the oil deposit. Moreover, the dissolution of the reservoirs also results in a slight decrease in the travel time. Another notable difference between the settings previously shown in wave patterns is the change in the form of the reflection from the reservoirs which coincides with the transformation of the shape of the responses on the seismograms. At the same time, it becomes harder to identify the changes as the reservoirs become smaller. So, we can conclude that it is crucial to obtain seismic measurements of the initial state of the oil deposit for further gradual comparisons.

Conclusion

As the result of conducted simulation, the direct problem of marine seismology in the Arctic during oil migration was solved. The distinctive feature of this work is the creation of the computational domain that encompasses the basic features of the offshore areas, including the permafrost ground, and the reflection of gradient change of elastic parameters in the layered ground obtained in real measurements. The usage of the grid-characteristic method allowed for effective solution of problem accelerated by the parallelization using OpenMP. Deposit dissolution was accounted for by the gradual decrease in the size of the reservoir.

Calculated wave patterns and synthesized seismograms not only showed the responses from the reservoir but also demonstrated complex wave phenomena, such as surface waves and seismic

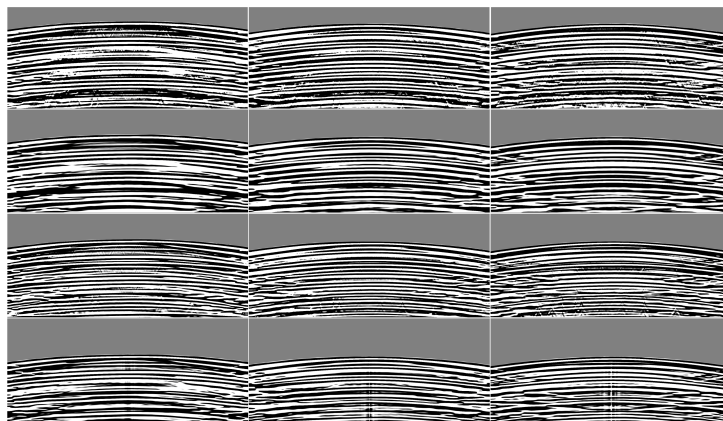


Figure 6. Differences between seismograms with and without oil deposit. The placement of the rows correspond to Fig. 5. The first column represents the setting with the larger reservoir, the second one – with the medium reservoir, the third one – with the smallest reservoir. Top one-third of each picture is not included

ghost reflections. Due to the presence of such waves, the obtained seismograms contain a lot of noise and require further processing. Therefore, in situ measurements should be conducted consecutively to register initial state of the oil deposit for further gradual comparisons. More adequate estimations can be conducted based on the real data and specific deposits. The precise reason for oil migration should also be taken into account, in order to be able to identify particular attributes of the process. Comparison of the results of the simulations in the plane problem to the three-dimensional case can also be noted as the direction for further work.

Acknowledgements

The reported study was funded by the Russian Science Foundation, project No. 23-11-00035, <https://rscf.ru/project/23-11-00035/>.

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. Beletskaya, S.: Mechanisms and factors of primary oil migration. modeling of primary migration processes (in russian). Oil and Gas Geology. Theory and Practice 2, 28 (2007), <http://www.ngtp.ru/rub/1/028.pdf>
2. Biryukov, V.A., Miryakha, V.A., Petrov, I.B., Khokhlov, N.I.: Simulation of elastic wave propagation in geological media: Intercomparison of three numerical methods. Computational Mathematics and Mathematical Physics 56, 1086–1095 (2016). <https://doi.org/10.1134/s0965542516060087>
3. Chilingar, G., Katz, S., Fedin, L., *et al.*: Primary versus secondary migration of oil. Journal of Sustainable Energy Engineering 2(4), 323–330 (2015). <https://doi.org/10.7569/JSEE.2014.629517>

4. Fatyanov, A.: A wave method of suppressing multiple waves for any complex subsurface geometry. *Numerical Analysis and Applications* 5, 187–190 (2012). <https://doi.org/10.1134/s1995423912020140>
5. Favorskaya, A.V., Petrov, I.B.: Grid-characteristic method. In: Favorskaya, A., Petrov, I. (eds.) *Innovations in Wave Processes Modelling and Decision Making, Smart Innovation, Systems and Technologies*, vol. 90, pp. 117–160. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76201-2_5
6. Favorskaya, A.V., Petrov, I.B.: Calculation of seismic stability of buildings in the Far North using the Grid-Characteristic Method. *Lobachevskii Journal of Mathematics* 45(1), 213–222 (2024). <https://doi.org/10.1134/S1995080224010153>
7. Golubev, V.I., Nikitin, I., Beklemysheva, K.A.: Model of fractured medium and nondestructive control of composite materials. *Chinese Journal of Aeronautics* 37(2), 93–99 (2024). <https://doi.org/10.1016/j.cja.2023.11.023>
8. Guseva, E.K., Golubev, V.I., Petrov, I.B.: Investigation of wave phenomena during the seismic survey in the permafrost areas using two approaches to numerical modeling. *Lobachevskii Journal of Mathematics* 45(1), 231–238 (2024). <https://doi.org/10.1134/S1995080224010190>
9. Guseva, E.K., Golubev, V.I., Petrov, I.B.: Investigation of Wave Phenomena in the Offshore Areas of the Arctic Region in the Process of the Seismic Survey. *Lobachevskii Journal of Mathematics* 45(1), 223–230 (2024). <https://doi.org/10.1134/S1995080224010189>
10. Guseva, E., Golubev, V., Petrov, I.: Linear quasi-monotone and hybrid grid-characteristic schemes for the numerical solution of linear acoustic problems. *Numerical Analysis and Applications* 16, 112–122 (2023). <https://doi.org/10.1134/S1995423923020027>
11. Henderson, J., Loe, J.S.P.: The prospects and challenges for Arctic oil development. The Oxford Institute for Energy Studies (2014). <https://doi.org/10.26889/9781784670153>
12. Khokhlov, N., Ivanov, A., Zhdanov, M., *et al.*: Applying OpenCL technology for modelling seismic processes using grid-characteristic methods. In: Vishnevskiy, V., Samouylov, K., Kozyrev, D. (eds.) *Distributed Computer and Communication Networks. DCCN 2016, Communications in Computer and Information Science*, vol. 678, pp. 577–588. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51917-3_49
13. Khokhlov, N.I., Petrov, I.B.: Application of the grid-characteristic method for solving the problems of the propagation of dynamic wave disturbances in high-performance computing systems (in russian). *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)* 31(6), 237–252 (2019). [https://doi.org/10.15514/ispras-2019-31\(6\)-16](https://doi.org/10.15514/ispras-2019-31(6)-16)
14. Khutorskoy, M.D., Akhmedzyanov, V.R., Ermakov, A.V., *et al.*: Geothermics of the Arctic seas (in russian). In: *Transactions of the Geological Institute*, vol. 605. GEOS, Moscow (2013)

15. Kruk, M.N., Nikulina, A.Y.: Economic estimation of project risks when exploring sea gas and oil deposits in the Russian Arctic. *International Journal of Economics and Financial Issues* 6(2), 138–150 (2016), <https://dergipark.org.tr/tr/pub/ijefi/issue/31980/352620>
16. Nazarov, G.N.: Methodical instructions for complex seismic-geological and engineering-geological studies using portable seismic exploration equipment (in russian). Publishing house VIA, Moscow (1969)
17. Nkemzi, D.: A new formula for the velocity of Rayleigh waves. *Wave Motion* 6(2), 199–205 (1997). [https://doi.org/10.1016/s0165-2125\(97\)00004-8](https://doi.org/10.1016/s0165-2125(97)00004-8)
18. Petrov, D.I., Petrov, I.B., Favorskaya, A.V., Khokhlov, N.I.: Numerical solution of seismic exploration problems in the Arctic region by applying the grid-characteristic method. *Computational Mathematics and Mathematical Physics* 56, 1128–1141 (2016). <https://doi.org/10.1134/s0965542516060208>
19. Petrov, I.B., Khokhlov, N.I.: Modeling 3D seismic problems using high-performance computing systems. *Mathematical Models and Computer Simulations* 6, 342–350 (2014). <https://doi.org/10.1134/s2070048214040061>
20. Romanovsky, V.E., Drozdov, D.S., Oberman, N.G., *et al.*: Thermal state of permafrost in Russia. *Permafrost Periglac* 21, 136–155 (2010). <https://doi.org/10.1002/ppp.683>
21. Rusanov, V.V.: Difference schemes of the third order of accuracy for the forward calculation of discontinuous solutions. *Doklady Akademii Nauk* 180, 1303–1305 (1968), <https://www.mathnet.ru/eng/dan33936>
22. Scholte, J.: On the Stoneley wave equation. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen* 45(1), 20–25 (1942)
23. Shearer, P.M.: Introduction to seismology. Cambridge university press (2019). <https://doi.org/10.1017/9781316877111>
24. Stognii, P.V., Khokhlov, N.I., Petrov, I.B.: Numerical modeling of wave processes in multilayered media with gas-containing layers: Comparison of 2D and 3D models. *Doklady Mathematics* 100, 586–588 (2019). <https://doi.org/10.1134/S1064562419060152>
25. Stognii, P.V., Khokhlov, N.I., Petrov, I.B., Favorskaya, A.: The comparison of two approaches to modeling the seismic waves spread in the heterogeneous 2D medium with gas cavities. In: Favorskaya, M.N., Favorskaya, A.V., Petrov, I.B., Jain, L.C. (eds.) *Smart Modelling For Engineering Systems, Smart Innovation, Systems and Technologies*, vol. 214, pp. 101–114. Springer, Cham (2021). https://doi.org/10.1007/978-981-33-4709-0_9
26. Trifonov, B.A., Milanovsky, S.Y., Nesynov, V.V.: Assessment of seismic impacts in conditions of permafrost degradation (in russian). *Bulletin of KRAUNC. Series: Earth Sciences* 56(4), 59–74 (2022). <https://doi.org/10.31431/1816-5524-2022-4-56-59-74>

A Modification of Adaptive Greedy Algorithm for Solving Problems of Fractured Media Geophysics

Alena V. Favorskaya^{1,2} , *Nikolay I. Khokhlov*^{1,2} ,
*Dmitry A. Podlesnykh*¹ 

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

Nowadays, the issue of direct modeling of seismic exploration problems is becoming increasingly important due to the development of a new field of application of such algorithms as generation of a training samples for subsequent solution of the appropriate inverse problem using neural networks. This challenges scientists to develop corresponding parallel algorithms and improve their efficiency. The current manuscript is devoted to the algorithm for decomposing a large number of individual computational grids of various sizes for a large number of MPI processes using the example of a 3D direct problem of seismic exploration of geological media treating the complex topology of the Earth's surface, the complex shape of interfaces between geological layers and a large number of explicitly treated geological fractures, that are not aligned with the coordinate axes. Three modifications of the grid-characteristic numerical method on Chimera and curvilinear computational grids are compared with each other. The dependence on different numbers of fractures is studied. A large number (several hundreds or thousands) of fractures in the geological media significantly increases the amount of transmitted data, which imposes requirements on the developed modification of the greedy algorithm.

Keywords: decomposition, greedy algorithm, large number of grids, large number of fractures, Chimera meshes, patch grids, grid-characteristic method, elastic wave, seismic wave, 3D simulation.

Introduction

Methods for solving inverse problems can be divided into inversion, migration, and the use of neural networks, with each approach imposing its own parallelization features [9, 16]. With the development of neural networks, the issue of direct modeling of seismic exploration problems is becoming increasingly relevant, as a new area of application for this type of computer modeling is actively developing. Training of neural networks for solving inverse problems of wave dynamics on simulated data has been actively carried out in recent years [7, 8, 14, 15]. Another new way of using solvers for direct problems of wave dynamics due to the development of neural networks is the use of deep learning to obtain seismograms from a velocity model [22], which also creates a need to use synthetic training samples. This means that the development of new high-precision methods for three-dimensional modeling of seismic waves in geological media, treating the complex surface topography and internal structure, with explicit identification of fractures, that are not co-directed with the coordinate axes, is acquiring additional significance and relevance. In turn, the question arises of the quality of parallelization of the developed algorithms and taking into account the efficiency of parallelization as a factor in choosing the best method of computer modeling.

In recent years, the following numerical methods for solving direct problems of seismic exploration, the development of which is underway, have become popular, i.e., finite-difference method on staggered grids [21], the discontinuous Galerkin method [20, 24], and the spectral element method [23].

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

²Scientific Research Institute for System Analysis of the National Research Centre "Kurchatov Institute", Moscow, Russian Federation

The following two main types of accounting for fracturing in geological environments are distinguished, i.e., explicit treatment of fractures [4, 6, 10] and different averaged models [4, 17].

The paper [24] is devoted to parallelized modification of the discontinuous Galerkin method for solving geophysical problems, but a large number of individual computational grids does not arise. The paper [1] considers the issue of decomposition of structured and unstructured grids when solving problems of linear elasticity and using fine and coarse grids with different integration steps. The papers [11, 18] are devoted to the decomposition of methods using overlapping patch grids, which are used to increase the number of nodes in the desired area.

In this manuscript, we compare three modifications of the grid-characteristic method in terms of their parallelization efficiency, i.e., the grid-characteristic method on curvilinear structured grids [5] (type #1), the grid-characteristic method on Chimera grids for describing the topography of the Earth's surface and interfaces between geological rocks [2], which we combine with two methods for treatment fracturing, i.e., the use of Chimera rotated Cartesian grids around fractures [12] (type #2) and the use of overlapping patch grids around fractures [13] (type #3). The novelty of this study is that we consider a different large numbers of fractures and a larger number of MPI processes.

The article is organized as follows. Section 1 is devoted to problem statement and mathematical model. In section 2 we introduce decomposition and partitioning algorithms. Section 3 contains speed up testing. Conclusion summarizes the study and points directions for further work.

1. Problem Statement and Mathematical Model

To describe the propagation of seismic waves in geological media with complex boundaries (see Fig. 1) and multiple fractures (see Figs. 2–4), we seek a solution to the following initial boundary value problem of the hyperbolic elastic wave equation.

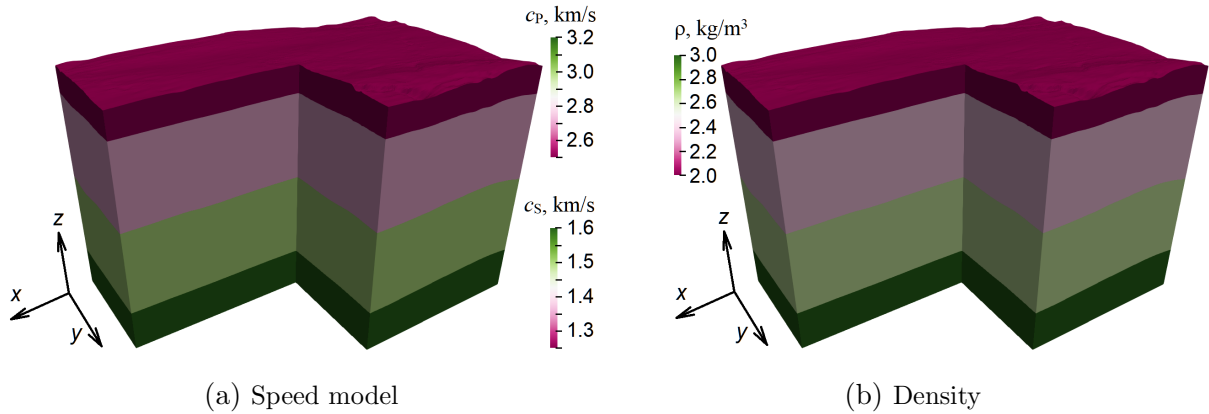


Figure 1. Geological model with 4 layers

The system of equations inside the integration domain in each geological layer $\#l$ is written as follows:

$$\begin{aligned} \frac{\partial \boldsymbol{\sigma}_l(t, \mathbf{r})}{\partial t} = & \rho (c_{P,l}^2 - 2c_{S,l}^2) (\nabla \cdot \mathbf{v}_l(t, \mathbf{r})) \mathbf{I} + \\ & + \rho c_{S,l}^2 (\nabla \otimes \mathbf{v}_l(t, \mathbf{r}) + (\nabla \otimes \mathbf{v}_l(t, \mathbf{r}))^\top) - P(t, \mathbf{r}) \mathbf{I}, \end{aligned} \quad (1)$$

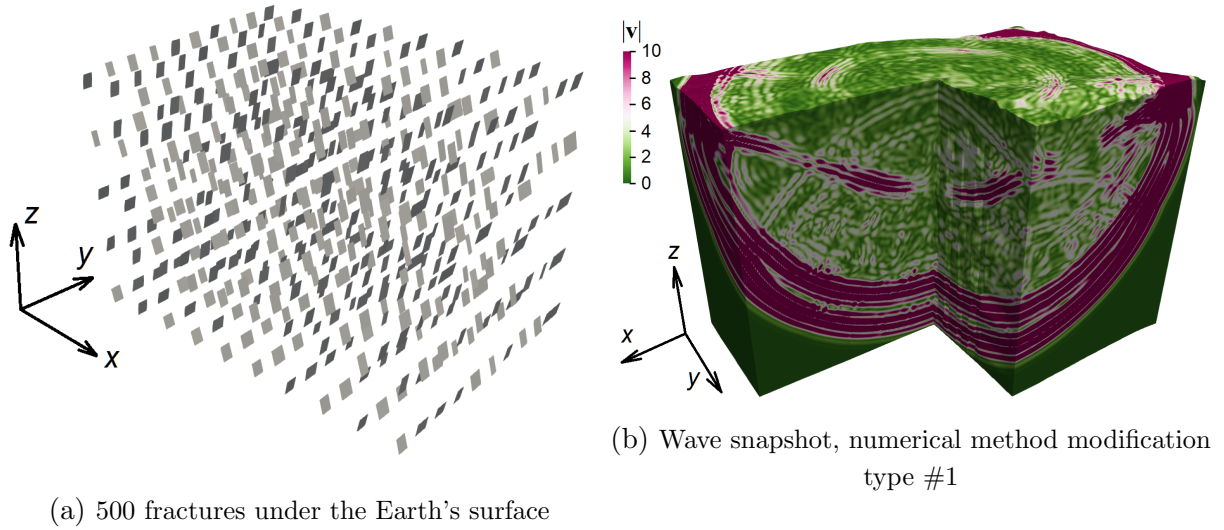


Figure 2. Example with 500 fractures in geological media

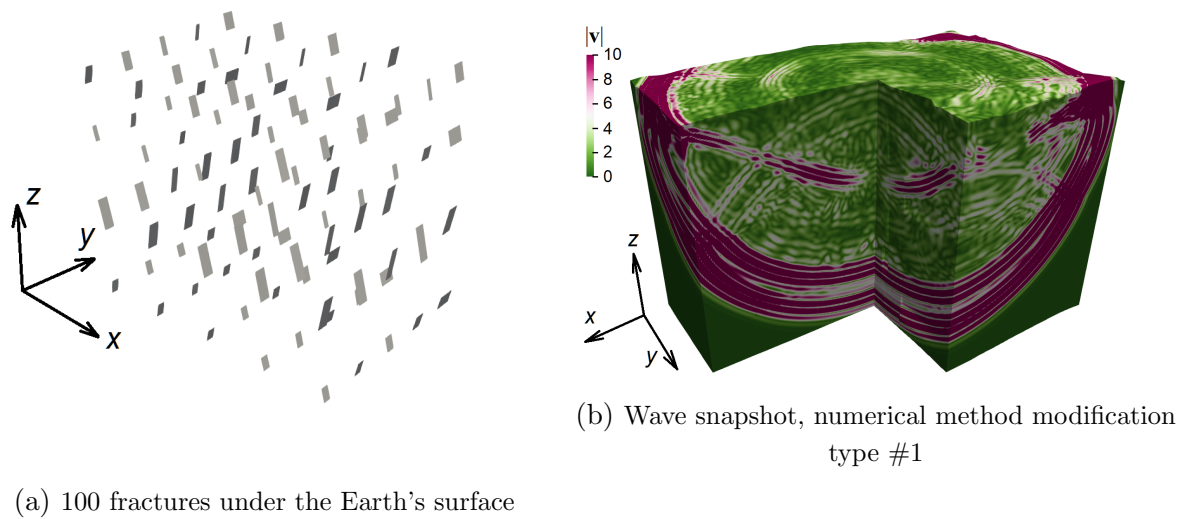


Figure 3. Example with 100 fractures in geological media

$$\rho_l \frac{\partial \mathbf{v}_l(t, \mathbf{r})}{\partial t} = (\nabla \cdot \boldsymbol{\sigma}_l(t, \mathbf{r}))^\top,$$

$$\mathbf{r} \in \Omega_l, t \in [0, T], l \in [1, L].$$

Here, in Eq. (1), the right-hand side (the seismic wave source) is described by the following expression:

$$P(t, \mathbf{r}) = \begin{cases} \left(1 - 2(\pi f t - 1.3\sqrt{6})^2\right) \exp\left(-(\pi f t - 1.3\sqrt{6})^2\right), & t \in \left[0, 3.9 \frac{\sqrt{6}}{\pi f}\right], \mathbf{r} \in \Xi, \\ 0, & t \notin \left[0, 3.9 \frac{\sqrt{6}}{\pi f}\right], \mathbf{r} \notin \Xi. \end{cases} \quad (2)$$

Here, in Eq. (2), the numerical coefficients arise in order to use the exact zero value on that part of the real time axis where the Ricker wavelet is sufficiently small.

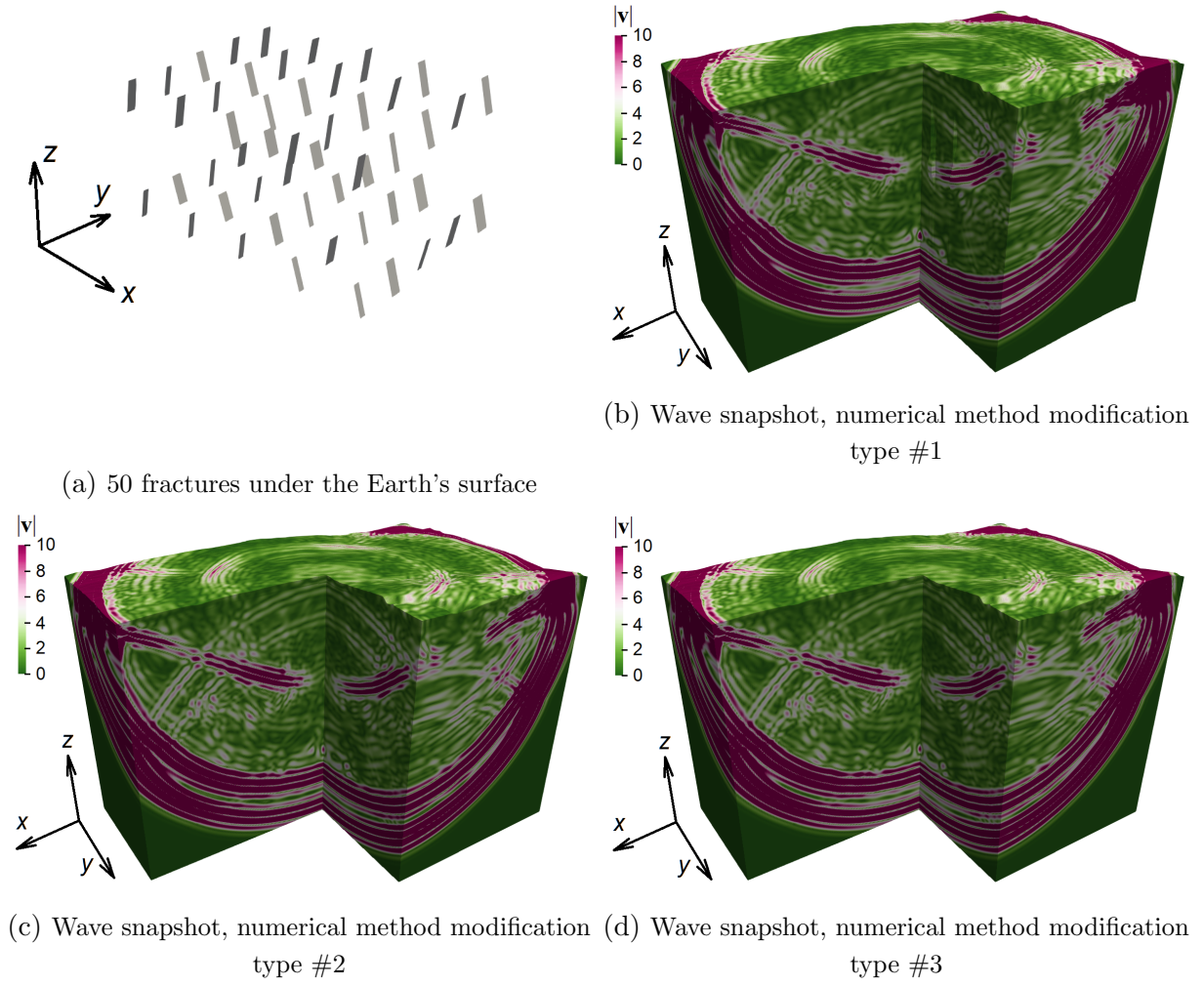


Figure 4. Example with 50 fractures in geological media

The following free boundary condition is established on the upper layer #1, at the contact with air:

$$\boldsymbol{\sigma}_1(t, \mathbf{r}) \cdot \mathbf{m}_1(\mathbf{r}) = 0,$$

$$\mathbf{m}_1(\mathbf{r}) \perp \Gamma_0, \mathbf{r} \in \Gamma_0, t \in [0, T].$$

At the boundaries of the sections of geological rocks, the following contact conditions are established:

$$\boldsymbol{\sigma}_{l-1}(t, \mathbf{r}) \cdot \mathbf{m}_l(\mathbf{r}) = \boldsymbol{\sigma}_l(t, \mathbf{r}) \cdot \mathbf{m}_l(\mathbf{r}),$$

$$\mathbf{v}_{l-1}(t, \mathbf{r}) = \mathbf{v}_l(t, \mathbf{r}),$$

$$\mathbf{m}_l(\mathbf{r}) \perp \Gamma_l, \mathbf{r} \in \Gamma_l, t \in [0, T], l \in [2, L].$$

The following contact condition is established, which is a special case [12] of Schoenberg linear slip model [19], to describe the scattering of waves on the fracture # i at the location of the fracture:

$$\mathbf{m}_i(\mathbf{r}) \cdot \boldsymbol{\sigma}_i^L(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r}) = \mathbf{m}_i(\mathbf{r}) \cdot \boldsymbol{\sigma}_i^R(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r}),$$

$$\boldsymbol{\sigma}_i^L(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r}) - (\mathbf{m}_i(\mathbf{r}) \cdot \boldsymbol{\sigma}_i^L(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r})) \mathbf{m}_i(\mathbf{r}) = 0,$$

$$\sigma_l^R(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r}) - (\mathbf{m}_i(\mathbf{r}) \cdot \sigma_l^R(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r})) \mathbf{m}_i(\mathbf{r}) = 0,$$

$$\mathbf{v}_l^L(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r}) = \mathbf{v}_l^R(t, \mathbf{r}) \cdot \mathbf{m}_i(\mathbf{r}),$$

$$\mathbf{m}_i(\mathbf{r}) \perp \Theta_i, \mathbf{r} \in \Theta_i, t \in [0, T], i \in [1, F].$$

Here and further in the text, σ is Cauchy stress tensor, \mathbf{v} is velocity, c_P , c_S , and ρ are elastic parameters (P-wave speed, S-wave speed, and density), f means frequency, \mathbf{m} means unit normals, L is the number of geological layers, F is the number of fractures, Ξ is the source location, Γ_l is the boundary between l and $l + 1$ layers, Θ_i is the i fracture location.

2. Decomposition and Partitioning Algorithms

In this manuscript, we compare how the proposed decomposition algorithm works for three modifications of the grid-characteristic method, also varying the number of geological fractures. Examples of snapshots of wave fields are shown in Figs. 4b, 4c, and 4d for computational method modification types #1, #2, #3, respectively.

In calculations with numerical method modification type #1, we have the following set of computational grids:

- large computational grids, curvilinear structured grids covering geological layers in the amount of geological layers L , see Fig. 5a,
- tiny computational grids of the minimal size for identifying a rupture on a fracture in the amount of $2F$, see Fig. 6a.

That is, we have

$$G_L = L$$

large grids and

$$G_S = 2F$$

small grids.

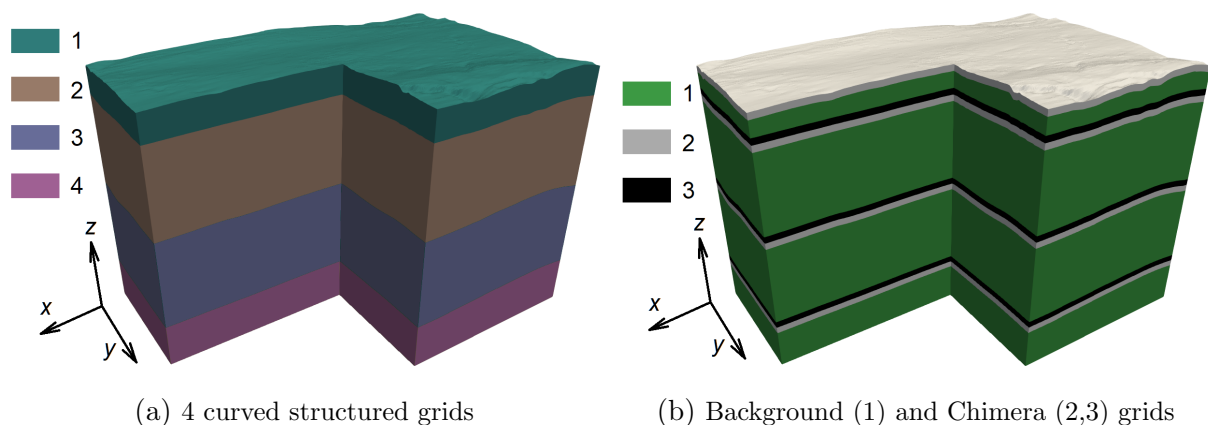
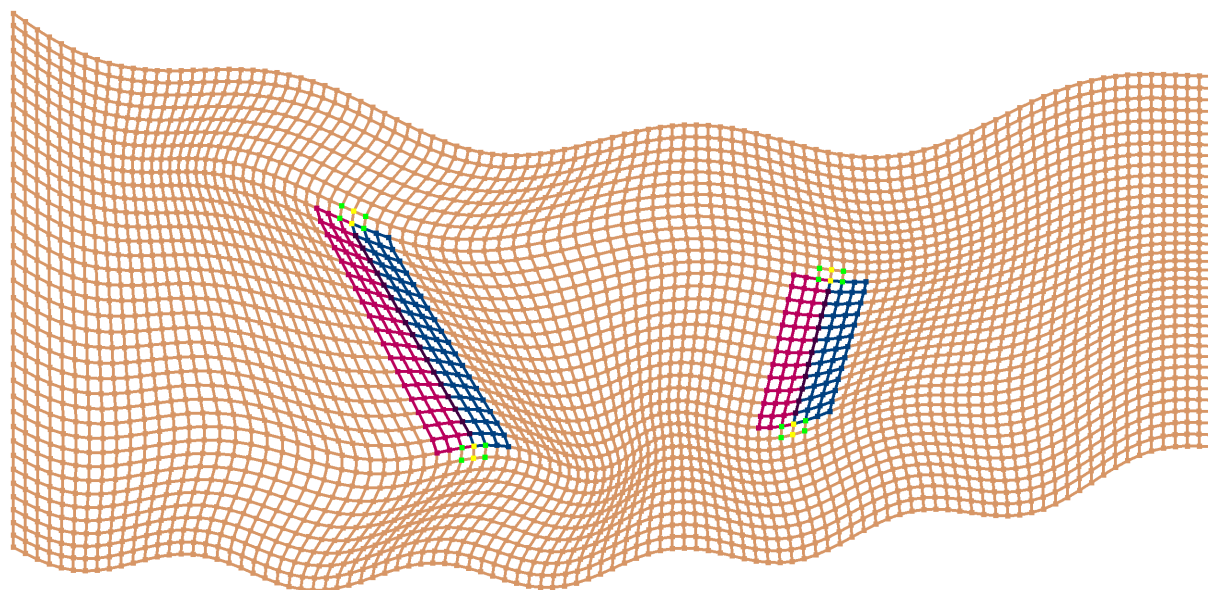
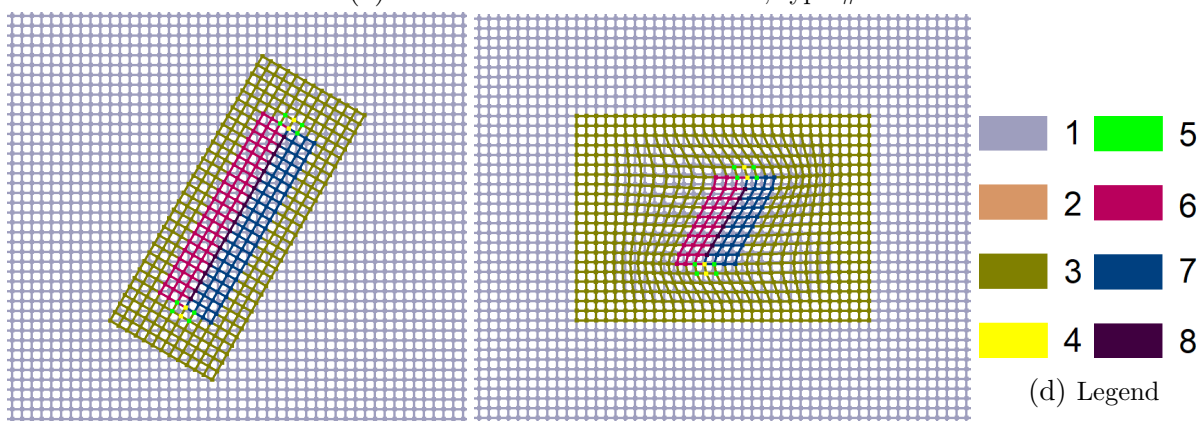


Figure 5. Large computational grids

At the legend in Fig. 6d, 1 is a part of large background Cartesian grid, 2 is a part of large curved structured grid covering geological layer, 3 is Chimera rotated Cartesian or curved structured patch grid around the fracture, 4 means nodes of grids of types 2 or 3, to which data is copied to smooth out the gap at the ends of the fracture, 5 means nodes of grids of types 2



(a) Numerical method modification, type #1



(b) Numerical method modification, type #2

(c) Numerical method modification, type #3

(d) Legend

Figure 6. Small and tiny computational grids around fractures, OXY, example

or 3, from which data is copied, 6 is left tiny grid of the minimal size around the fracture, 7 is right tiny grid of the minimal size around the fracture, 8 is the fracture itself.

In calculations with numerical method modifications types #2 and #3, we have the following set of computational grids:

- large Cartesian background grid, see Fig. 5b,
- large Chimera curvilinear computational grids along layers in the amount of $2L1$, see Fig. 5b,
- small computational grids around fractures in the amount of F (built according to different algorithms and having different number of nodes for calculations of type #2 (see Fig. 6b) and type #3 (see Fig. 6c), respectively),
- tiny computational grids of the minimal size for highlighting a rupture on a fracture in the amount of $2F$, see Figs. 6b and 6c for numerical method modifications types #2 and #3, respectively.

That is, we have

$$G_L = 8L$$

large grids and

$$G_S = 3F$$

small grids.

The number of computational operations spent on calculations on different types of grids is different. For example, we give below the formulae for calculating the first two Riemann invariants in the three-dimensional case for a Cartesian grid:

$$\omega_{1,2}^X = v_X \mp \frac{1}{\rho c_P} \sigma_{XX},$$

and for a curvilinear structured grid:

$$\omega_{1,2} = \mathbf{n} \cdot \mathbf{v} \mp \frac{1}{\rho c_P} (\mathbf{N}_{00} : \boldsymbol{\sigma}). \quad (3)$$

In Eq. (3), \mathbf{n} is a vector, and \mathbf{N}_{00} is a symmetric tensor of the second rank, depending on the node of the curved computational grid.

Therefore, before starting the calculations, the grid node weighting coefficients for each type of computational grid are determined by running test calculations, which do not take much time. The coefficients used in current study are given in Tab. 1.

Table 1. Weights of nodes in different types of computational grids

computational grid	figure	grid type	method type	weight w
grid covering geological layer	Fig. 5a	curved structured	#1	3.0
large background	Fig. 5b	Cartesian	#2, #3	1.0
large Chimera along interfaces	Fig. 5b	curved structured	#2, #3	2.0
small Chimera around fractures	Fig. 6b	Cartesian rotated	#2	1.0
small patch around fractures	Fig. 6c	curved structured	#3	4.0
tiny around fractures	Fig. 6a	curved structured	#1	2.0
tiny around fractures	Fig. 6b	Cartesian rotated	#2	1.0
tiny around fractures	Fig. 6c	curved structured	#3	2.0

We calculate the value

$$N = \sum_{g=1}^{G_L+G_S} w_g N_X^g N_Y^g N_Z^g$$

to perform the decomposition.

The numbers of processes allocated to the large grids are calculated using the following formulae:

$$P_g = \frac{P}{S} w_g N_X^g N_Y^g N_Z^g,$$

$$g \in [1, G_1].$$

Then these real numbers are rounded to an integers and the set of these numbers for all large grids is adjusted so that the sum does not exceed the total number of processes P :

$$P_F = P - \sum_{g=1}^{G_L} P_g,$$

$$P_F \geq 0.$$

Minimal Cross Section algorithm for parallel grid partitioning introduced in [3] is used for these large grids to divide each g grid by a selected number of processes P_g , $g \in [1, G_1]$.

1. All possible partitions of the number of processes P_g into 3 divisors $\{L_X^{g,j}, L_Y^{g,j}, L_Z^{g,j}\}$ are considered, without repetitions based on the known tables of natural numbers divisors. If the P_g has a large amount of divisors, one can choose not all divisors, but only some, approximately equidistant from each other, and including 1 and P_g .

$$P_g = L_X^{j,g} \cdot L_Y^{j,g} \cdot L_Z^{j,g}.$$

2. The sum of the cross sections is calculated for each set of divisors $\{L_X^{g,j}, L_Y^{g,j}, L_Z^{g,j}\}$ using the following formula:

$$S_g^j = \left(L_X^{g,j} - 1\right) \cdot N_Y^g \cdot N_Z^g + \left(L_Y^{g,j} - 1\right) \cdot N_X^g \cdot N_Z^g + \left(L_Z^{g,j} - 1\right) \cdot N_X^g \cdot N_Y^g.$$

3. Such a set of divisors $\{L_X^{g,j}, L_Y^{g,j}, L_Z^{g,j}\}$ is chosen, which provides the minimal value of the cross section S_g^j .

Then the distribution of small (and tiny) grids among the processes begins.

In the case of a modification of the numerical method of type #1, at first the P_F processes are filled with the number of grids sequentially each, until the total number of nodes allocated to the process begins to exceed the average filling of the remaining processes with nodes of large grids $\frac{N}{P}$. In this case and further, the nodes are taken into account with weights. That is, the value $w_g N_X^g N_Y^g N_Z^g$ is considered, and not $N_X^g N_Y^g N_Z^g$. The remaining part of the tiny grids of the minimal size is distributed evenly among all processes evenly.

For modifications of the numerical method of types #2, #3, the coefficient of the ratio K of the average size of the small grid around the fracture to the average size of the tiny grid of the minimal size is calculated. Then,

1. The P_F processes are filled according to the following algorithm sequentially each.
 - (a) We try to place as many small grids around the fractures as possible so that the total number of nodes in the process under consideration does not exceed $\frac{N}{P}$.
 - (b) If possible, we place as many tiny grids of the minimal size as possible.
 - (c) We move on to the next process.
2. We distribute the remaining small grids around the fractures evenly across all processes until we stop at process number P_0 .
3. We place K tiny grids of minimal size into the remaining PP_0 processes.
4. We distribute the remaining tiny grids of minimal size evenly across all processes evenly.

Note that, the P_F number is often zero for an insufficiently large number of fractures and number of MPI processes, see Tables 2–4.

As for large Chimera grids placed along geological layers, for a small number of MPI processes, one process also often has been allocated for each large Chimera grid, see Tables 5 and 6.

Table 2. Dependence of the number of MPI processes allocated to all grids around fractures only P_F on the total number of MPI processes P for the different number of fractures, numerical method modification type #1

P	12	16	20	24	28	32	40	50	60	70	80	90	100	150	250	500
50	1	0	0	1	0	0	1	1	0	0	1	0	0	1	1	2
100	1	0	0	1	1	0	1	1	0	0	1	0	0	1	1	2
500	1	1	0	1	1	1	2	1	1	2	3	2	2	4	6	13
1000	1	1	2	1	1	1	2	2	4	4	4	4	5	8	13	24
2000	1	1	2	2	2	3	3	5	5	6	7	9	10	13	23	47

Table 3. Dependence of the number of MPI processes allocated to all grids around fractures only P_F on the total number of MPI processes P for the different number of fractures, numerical method modification type #2

P	12	16	20	24	28	32	40	50	60	70	80	90	100	150	250	500
50	0	0	2	0	0	1	0	2	0	0	2	0	0	0	3	0
100	0	0	2	0	0	1	0	2	0	0	2	0	4	0	3	7
500	0	1	3	0	0	1	0	3	1	5	3	1	6	9	8	22
1000	0	1	3	5	0	2	6	4	2	7	5	10	8	12	19	39
2000	0	2	4	6	1	3	7	6	11	9	8	13	11	24	35	70

Table 4. Dependence of the number of MPI processes allocated to all grids around fractures only P_F on the total number of MPI processes P for the different number of fractures, numerical method modification type #3

P	12	16	20	24	28	32	40	50	60	70	80	90	100	150	250	500
50	0	0	2	0	0	1	0	2	0	0	2	0	4	0	4	8
100	0	1	2	0	0	1	0	3	0	5	3	0	5	1	6	11
500	0	1	3	5	1	3	7	5	10	8	6	12	10	15	25	56
1000	0	2	4	7	9	4	9	7	13	12	17	16	22	29	51	101
2000	1	3	6	8	11	13	12	18	17	24	30	30	36	47	83	165

Therefore, the decomposition algorithm proposed in this manuscript is not suitable for a small number of MPI processes.

Table 5. Dependence of the number of MPI processes allocated to one large Chimera grid on the total number of MPI processes P for the different number of fractures, numerical method modification type #2

P	12	16	20	24	28	32	40	50	60	70	80	90	100	150	250	500
50	1	1	1	1	1	2	2	3	4	4	5	6	6	9	16	32
100	1	1	1	1	1	2	2	3	4	4	5	6	6	9	16	32
500	1	1	1	1	2	2	2	3	4	4	5	6	6	9	16	31
1000	1	1	1	1	2	2	2	3	4	4	5	5	6	9	15	30
2000	1	1	1	1	2	2	2	3	3	4	5	5	6	8	14	28

Table 6. Dependence of the number of MPI processes allocated to one large Chimera grid on the total number of MPI processes P for the different number of fractures, numerical method modification type #3

P	12	16	20	24	28	32	40	50	60	70	80	90	100	150	250	500
50	1	1	1	1	1	2	2	3	4	4	5	6	6	10	16	32
100	1	1	1	1	1	2	2	3	4	4	5	6	6	10	16	32
500	1	1	1	1	2	2	2	3	3	4	5	5	6	9	15	29
1000	1	1	1	1	1	2	2	3	3	4	4	5	5	8	13	26
2000	1	1	1	1	1	1	2	2	3	3	3	4	4	7	11	22

3. Speed Up Testing

This section presents the graphs of the speedup dependence on the number of MPI processes for various modifications of the numerical method and various numbers of fractures in Fig. 7.

The 9 dependences of the speedup on the number of MPI processes are grouped on the graphs as follows. For the same types of numerical method modification and different numbers of fractures on one graph in Figs. 7a, 7c, and 7e for types #1, #2, #3, respectively. For the same numbers of fractures and different numerical methods modifications on the same graph in Figs. 7b, 7d, and 7f for the numbers of fractures 50, 100, 500, respectively.

Thus, Figs. 7a, 7c, and 7e allow one to see the effect of the number of fractures on the parallelization efficiency. While Figs. 7b, 7d, and 7f allow one to compare the parallelization efficiency of different modifications of the numerical method for the same number of fractures.

Conclusion

In the manuscript, we proposed and studied the algorithm for decomposing a large number of individual computational grids of various sizes using the example of a direct problem of seismic exploration of geological media treating the complex topology of the Earth’s surface, the complex shape of interfaces between geological layers, and a large number of explicitly treated geological fractures, that are not aligned with the coordinate axes. The developed decomposition algorithm was tested for three modifications of the grid-characteristic method on Chimera and curvilinear structured computational grids. The effect of the number of fractures on the efficiency of parallelization was studied.

The modification of the numerical method type #1 is more efficiently parallelized. But, when this modification is applied, the complexity is the construction of curvilinear structured grids, which, on the one hand, allow describing the complex topography of interfaces between geological rocks and the Earth’s surface boundary-conforming, and on the other hand, node by node, geometrically fall into each fracture. Such computational grids are difficult to construct with a large number of geological fractures, both algorithmically and in terms of RAM and time costs at the preprocessing stage.

The modification of the numerical method type #3 is more accurate computationally (see works [12, 13]), but loses in parallelization efficiency in comparison to type #2 with a large number of fractures and a large number of MPI processes and large fracture inclination angles relative to the coordinate axes. This is due to the fact that the modification of the method type

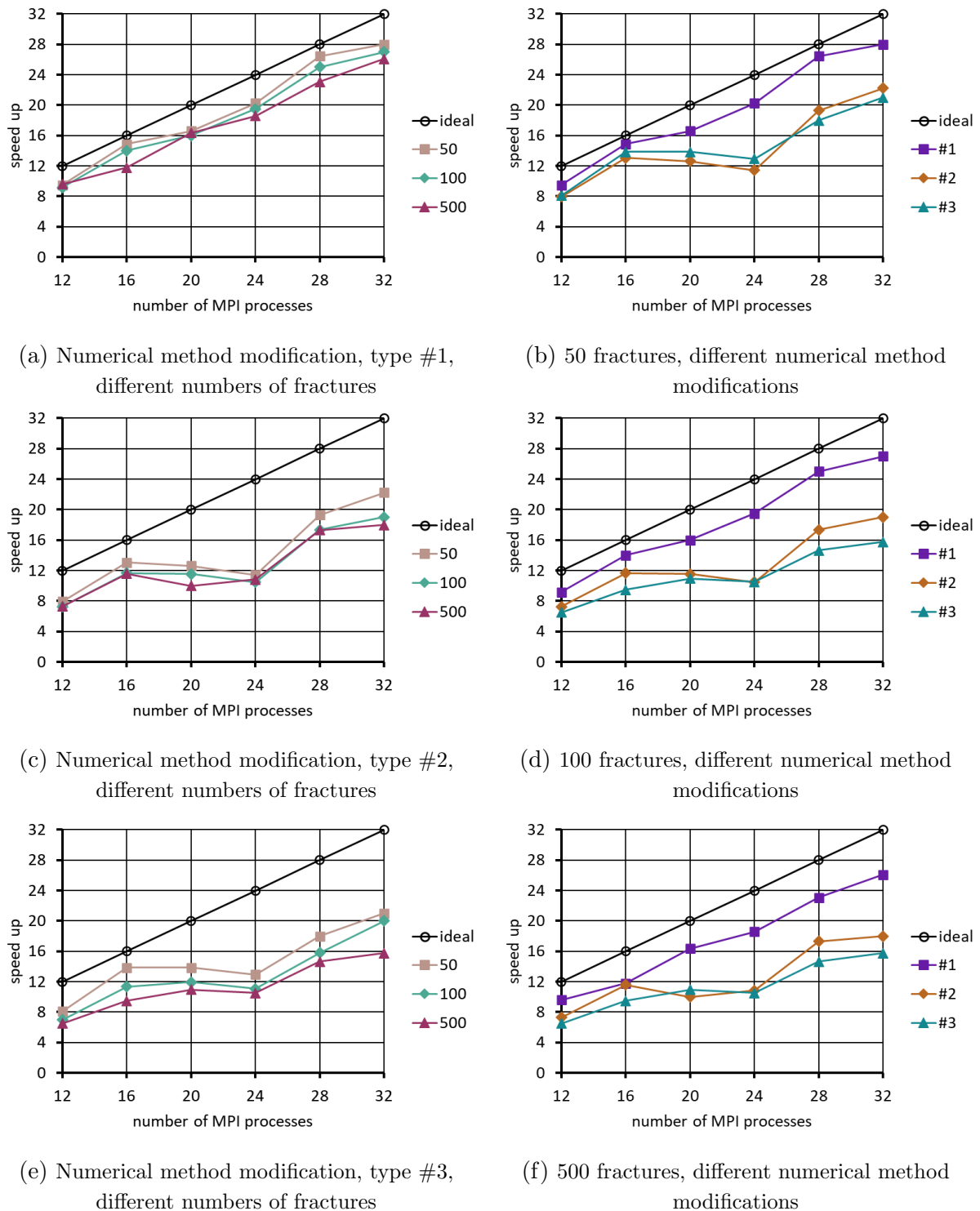


Figure 7. Dependences of the speedup on the number of MPI processes

#3 requires constructing a grid around a fracture of a slightly larger size (see Figs. 6b and 6c) compared to type #2, and this begins to affect under the above conditions.

For all modifications of the numerical method, the growth of fracture number leads to an increase in the volume of transmitted information, which reduces the efficiency of parallelization. The effect is weakest for the modification of the numerical method type #1, since this modification contains only tiny computational grids of minimal size around fractures. In turn,

for the modification of the numerical method type #3, the effect is somewhat stronger than for the modification of the numerical method type #2 for the same reason.

As a direction for further development of the proposed in the manuscript decomposition algorithm, we can name the identification of the dependence of weight coefficients for grids around fractures taking into account the average amount of transmitted data.

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. Badia, S., Martín, A.F., Principe, J.: A highly scalable parallel implementation of balancing domain decomposition by constraints. *SIAM Journal on Scientific Computing* 36(2), C190–C218 (2014), <https://doi.org/10.1007/s11075-022-01268-0>
2. Favorskaya, A., Khokhlov, N.: Accounting for curved boundaries in rocks by using curvilinear and Chimera grids. *Procedia Computer Science* 192, 3787–3794 (2021), <https://doi.org/10.1016/j.procs.2021.09.153>
3. 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: Voevodin, V., Sobolev, S., Yakobovskiy, M., Shagaliev, R. (eds.) *Supercomputing. RuSCDays 2022, Lecture Notes in Computer Science*, vol. 13708, pp. 199–213. Springer, Cham (2022), https://doi.org/10.1007/978-3-031-22941-1_14
4. Favorskaya, A., Petrov, I., Grinevskiy, A.: Numerical simulation of fracturing in geological medium. *Procedia Computer Science* 112, 1216–1224 (2017), <https://doi.org/10.1016/j.procs.2017.08.042>
5. Favorskaya, A.V., Khokhlov, N.I., Petrov, I.B.: Grid-characteristic method on joint structured regular and curved grids for modeling coupled elastic and acoustic wave phenomena in objects of complex shape. *Lobachevskii Journal of Mathematics* 41, 512–525 (2020), <https://doi.org/10.1134/S1995080220040083>
6. Favorskaya, A.V., Petrov, I.B.: The use of full-wave numerical simulation for the investigation of fractured zones. *Mathematical Models and Computer Simulations* 11, 518–530 (2019), <https://doi.org/10.1134/S2070048219040069>
7. Golubev, V., Nikitin, I., Beklemysheva, K.: Model of fractured medium and nondestructive control of composite materials. *Chinese Journal of Aeronautics* 37(2), 93–99 (2024), <https://doi.org/10.1016/j.cja.2023.11.023>

8. Golubev, V., Nikitin, I., Vasyukov, A., Nikitin, A.: Fractured inclusion localization and characterization based on deep convolutional neural networks. *Procedia Structural Integrity* 43, 29–34 (2023), <https://doi.org/10.1016/j.prostr.2022.12.230>
9. Goncharsky, A.V., Romanov, S.Y., Seryozhnikov, S.Y.: Supercomputer technologies in tomographic imaging applications. *Supercomputing Frontiers and Innovations* 3(1), 41–66 (2016), <https://doi.org/10.14529/jsfi160103>
10. Hall, F., Wang, Y.: Seismic response of fractures by numerical simulation. *Geophysical Journal International* 189(1), 591–601 (2012), <https://doi.org/10.1111/j.1365-246X.2012.05360.x>
11. Henshaw, W.D., Schwendeman, D.W.: Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement. *Journal of Computational Physics* 227(16), 7469–7502 (2008), <https://doi.org/10.1016/j.jcp.2008.04.033>
12. Khokhlov, N., Favorskaya, A., Stetsyuk, V., Mitskovets, I.: Grid-characteristic method using Chimera meshes for simulation of elastic waves scattering on geological fractured zones. *Journal of Computational Physics* 446, 110637 (2021), <https://doi.org/10.1016/j.jcp.2021.110637>
13. Khokhlov, N.I., Favorskaya, A.V., Furgailo, V.: Grid-characteristic method on overlapping curvilinear meshes for modeling elastic waves scattering on geological fractures. *Minerals* 12(12), 1597 (2022), <https://doi.org/10.3390/min12121597>
14. Micucci, M., Iula, A.: Recent advances in machine learning applied to ultrasound imaging. *Electronics* 11(11), 1800 (2022), <https://doi.org/10.3390/electronics11111800>
15. Muratov, M., Ryazanov, V., Biryukov, V., Petrov, D., Petrov, I.: Inverse problems of heterogeneous geological layers exploration seismology solution by methods of machine learning. *Lobachevskii Journal of Mathematics* 42(7), 1728–1737 (2021), <https://doi.org/10.1134/S1995080221070180>
16. Pleshkevich, A.L., Ivanov, A.V., Levchenko, V.D., *et al.*: Efficient parallel implementation of multi-arrival 3D prestack seismic depth migration. *Supercomputing Frontiers and Innovations* 6(1), 4–8 (2019), <https://doi.org/10.14529/jsfi190101>
17. Qi, Y., Chen, X., Zhao, Q., *et al.*: Seismic wave modeling of fluid-saturated fractured porous rock: including fluid pressure diffusion effects of discretely distributed large-scale fractures. *Solid Earth* 15(4), 535–554 (2024), <https://doi.org/10.5194/se-15-535-2024>
18. Rybakin, B., Goryachev, V.: Heterogeneous computing systems in problems of modeling filaments formation and pre-stellar objects. In: Voevodin, V., Sobolev, S., Yakobovskiy, M., Shagaliev, R. (eds.) *Supercomputing. RuSCDays 2022, Lecture Notes in Computer Science*, vol. 13708, pp. 127–139. Springer, Cham (2022), https://doi.org/10.1007/978-3-031-22941-1_9
19. Schoenberg, M.: Elastic wave behavior across linear slip interfaces. *The Journal of the Acoustical Society of America* 68(5), 1516–1521 (1980), <https://doi.org/10.1121/1.385077>

20. Vamaraju, J., Sen, M.K., De Basabe, J., Wheeler, M.: A hybrid Galerkin finite element method for seismic wave propagation in fractured media. *Geophysical Journal International* 221(2), 857–878 (2020), <https://doi.org/10.1093/gji/ggaa037>
21. Wang, K., Peng, S., Lu, Y., Cui, X.: The velocity-stress finite-difference method with a rotated staggered grid applied to seismic wave propagation in a fractured medium. *Geophysics* 85(2), T89–T100 (2020), <https://doi.org/10.1190/geo2019-0186.1>
22. Xiang, Y., Wang, Z., Song, Z., *et al.*: Seismictransformer: An attention-based deep learning method for the simulation of seismic wavefields. *Computers & Geosciences* p. 105629 (2024), <https://doi.org/10.1016/j.cageo.2024.105629>
23. Xu, J., Hu, H., Liu, Q.H., Zhan, Q., Zhuang, M.: Spectral element modeling of elastic wave propagation in an anisotropic background with discrete anisotropic fractures. *Geophysical Journal International* 227(2), 832–848 (2021), <https://doi.org/10.1093/gji/ggab226>
24. Xu, Y., Chen, X., Zhang, W., Pan, X.: An adaptive modal discontinuous Galerkin finite element parallel method using unsplit multi-axial perfectly matched layer for seismic wave modeling. *Commun. Comput. Phys.* 31(4), 1083–1113 (2022). <https://doi.org/10.4208/cicp.0A-2021-0118>

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*¹ , *Nikolay I. Khokhlov*^{1,2} , *Vladislav O. Stetsyuk*¹,
*Sergey D. Agibalov*¹

© 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-

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

²Scientific 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, σ_{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 λ and μ 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 δ_{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 ω_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 Λ 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.

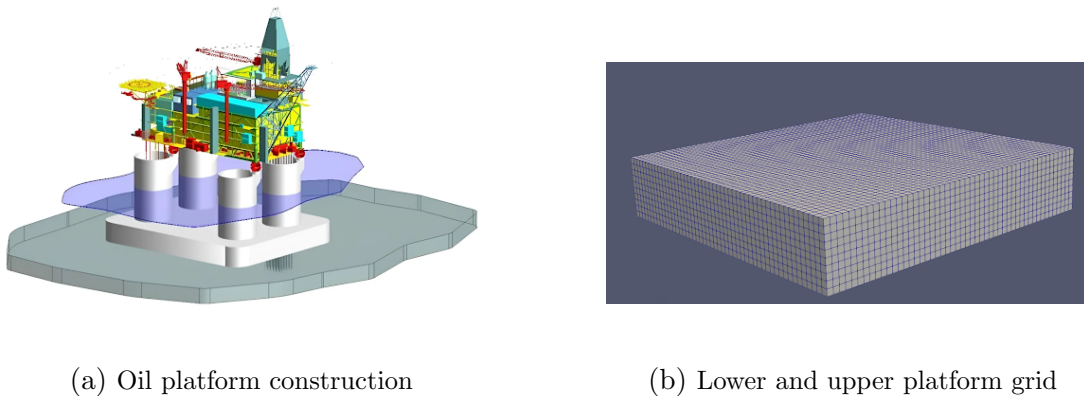


Figure 1. Construction of lower and upper platforms grids

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).

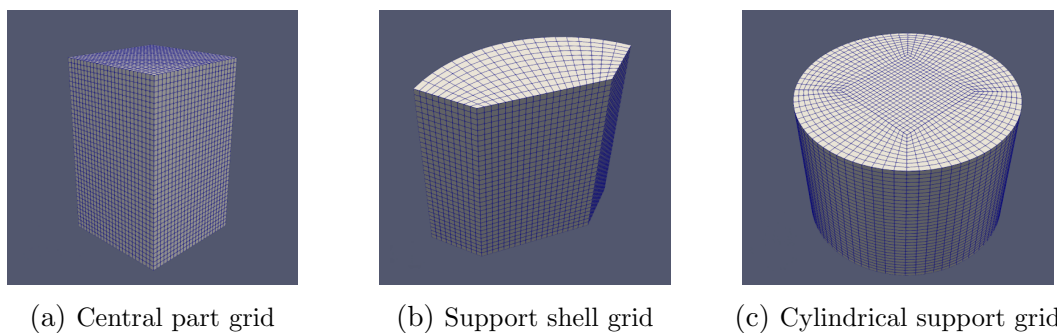


Figure 2. Construction of cylindrical supports grids

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.

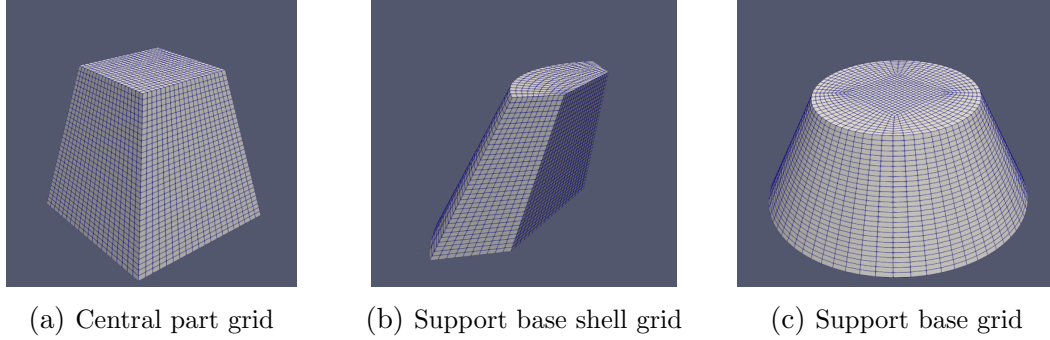


Figure 3. Construction of supports bases grids

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.

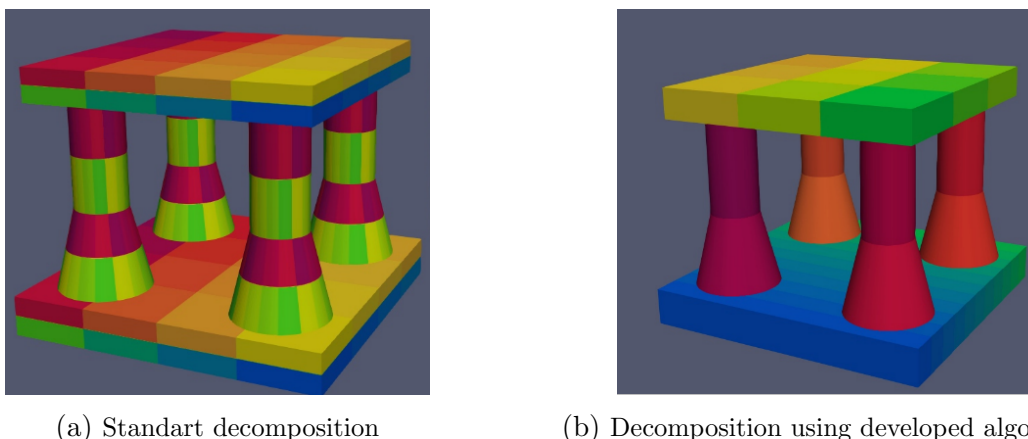


Figure 4. Decomposition of the computational domain. Color indicates the rank of the process

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.

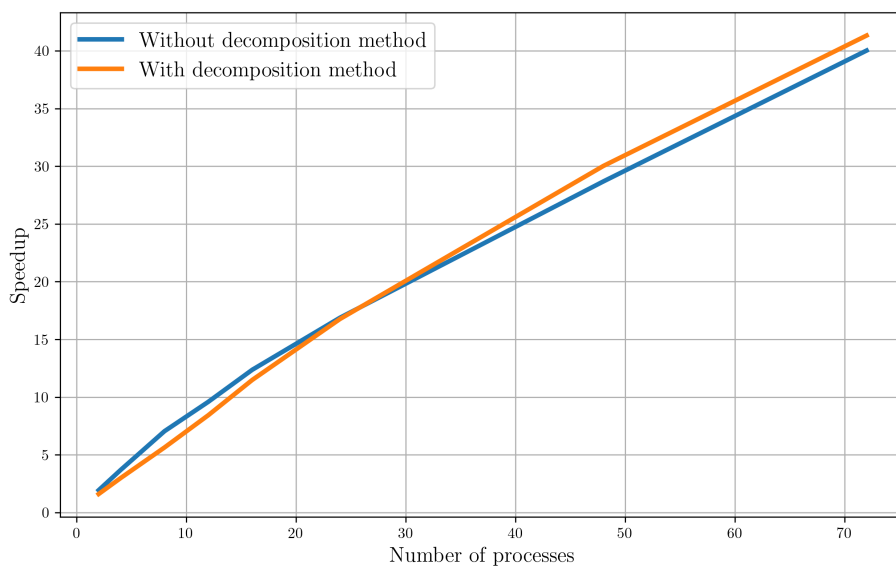


Figure 5. Graph of speedup dependence on the number of processes

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.

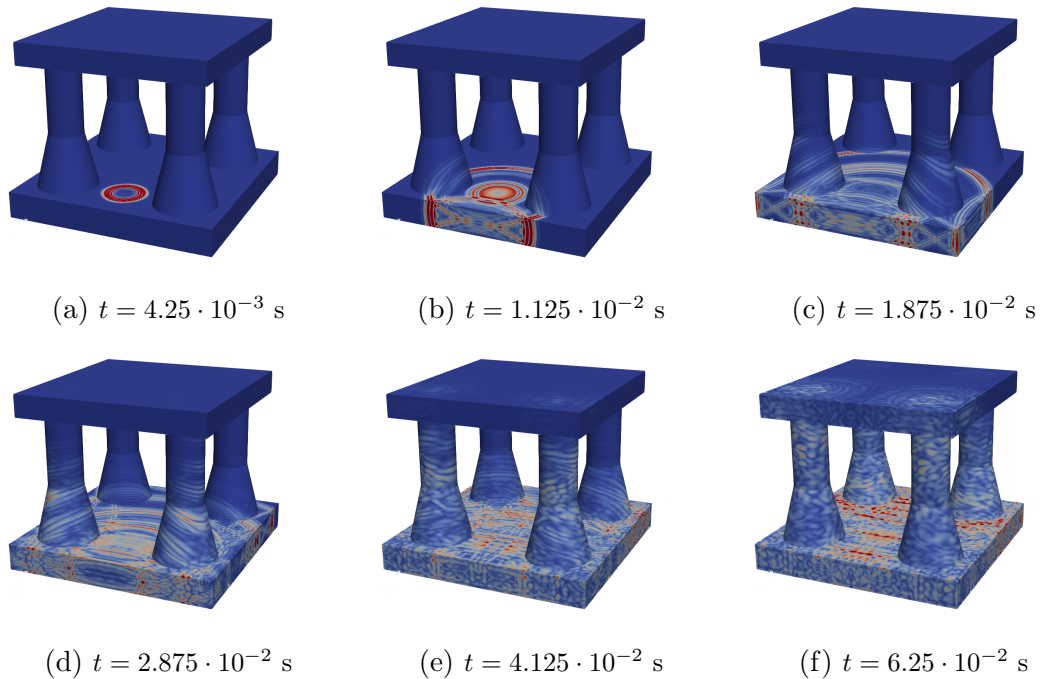


Figure 6. Wave disturbance propagation in the oil platform structure following an explosion at the platform base at various time points

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.

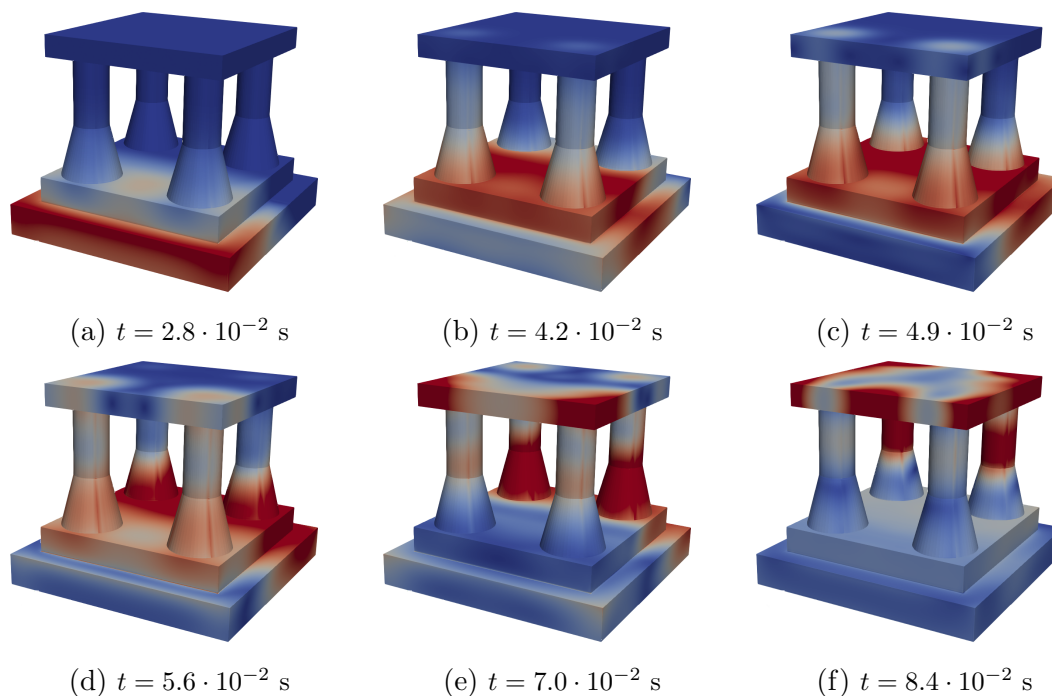


Figure 7. Wave disturbance propagation in the oil platform structure during an earthquake at various time points

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.

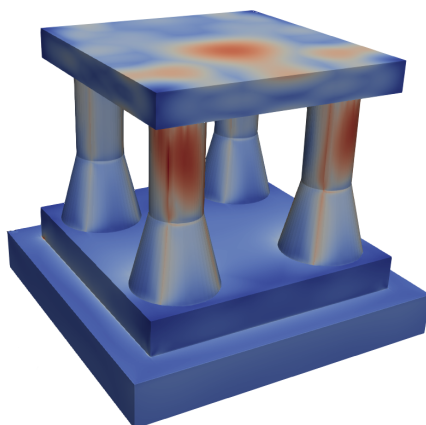


Figure 8. Investigation of stress distribution in the oil platform structure. The places of the highest intensity of stresses are highlighted by color

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
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
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
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
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>

Parallelization Strategies for Ultrasonic Wave Propagation in Composite Materials Considering Microstructural Details

Evgeniy Pesnya¹ , Alena V. Favorskaya^{1,2} , Igor B. Petrov^{1,2} ,
Nikolay I. Khokhlov^{1,2} 

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

This paper explores advanced parallelization strategies for simulating ultrasonic wave propagation in composite materials considering their complex microstructure. The grid-characteristic method and the use of Chimera grids in the simulations allow us to represent the composite material as an isotropic, linear-elastic medium and focus on improving the computational efficiency through efficient grid partitioning techniques. We used MPI (Message Passing Interface) technology on a high-performance computing cluster to test different methods for distributing computational grids across multiple processes. Our results highlight that partitioning grids according to material fiber layers improves the performance, especially when the number of processes matches the number of composite layers. This method not only provides better load balancing but also reduces communication overhead, making it the most efficient strategy tested. We present a comprehensive comparison of execution times and speedups for different partitioning approaches. Future work will aim to extend the study by increasing the number of layers and exploring how this approach scales with more complex and heterogeneous microstructures, potentially identifying further optimizations for parallel modeling.

Keywords: composite material, microstructure, grid characteristic method, parallelization.

Introduction

The accurate modeling of ultrasonic wave propagation in composite materials is essential for non-destructive evaluation (NDE) [4, 21], material testing, and structural health monitoring in industries such as aerospace [5, 12, 18], automotive [6, 22], and civil engineering [19, 23, 24]. Composite materials, due to their complex microstructures, require advanced computational approaches to capture the interactions between waves and internal structures, including fibers, voids, and other inclusions. These materials are typically heterogeneous, with varying elastic properties across different regions, making the wave propagation behavior significantly more complex compared to homogeneous materials.

Traditional numerical methods like the finite element method (FEM) and finite difference time domain (FDTD) are widely used in simulating ultrasonic waves in various materials [9, 10, 20]. However, when applied to composite materials, these methods often face limitations in terms of accuracy and computational efficiency, especially when high-frequency waves interact with fine microstructural features. The computational cost grows rapidly with the need to model small-scale features across large domains, particularly when simulating ultrasonic wave propagation at high frequencies or through intricate microstructures. Moreover, FEM and FDTD approaches often require complex meshing strategies, which can introduce errors and significantly increase computational overhead in handling boundary conditions between different material phases.

An alternative approach to simulating ultrasonic wave propagation in composite materials involves the use of anisotropic material models [2, 17, 25]. They are advantageous for representing the material's mechanical behavior with less need for complex meshing. However, their main drawback is the limited accuracy in handling intricate microstructural interactions, especially

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

²Scientific Research Institute for System Analysis of the Russian Academy of Sciences, Moscow, Russian Federation

when fibers are oriented differently. High-precision methods like FEM offer better resolution for wave phenomena at fine scales, such as reflection and diffraction at material interfaces. Consequently, anisotropic models are less effective for simulations requiring detailed accuracy in complex microstructures.

To address these challenges, more advanced numerical techniques have been proposed, including the use of grid-characteristic methods (GCM) [15, 16] and Chimera grid approaches [8, 14], which have demonstrated significant potential in improving both the accuracy and scalability of wave propagation simulations in complex media [1, 3]. The grid-characteristic method provides a more natural way to handle wavefront propagation across material interfaces, allowing for higher accuracy in capturing reflection, refraction, and diffraction phenomena. By integrating Chimera grids, which allow the computational domain to be divided into overlapping subdomains with independent grids, it becomes possible to model composite materials with complex microstructures in a more efficient and flexible manner [13, 14].

Despite these advances, large-scale simulations of wave propagation in composite materials remain computationally demanding. The high-resolution grids required to accurately model the material's microstructure, coupled with the need for long simulation times to capture wave interactions, make these problems difficult to solve on single-node machines. For instance, when modeling ultrasonic wave propagation at high frequencies in a composite material with multiple layers or complex fiber orientations, the memory and computational power required can far exceed the capabilities of even high-end workstations. This is particularly true for non-destructive evaluation (NDE) and structural health monitoring applications, where high precision is required to detect flaws or defects within the material.

Given these challenges, parallel computing has come to the forefront to achieve acceleration of large-scale simulations. In particular, the use of distributed memory parallelization techniques, such as those enabled by Message Passing Interface (MPI), has become critical for solving these computationally intensive problems [7]. In our problem, it is possible to divide the computational grid between several processors, which allows scaling the simulation on high-performance computing (HPC) clusters. By distributing the computational load across multiple nodes, each with its own memory and computing unit, MPI enables high-resolution simulation of ultrasonic waves in problems with huge scale and many complex structures, such as composite materials.

In this study, we explored several grid parallelization approaches to achieve optimization in multilayer composite modeling. We have two types of computational grids describing both the fibers and the matrix within the composite, which can be partitioned in different ways for parallel processing. One common approach is uniform grid partitioning, where the domain is divided into equally sized subdomains. This approach is generally easy to implement, but may not be as efficient when applied to composites with highly heterogeneous microstructures. For example, uniform partitioning may result in some processors processing regions with complex fiber arrangements, while others process simpler regions, leading to load imbalances and ultimately reducing overall efficiency.

One of the most effective strategies, especially for multilayer composites, seems to be grid partitioning based on the natural structure of the material. This approach partitions the grid according to the arrangement of fibers in each layer, and this allows for a more even distribution of the computational load. In addition to evenly distributing the computational work, it is also important to minimize communication with other threads. This is especially effective when the

number of processes is a multiple of the number of layers, as it avoids sharing fibers from the same layer between different processes and minimizes the overhead of inter-process communication.

The performance of the uniform, layered-based, and hybrid partitioning strategies was evaluated using an HPC cluster. Each node in the cluster was equipped with dual-socket Intel Xeon processors and 256 GB of RAM. Performance metrics including execution time, speedup, and efficiency were measured as the number of processes varied. The results showed that the layer-based grid partitioning outperformed the uniform approach, especially as the number of processes increased.

By aligning the grid partitioning with the composite material structure, the layer-based approach achieved better load balancing and higher computational efficiency. A speedup analysis that compares the execution time with multiple processes to the execution time with a single process showed that the layer-based method provided improvements, especially when the number of processes was a multiple of the number of layers in the material. Of further interest is how the efficiency depends on the number of composite layers and the area of the material being modeled.

The paper is structured as follows. The Methods section describes our methods for modeling ultrasonic wave propagation, including the grid-characteristic approach and MPI-based parallelization. The Results section outlines the experimental setup and analyzes the results of different grid partitioning strategies. The conclusion and future directions are presented in the Conclusion section.

1. Methods

In this section, we provide a more detailed description of the methods used to model ultrasonic wave propagation in composite materials. The main focus is on the integration of the grid-characteristic method (GCM), Chimera grids, and parallelization techniques using MPI.

1.1. Grid-Characteristic Method (GCM)

In our study, the governing equations for wave propagation in linear elastic materials, such as fiber materials, form the system:

$$\rho \dot{\mathbf{v}} = (\nabla \cdot \boldsymbol{\sigma})^T, \quad (1)$$

$$\dot{\boldsymbol{\sigma}} = \lambda (\nabla \cdot \mathbf{v}) \mathbf{I} + \mu (\nabla \otimes \mathbf{v} + (\nabla \otimes \mathbf{v})^T), \quad (2)$$

where \mathbf{v} is velocity vector, $\boldsymbol{\sigma}$ is Cauchy stress tensor and λ, μ are Lamé constants characterizing the properties of the material.

This is a system of first-order hyperbolic partial differential equations, which can be transformed to the following representation:

$$\frac{\partial \mathbf{u}}{\partial t} + A \frac{\partial \mathbf{u}}{\partial x} + B \frac{\partial \mathbf{u}}{\partial y} + C \frac{\partial \mathbf{u}}{\partial z} = 0, \quad (3)$$

where \mathbf{u} represents the vector of unknowns (displacement and stress), and $A, B,$ and C are coefficient matrices that depend on the material properties (e.g., density, elastic moduli).

GCM solves this system of equations by propagating information along the characteristic directions of the system using a structured grid that can be adapted to the geometry of the composite material. This approach is particularly effective in capturing wave reflections, transmissions, and interactions at interfaces between different materials within the composite.

In this work, free boundary conditions are applied at the outer boundaries of the computational domain. In addition, a no-slip condition is used to describe the interaction between the fibers and the surrounding matrix, simulating the physical adhesion between these components. These boundary conditions are conveniently taken into account by GCM, since the method handles such conditions naturally without the need for complex meshing or computational costs, preserving accuracy and simplifying the simulation setup.

1.2. Structured Curvilinear Grids

In fiber-reinforced composites, the fibers often have complex curved geometries. To accurately model the geometry and capture fine details of wave interactions, we use structured curvilinear grids. These grids allow us to represent the fibers and surrounding matrix with high accuracy without requiring excessive grid refinement in regions away from the interfaces.

To create curved grids, we map the physical domain of the composite material onto a reference domain using a transformation that preserves the geometric features of the fibers and matrix. This transformation is applied to the system of governing equations, yielding a modified set of equations that account for the curved nature of the grid. The coefficients of the transformed system are dynamically updated based on the Jacobian of the mapping at each grid point, as described in [8].

1.3. Chimera Grid Integration

One of the key innovations of our method is the integration of Chimera grids into the GCM framework. Chimera grids are overlapping grids that allow different regions of the composite material to be modeled independently at different grid resolutions. This is particularly useful when dealing with the heterogeneous nature of composite materials, where certain regions (e.g., near fibers) require finer grids, while other regions (e.g., in the matrix) can be modeled with coarser grids.

In our simulations, the background grid represents the global structure of the composite material, while individual Chimera grids are used to model individual fibers or other microstructural features at higher resolution. The Chimera grids are overlapped by the background grid, and the solution is interpolated between the grids at each time step.

Figure 1 illustrates the region of interest in the laminated composite material, where the computational grids were constructed. The background grid is shown in blue and the fibers are depicted in gray. The corresponding material parameters for this region are detailed in Tab. 1.

Table 1. Elasticity parameters of materials

Material	$c_p, m/s$	$c_s, m/s$	$\rho, kg/m^3$
Fiber	4003	3004	2000
Matrix	2553	1194	1160

Using Chimera grids allows us to focus computational resources on regions of interest, such as interfaces between materials, while maintaining the overall accuracy of the simulation.

Additionally, the independent nature of the Chimera grids lends itself well to parallelization, as the computations in each grid can be performed independently and then the results are interpolated back onto the background grid.

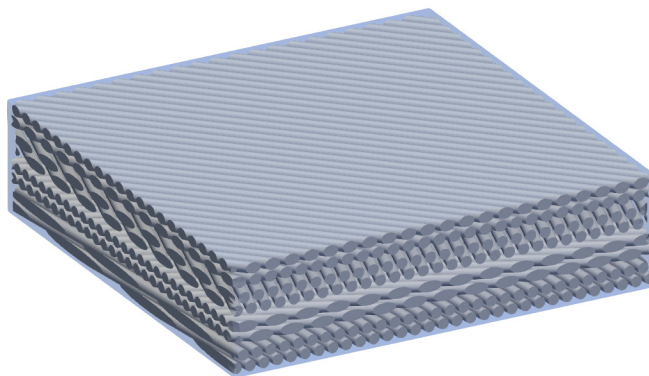


Figure 1. Representation of the structure of a composite material in the problem domain

1.4. Parallelization Using MPI

Given the complexity of the problem and the need for high-resolution simulations, parallelization is essential for achieving reasonable computation times. In our study, we employed MPI (Message Passing Interface) for the simulation process parallelization across multiple processors in a distributed memory high-performance computing (HPC) cluster.

1.4.1. Parallelization strategy

Several parallelization strategies have been considered as a means of dividing the computational domain among several processes. These strategies include:

1. Uniform Grid Division (Fig. 2): In this approach, the computational grids are divided into uniform subdomains, with each subdomain assigned to a different process. This method is simple to implement but can lead to load imbalances, particularly in regions with complex microstructure, where some processes may have significantly more work than others.

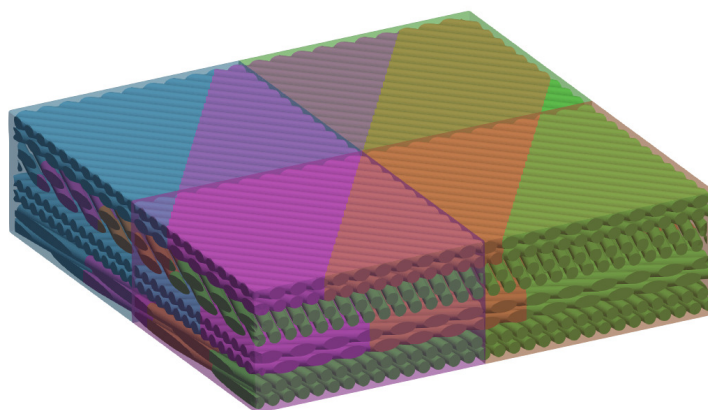


Figure 2. Uniform grid division (four colors correspond to four different processes)

2. Layer-Based Division (Fig. 3): Given the layered structure of many composite materials, we explored a parallelization strategy that divides the computational domain by layers. Each process is responsible for simulating one or more layers of the composite, with the number

of layers assigned to each process determined based on the total number of processes. This approach is particularly effective when the number of processes is a multiple of the number of layers, as it ensures a balanced distribution of the workload. The layer-based division strategy proved to be the most successful for our simulations. By aligning the computational grid division with the physical structure of the material (i.e., the location of fibers and matrix layers), we were able to achieve better load balancing and minimize communication overhead between processes.

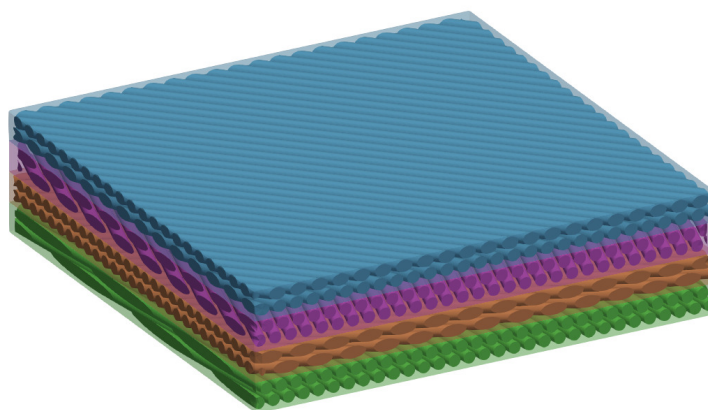


Figure 3. Layer-based division (four colors correspond to four different processes)

3. Hybrid Approach (Fig. 4): The idea here was to combine uniform grid partitioning and layer-based partitioning to achieve more flexible parallelization. For example, within each layer, the domain could be divided into smaller subdomains, which are then assigned to individual processes. This hybrid approach allowed us to gain a broader understanding of how each approach impacts the overall performance.

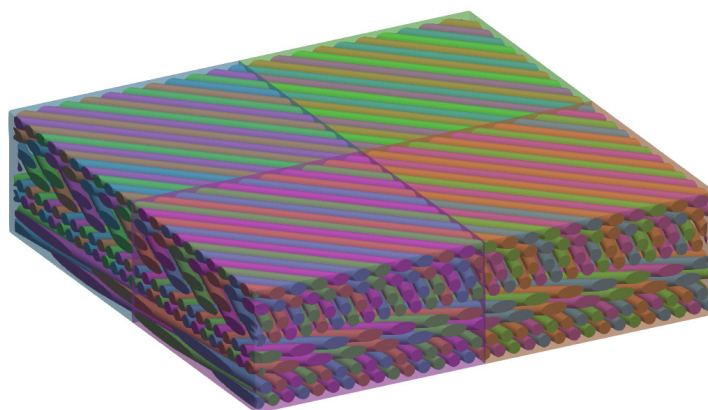


Figure 4. Hybrid approach division (four colors correspond to four different processes)

1.4.2. Communication and synchronization

MPI-based parallelization involves communication between processes to exchange information about the solution at the boundaries of each subdomain. In case of uniform grid division,

this communication occurs at the boundaries of each subdomain, where processes must exchange information about the displacement and stress fields. In the case of layer-based division, communication occurs between the processes responsible for adjacent layers.

To minimize the communication overhead, we implemented a non-blocking communication scheme using MPI's `MPI_Isend` and `MPI_Irecv` functions. This allows processes to continue computations while waiting for communication to complete, thereby reducing idle time and improving overall efficiency.

Synchronization between processes is required at each time step to ensure that the solution is consistent across the entire domain. This is particularly important when using Chimera grids, as the solution on each grid must be interpolated onto the background grid before proceeding to the next time step.

1.4.3. Performance metrics

To evaluate the performance of our parallelization strategies, we measured the execution time, speedup and efficiency for various configurations of processes and grid divisions. The speedup is defined as the ratio of the execution time on a single process to the execution time on multiple processes:

$$Speedup = \frac{T_1}{T_p},$$

where T_1 is the execution time on a single process, and T_p is the execution time on p processes. The efficiency is defined as follows:

$$Efficiency = \frac{Real\ Speedup}{Linear\ Speedup}.$$

Our results, discussed in detail in the Results section, show that the layer-based division strategy provided the best performance, particularly when the number of processes was a multiple of the number of layers. This strategy also scaled well with increasing numbers of processes, achieving near-linear speedup in many cases.

2. Results

As a test case, we considered the problem of an elastic wave propagating through the cross-section of a composite material. A more detailed problem setup can be found in [13]. Solving this problem required approximately 50 GB of RAM due to the large number of computational grids, making the use of a cluster necessary. Figure 5 represents the speed field over the part of vertical cross-section of the composite material.

A comparison of execution time, speedup, and efficiency for different parallelization approaches with varying numbers of processes revealed similar results across methods (Fig. 6). However, the layer-based grid partitioning approach demonstrated better efficiency, particularly as the number of processes increased. Previous works on parallelizing composite grids for other types of problems showed less favorable results for the layer-based approach [11]. In contrast, the specific characteristics of our problem allowed us to achieve a significant performance improvement using this method.

It is important to note that parallelization in this context is particularly challenging, as the fibers, each assigned separate Chimera computational grids, are oriented in different directions.

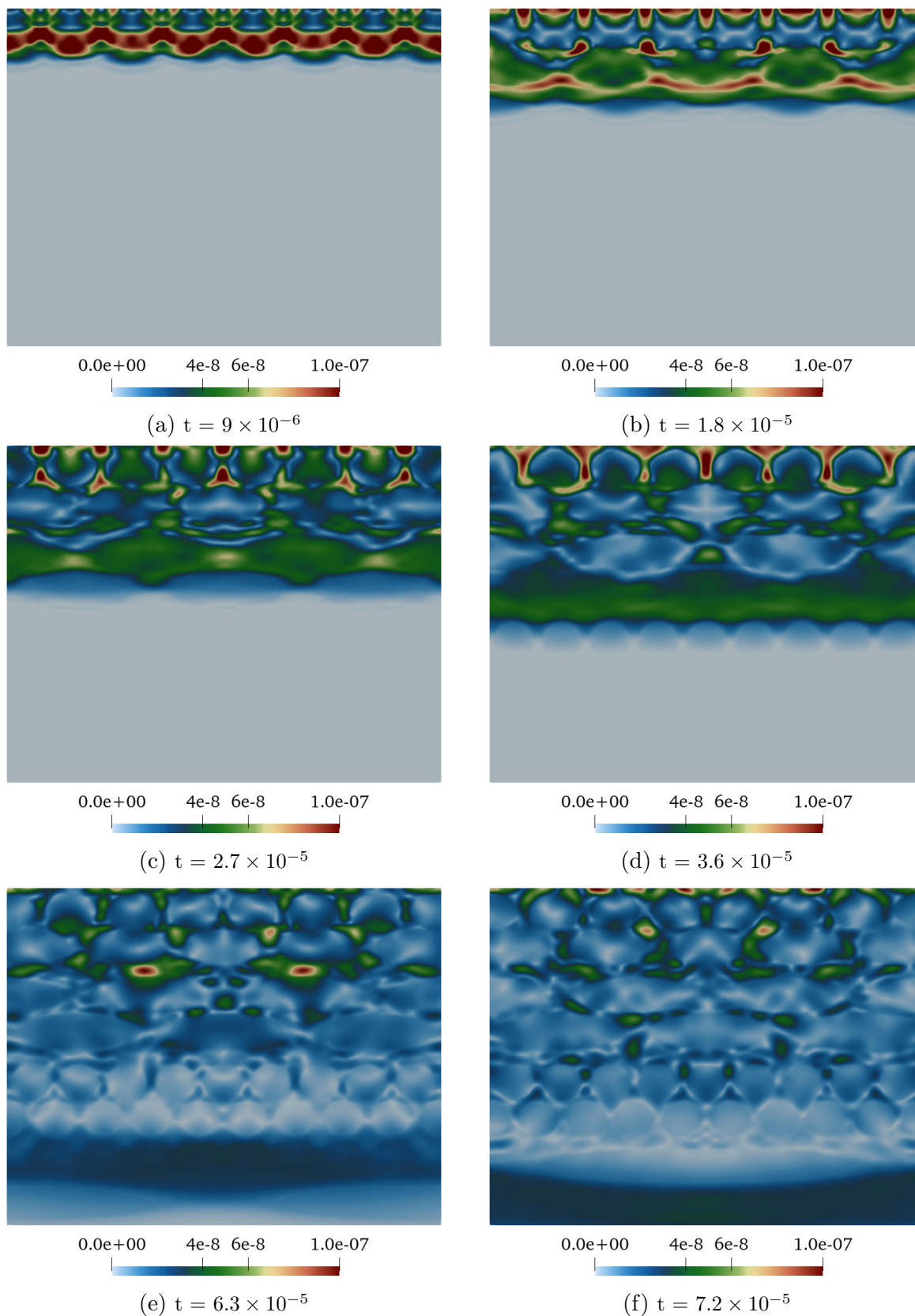


Figure 5. Field $\|\mathbf{v}\|$ in the vertical section of composite at different moments of time t

This orientation complexity makes it difficult to group closely related grid elements into a single process, further complicating the parallelization strategy.

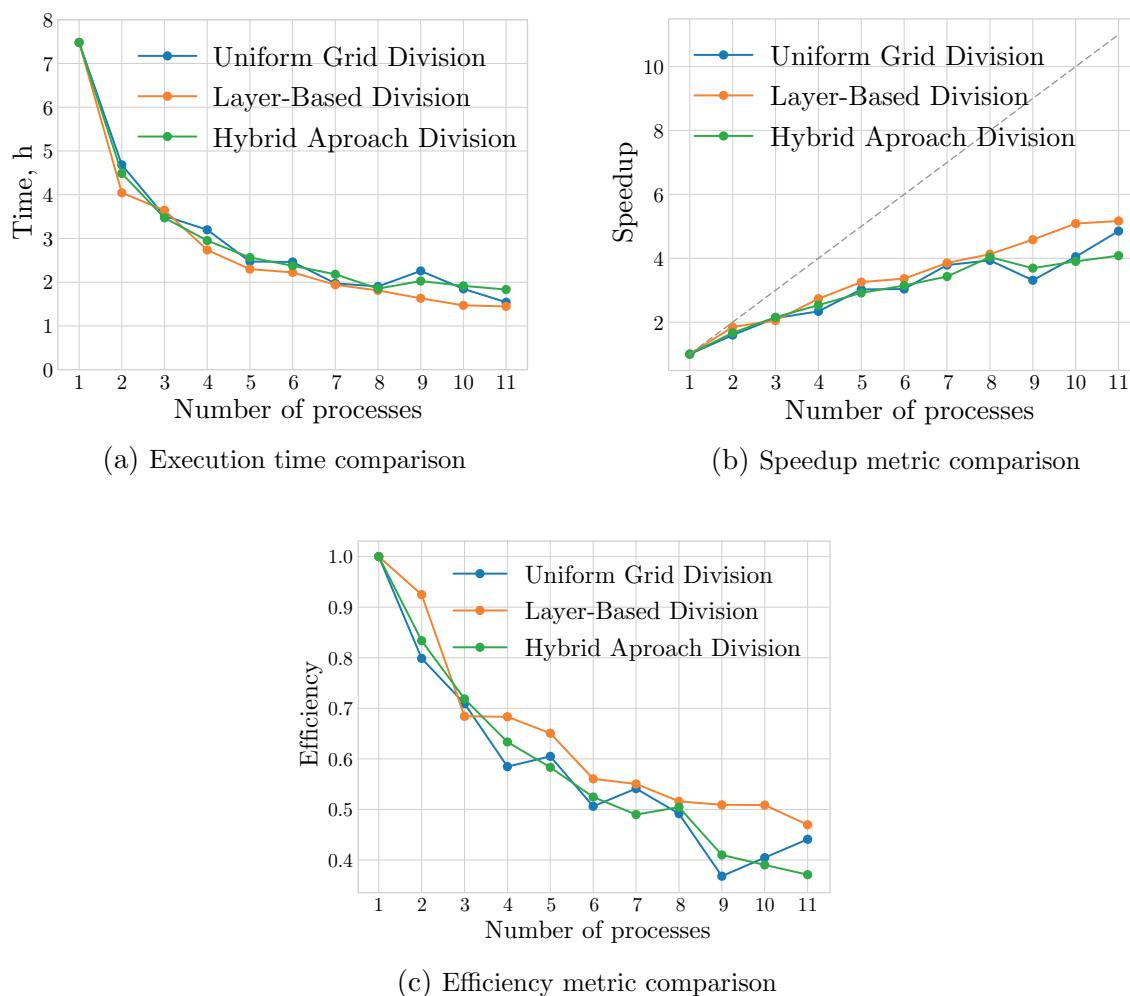


Figure 6. Metrics for different approaches depending on the number of processes

Conclusion

This study presents a novel parallelization approach for simulating ultrasonic wave propagation in composite materials, specifically focusing on optimizing grid partitioning based on the materials microstructure. The layer-based division strategy, when aligned with the fiber layout of the composite, provides better performance, especially in cases where the number of processes matches the number of layers. Future work will focus on scaling these methods for more complex composites and exploring additional parallelization strategies.

By integrating the grid-characteristic method with MPI-based parallelization, we have laid the foundation for more efficient and accurate simulations, enabling better predictive capabilities for non-destructive testing and structural health monitoring in composite materials.

Acknowledgements

The research was supported by the Russian Science Foundation grant No. 23-11-00035, <https://rscf.ru/project/23-11-00035/>. 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. Beklemysheva, K., Golubev, V., Petrov, I., Vasyukov, A.: Determining effects of impact loading on residual strength of fiber-metal laminates with grid-characteristic numerical method. *Chinese Journal of Aeronautics* 34(7), 1–12 (2021). <https://doi.org/10.1016/j.cja.2020.07.013>
2. Car, E., Zalamea, F., Oller, S., *et al.*: Numerical simulation of fiber reinforced composite materials-two procedures. *International Journal of Solids and Structures* 39(7), 1967–1986 (2002). [https://doi.org/10.1016/S0020-7683\(01\)00240-2](https://doi.org/10.1016/S0020-7683(01)00240-2)
3. Favorskaya, A., Petrov, I.: Calculation of the destruction of ice structures by the grid-characteristic method on structured grids. *Procedia Computer Science* 192, 3768–3776 (2021). <https://doi.org/10.1016/j.procs.2021.09.151>
4. Gupta, R., Mitchell, D., Blanche, J., *et al.*: A review of sensing technologies for non-destructive evaluation of structural composite materials. *Journal of Composites Science* 5(12) (2021). <https://doi.org/10.3390/jcs5120319>
5. Kastratovic, G., Grbovic, A., Sedmak, A., *et al.*: Composite material selection for aircraft structures based on experimental and numerical evaluation of mechanical properties. *Procedia Structural Integrity* 31, 127–133 (06 2021). <https://doi.org/10.1016/j.prostr.2021.03.021>
6. Khan, F., Hossain, N., Mim, J.J., *et al.*: Advances of composite materials in automobile applications - A review. *Journal of Engineering Research* (2024). <https://doi.org/10.1016/j.jer.2024.02.017>
7. Khokhlov, N.: Application of the grid-characteristic method for solving the problems of the propagation of dynamic waves using HPC systems. *Computational Mathematics and Information Technologies* 1, 20–30 (07 2022). <https://doi.org/10.23947/2587-8999-2022-1-1-20-30>
8. Khokhlov, N.I., Favorskaya, A., Furgailo, V.: Grid-characteristic method on overlapping curvilinear meshes for modeling elastic waves scattering on geological fractures. *Minerals* 12(12) (2022). <https://doi.org/10.3390/min12121597>
9. Lozano, D., Bulling, J., Asokkumar, A., *et al.*: 3D simulations of ultrasonic waves in plates using the scaled boundary finite element method and high-order transition elements. *Wave Motion* 120, 103158 (2023). <https://doi.org/10.1016/j.wavemoti.2023.103158>
10. Marković, N., Stojić, D., Cvetkovic, R., *et al.*: Numerical modeling of ultrasonic wave propagation - by using of explicit fem in abaqus. *Facta universitatis. Series: Architecture and Civil Engineering* 16, 135–147 (01 2018). <https://doi.org/10.2298/FUACE170830011M>

11. Mitskovets, I., Sagan, V., Khokhlov, N.: Parallel modeling of elastic wave propagation, with explicit pore delineation using overset grids method. *Physics of Particles and Nuclei* 55(3), 516–518 (Jun 2024). <https://doi.org/10.1134/S1063779624030602>
12. Parveez, B., Kittur, M.I., Badruddin, I.A., *et al.*: Scientific advancements in composite materials for aircraft applications: A review. *Polymers* 14(22) (2022). <https://doi.org/10.3390/polym14225007>
13. Pesnya, E., Petrov, I.B.: Dynamic processes calculation in composites considering internal structure using the Grid-Characteristic method. *Lobachevskii Journal of Mathematics* 45(1), 319–327 (Jan 2024). <https://doi.org/10.1134/S1995080224010438>
14. Pesnya, E., Favorskaya, A., Khokhlov, N.: Grid-characteristic method on Chimera meshes using a single background grid with varying elastic properties. *Procedia Computer Science* 207, 1398–1407 (2022). <https://doi.org/10.1016/j.procs.2022.09.196>
15. Petrov, I.B.: Application of the grid-characteristic method for mathematical modeling in dynamical problems of deformable solid mechanics. *Comput. Res. Model.* 11(6), 1041–1048 (Dec 2019). <https://doi.org/10.20537/2076-7633-2019-11-6-1041-1048>
16. Petrov, I.: Grid-characteristic methods. 55 years of developing and solving complex dynamic problems. *Computational Mathematics and Information Technologies* 6, 6–21 (03 2023). <https://doi.org/10.23947/2587-8999-2023-6-1-6-21>
17. Petrov, I., Favorskaya, A., Vasyukov, A., *et al.*: Numerical simulation of wave propagation in anisotropic media. *Doklady Mathematics* 90, 778–780 (11 2015). <https://doi.org/10.1134/S1064562414070163>
18. Rahul, V., Alokita, S., Jayakrishna, K., *et al.*: Structural health monitoring of aerospace composites. In: Jawaid, M., Thariq, M., Saba, N. (eds.) *Structural Health Monitoring of Biocomposites, Fibre-Reinforced Composites and Hybrid Composites*, pp. 33–52. Woodhead Publishing Series in Composites Science and Engineering, Woodhead Publishing (2019). <https://doi.org/10.1016/B978-0-08-102291-7.00003-4>
19. Revel, G., Pandarese, G., Cavuto, A.: Advanced ultrasonic non-destructive testing for damage detection on thick and curved composite elements for constructions. *Journal of Sandwich Structures and Materials* 15, 5–24 (01 2013). <https://doi.org/10.1177/1099636212456861>
20. Saeki, M., Bustamante, L., Misaki, T., *et al.*: FDTD simulation study of ultrasonic wave propagation in human radius model generated from 3D HR-pQCT images. *Physics in Medicine* 10, 100029 (2020). <https://doi.org/10.1016/j.phmed.2020.100029>
21. Segovia Ramrez, I., Garca Mrquez, F.P., Papaelias, M.: Review on additive manufacturing and non-destructive testing. *Journal of Manufacturing Systems* 66, 260–286 (2023). <https://doi.org/10.1016/j.jmsy.2022.12.005>
22. Wazeer, A., Das, A., Abeykoon, C., *et al.*: Composites for electric vehicles and automotive sector: A review. *Green Energy and Intelligent Transportation* 2(1), 100043 (2023). <https://doi.org/10.1016/j.geits.2022.100043>

23. Wong, K.Y., Ni, Y.Q.: Structural health monitoring of cable-supported bridges in Hong Kong. In: Karbhari, V.M., Ansari, F. (eds.) Structural Health Monitoring of Civil Infrastructure Systems, pp. 371–411. Woodhead Publishing Series in Civil and Structural Engineering, Woodhead Publishing (2009). <https://doi.org/10.1533/9781845696825.2.371>
24. Xu, X., Ran, B., Jiang, N., *et al.*: A systematic review of ultrasonic techniques for defects detection in construction and building materials. *Measurement* 226, 114181 (2024). <https://doi.org/10.1016/j.measurement.2024.114181>
25. Yin, B., Zreid, I., Lin, G., *et al.*: An anisotropic damage formulation for composite materials based on a gradient-enhanced approach: Formulation and implementation at small strain. *International Journal of Solids and Structures* 202, 631–645 (2020). <https://doi.org/10.1016/j.ijsolstr.2020.06.035>

Leveraging OpenMP Tasks for Efficient Parallel Modeling of the Elastic Eave Propagation in Multi-mesh Problems

Nikolay I. Khokhlov^{1,2} , Vladislav O. Stetsyuk¹

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

This paper presents a new algorithm for parallelizing the grid-characteristic method in shared-memory systems. The OpenMP task parallelism mechanism is used for parallelization. A modification of the grid-characteristic method is considered that uses a set of overlapped grids to determine a complex heterogeneous structure of the computational domain. The complexity of parallelizing the algorithm is represented by the presence of many different-sized grids. The proposed algorithm is described and compared with basic parallelization algorithms. Basic algorithms mean separate parallelization within each computational grid using the loop parallelization mechanism. An analysis of the efficiency of the post-doubling and parallel algorithms is performed. The advantage of the proposed algorithm for a number of problems is demonstrated. The results of testing and calculating the propagation of wave disturbances in a fractured layer are presented. Each crack in the example is specified by a separate computational grid, which significantly increases the multi-scale problem and the number of computational grids. Work is underway to transfer the algorithm to the three-dimensional case.

Keywords: grid-characteristic method, OpenMP, task based parallelism, overset meshes, geological fractures.

Introduction

Modeling of elastic and acoustic processes in two-dimensional and three-dimensional media is a frequently encountered problem. Thus, questions of propagation of dynamic wave disturbances arise in a wide range of problems of mathematical physics. These include problems of seismic exploration, geophysics, non-destructive testing, ultrasound and others. For the numerical solution of this kind of problem, a sufficiently large number of approaches already exist. The best known approaches include finite difference method, finite element method and spectral method. They all have their strengths and weaknesses, as well as the classes of problems they are best suited for. There are methods on unstructured and structured grids. For unstructured grids, finite element methods, spectral methods and discontinuous Galerkin method [13] are more typical. Finite difference methods are more common on structured grids [11, 25]. In some cases, the use of one or another approach is preferable. The works [5, 18] show that for solving seismic problems on sufficiently large computational grids, it is preferable to use structured grids. The method we use is called the grid-characteristic method [9] on the rectangular grids. It is well suited for computer simulations and is characterized by the simplicity of setting the area of integration and the ability to work with multiple meshes (chimera grid method) [23]. This method is also widely used to solve various problems of mathematical physics [15, 16] and in some cases has advantages over other calculation methods [5].

When solving numerical problems of dynamic disturbances in heterogeneous media, there is a need to use sufficiently large computational grids. Using such grids leads to significant time costs for performing calculations. To ensure acceptable time for solving the problem, it is necessary to use modern parallelization technologies. For modeling we use a software package developed by us that supports parallel execution in shared memory systems and distributed

¹Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

²Scientific Research Institute for System Analysis of the National Research Centre “Kurchatov Institute”, Moscow, Russian Federation

clusters [12]. This software package is parallelized on systems with distributed memory using MPI technology [14, 22]. On systems with shared memory, OpenMP technology is used [10]. Previously, when using this technology, parallelization was implemented using mechanisms for parallelizing cycles of the for type. The purpose of this paper is to investigate the possibility of using the task model that appeared in new versions of the OpenMP specification and to compare the efficiency of the implementation using this model with the previously performed implementation using the functionality of older OpenMP standards.

The work is organized as follows. Section 1 describes the numerical methods and mathematical model. Section 2 describes the implementation of the numerical algorithm and parallelization. Section 3 presents the results of testing the parallel algorithm. Practical Example presents example of calculating a seismic model with fractures. Conclusion summarizes the study and points directions for further work.

1. Computational Method

1.1. Elastic Wave Model

Elastic wave propagation is described using the Cauchy-Green tensor [24]

$$\epsilon_{i,j} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \sum_l \frac{\partial u_i}{\partial x_l} \frac{\partial u_j}{\partial x_l} \right). \quad (1)$$

We are assuming all displacements to be small, therefore second derivatives can be ignored. Newton's second law and Hooke's law take the following forms

$$\rho \frac{\partial^2 u_i}{\partial t^2} - \sum_j \frac{\partial \sigma_{ij}}{\partial x_i} - f_i = 0, \quad (2)$$

$$\sigma_{ij} = \sum_{k=1}^3 \sum_{l=1}^3 C_{ijkl} \epsilon_{kl}. \quad (3)$$

Tensors ϵ and σ are symmetric, so C can be transformed to a matrix using Voigt notation [26]. Moreover, in isotropic media this matrix can be fully defined using only two parameters: λ and μ , known as Lamé parameters

$$C_{\alpha\beta} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}. \quad (4)$$

Defining equation system can be simplified to the following

$$\begin{aligned} \rho \frac{\partial \vec{v}}{\partial t} &= (\nabla \cdot \sigma)^\top + \vec{f}, \\ \frac{\partial \sigma}{\partial t} &= \lambda(\nabla \cdot \vec{v})\mathbf{I} + \mu(\nabla \otimes \vec{v} + (\nabla \otimes \vec{v})^\top). \end{aligned} \quad (5)$$

1.2. Grid-characteristic Method

Equation system (5) can be written as single matrix equation with variable vector [9]

$$\mathbf{q} = \left[v_1 \quad v_2 \quad v_3 \quad \sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{23} \quad \sigma_{13} \quad \sigma_{12} \right]^\top. \quad (6)$$

After grouping derivatives along coordinates, it takes the following form

$$\frac{\partial}{\partial t} \mathbf{q} - \mathbf{A}_1 \frac{\partial}{\partial x_1} \mathbf{q} - \mathbf{A}_2 \frac{\partial}{\partial x_2} \mathbf{q} - \mathbf{A}_3 \frac{\partial}{\partial x_3} \mathbf{q} = 0. \quad (7)$$

We can split this equation into 3, i.e. 1 along each axis [17]

$$\frac{\partial}{\partial t} \mathbf{q} = \mathbf{A}_i \frac{\partial}{\partial x_i} \mathbf{q}. \quad (8)$$

Original grid-characteristic method [21] is based on solving the equation system using the transfer along the characteristics, but we are using the modification, which is more suitable for computer modelling. Since the original equation system is hyperbolic, all matrices \mathbf{A}_i have a full set of eigenvalues and eigenvectors. Therefore, they can be diagonalized: $\mathbf{A}_i = \mathbf{\Omega}_i^{-1} \mathbf{\Lambda}_i \mathbf{\Omega}_i$. We can replace the variable again: $\omega_i = \mathbf{\Omega}_i \mathbf{q}$ and the equation takes the following form

$$\frac{\partial}{\partial t} \omega_i + \mathbf{\Lambda}_i \omega_i = 0. \quad (9)$$

Matrix $\mathbf{\Lambda}_i$ is a diagonal matrix of eigenvalues, so the matrix equation can be split into independent equations for each component of ω . Moreover, each of those equations is an advection equation, which can be easily solved using finite-difference schemes.

1.3. Boundary Conditions

Applying the procedure described above to the nodes lying near the boundary of the integration area involves some additional steps. As it was mentioned, grid-characteristic method is based on transferring the values along the characteristics. For boundary nodes some of the characteristics are outgoing (i.e., the values we need to transfer are coming from outside the grid). To do a simulation step in these nodes, we need to define the boundary conditions and calculate the values using them.

We model only linear boundary conditions. They have the following form [7]

$$B\mathbf{q}(t + \tau) = b. \quad (10)$$

Now we can split the replacement into two summands, corresponding to the inner and outer characteristics

$$\begin{aligned} \vec{q}(t + \tau, \vec{x}) &= \mathbf{\Omega}^{int} \vec{\omega}^{int}(t + \tau, \vec{x}) + \mathbf{\Omega}^{(*)out} \vec{\omega}^{out}(t + \tau, \vec{x}) = \\ &= \vec{q}^{int}(t + \tau, \vec{x}) + \mathbf{\Omega}^{(*)out} \vec{\omega}(t + \tau, \vec{x}). \end{aligned} \quad (11)$$

Expressing $\vec{\omega}^{out}(t + \tau, \vec{x})$ from the boundary condition formula gives the following form of the equation

$$\vec{q}(t + \tau, \vec{x}) = \vec{q}^{int}(t + \tau, \vec{x}) + \mathbf{\Omega}^{(*)out} \left(B\mathbf{\Omega}^{(*)out} \right)^{-1} (b - B\vec{q}^{int}(t + \tau, \vec{x})). \quad (12)$$

During the modeling the summand corresponding to the inner characteristics is calculated from the known node values and the summand corresponding to the outer characteristics is

calculated and taken into account later during the value correction. It is made possible by the fact that we do not use ω values in the boundary conditions, all the operations are done using the “real” values (\mathbf{q}).

Some boundary conditions and the process of their modeling is described with more details in [8, 9]. It is also worth mentioning, that modeling an absorbing boundary is complicated. Modeling it with linear boundary conditions is possible, but the boundary has a noticeable reflection. A better way to model it is to use a Perfectly Matched Layer (PML) as described in [2, 20, 27].

1.4. Chimera Grids

Grid-characteristic method can be used for simulation using both structured and unstructured grids. However, simulations in unstructured grids require significantly more computational resources. Moreover, often the simulation area is mostly homogeneous and only contains a few subareas, where inhomogenities are located. Accurate modelling in these subareas requires covering them with grids that have a small spatial step, but covering the whole integration area with a grid with a step this small is a waste of computational resources. Methods, that use unstructured meshes, like finite elements method [19, 28], solve this by simply making a grid more detailed where it is needed, but we are working with rectangular and preferably structural grids, so cannot follow this path.

The method we are using is known as a chimera grid method [3, 4]. It is based on using multiple grids: a coarse grid, known as “main”, that covers the whole area and a set of fine grids, that cover the areas where we need to do the modeling more accurately. The main grid is structured, while additional grids (we call them overset grids) can be structured or unstructured. We are using interpolation to transfer the values between grids.

It is worth mentioning that we are not limited to just one singular main grid, it can be decomposed into blocks or we can have several main grids and transfer the values between them. Moreover, overset grids can be nested, and this is useful in situations like modeling a crack cluster.

2. Implementation

2.1. Solver Design

The modeling software package was designed to support both elastic and acoustic processes, multiple finite-difference schemes and flexible configuration without rebuilding. The class, responsible for modeling iterations is the `Solver` class. This class owns a set of `GridContainer` class instances, which represent grids, and a set of `GridCorrector` instances for grid contacts. Each `GridContainer` holds everything needed to make a simulation step in a grid: a `Grid`, a `Schema` and 8 sets of `GridCorrectors` (1 set of correctors and 1 set of fillers for each coordinate axis, a set of fillers and a set of correctors that are applied during a step over all axes).

`Grid` class is responsible for storing data, related to grid nodes or the whole grid. It supports both structured and unstructured curvilinear rectangular grids. Node data is stored in ZYX order (X is the fast axis) and uses array-of-structs layout.

`Schema` class is responsible for doing a simulation step on the nodes of a grid. It is a template class, parametrized by `Transformer` and `Reconstructor`. During the simulation step,

`Transformer` class is used to convert node values to ω . After that `Reconstructor` is used to calculate new ω values, and finally `Transformer` is used to calculate new node values from them. This split allows us to reuse finite-difference schemes, implemented as `Reconstructors` for modeling other physical processes.

Our implementation distinguishes between three types of `GridCorrectors`: fillers, correctors and contacts. Fillers are used to fill ghost nodes – nodes that lay near the edges of a grid chunk. They are mainly used to transfer values between parts of a grid in multi node cluster environments. Correctors are manipulating values in grid nodes and are used to implement boundary conditions, material property changes, destruction and inhomogeneities. Contacts are affecting multiple grids at once. Their use cases include interpolation in chimera grid approach and contacts between elastic and acoustic media.

2.2. Parallel Implementation

For parallel execution in shared memory we are using OpenMP [6].

As described above, we are doing simulation steps in turns over different axes, and a step for each axis is split into several subtasks: applying predictor and corrector contacts, applying boundary conditions and fillers, doing the step in grids.

Predictor and corrector contact application is similar, and the only difference is when it is done (before or after the grid step). Correctors are different from everything else, because they impact multiple grids at once. To avoid data races and ensure the correct results, we are enforcing the order of corrector applications. This is done using OpenMP task dependencies. Each corrector creates a task and declares that this task depends on the grids it is using in some way. For example, interpolation from one grid to the other declares that the first grid is an input dependency and the second is an output dependency. This allows OpenMP runtime to construct a task dependency graph, preventing multiple contacts from trying to write into the same grid at the same time or using the data that should be but is not yet updated by some contact as an input for the other contact. These constraints are actually very strict and can be relaxed a bit in some cases, for example if the node values near the left border of the grid are interpolated, values in nodes near the right border can be used in some other corrector. On the other hand, this dependency planning upgrade is hard to implement, because to do it, we need some kind of decomposer, like the one used for MPI. From our observations, applying grid contacts accounts for too small a percentage of the total simulation time for this feature to be a high priority, that is why we have not implemented it yet.

Boundary conditions are applied independently for each grid and are generally only taking a little time, therefore we are generating 1 task per grid to apply all boundary conditions to it. If in future boundary condition parallelism disbalance starts being an issue, we can switch to generating separate tasks for each boundary condition, but for now it does not really affect the simulation time.

Simulation step in each grid is also done independently from other grids, but there are some nuances. Step on the X axis is relatively simple. We are using a 5-point scheme, so for each grid node we update the value using 2 nodes to the left of it and 2 nodes to the right. Grid-characteristic method involves three phases for each node: transformation, reconstruction and increment. During the transformation phase, stress and velocity in the node and two of its neighbors on each side are converted to interim values called ω . We will refer to these values as *ppw*, *pw*, *w*, *nw* and *nnw*. During the reconstruction phase, a new ω value is calculated for the

node. Finally, the increment phase calculates new stress tensor and velocity from the new ω . After that this process is repeated for the next node. It is important to note that for this node we have already calculated 4 out of 5 ω values: pw , w , nw and nnw for the previous node are ppw , pw , w and nw for the next one correspondingly. Therefore we only need to calculate the value of nnw , and that will suffice for the reconstruction and increment. Parallel implementation of the step along the X axis consists in splitting the grid into stripes and generating an OpenMP task for each stripe. The task is doing a step for each row in the grid in its stripe. Since stripes do not overlap, these tasks are independent. Doing the step over the Y axis is more tricky. Since we are storing grid nodes in row-major layout (X is the fast axis), doing the step over the X axis naively is utilizing the CPU cache properly: next values, fetched into the cache from memory correspond to the next nodes we are going to work with. This is no longer true for the Y axis: trying to do a steps over all nodes in a column results in reading from 5 different memory locations and therefore many cache misses. To address this issue, we are using a different approach. Initially, the grid is again split into stripes on the Y axis, and each stripe is assigned its own OpenMP task. Inside the task we are doing the same phases: transformation, reconstruction and increment, but they are done not for individual nodes, but for whole rows at once. Figure 1 shows how this split works.

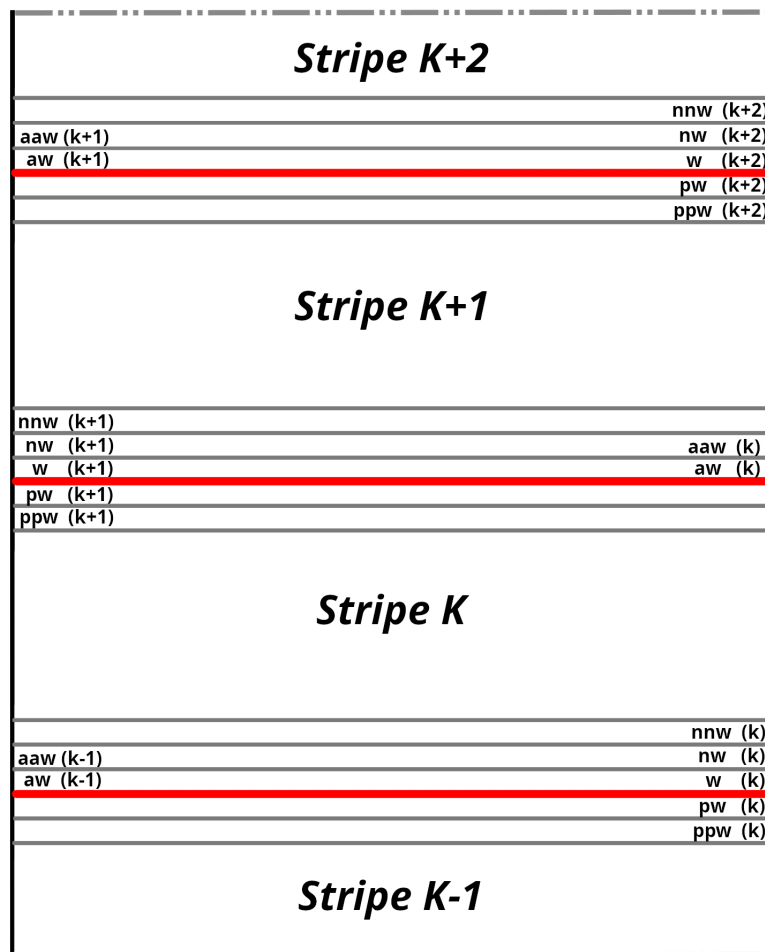


Figure 1. Illustration of splitting mesh to stripes and lines for which ω values are calculated at the initialization for each stripe

Since we are using the 5-node scheme, updating the last 2 rows in a stripe depends on the ω values of two first rows in the next stripe, referred to as aw and aaw in Fig. 1. We need to calculate these values in advance, since later they can be overwritten by a different task. But the ω values for these rows are also used as w and nw for the first 2 rows of the next stripe. This allows us to avoid computing them twice and simply copy them at the beginning. To balance the workload between the threads better, the number of tasks assigned for each grid is dynamically computed. At the beginning of the simulation we are only assigning 1 task for each grid, and measuring the time this task takes to complete. After that we use this time as a weight, and assign the number of tasks for each grid according to these weights. To avoid the unnecessary overhead, we also factor in sizes of grids, because small grids cannot efficiently utilize many tasks.

3. Testing

3.1. Problem Statement

To test the solver performance, we simulated a two-dimensional elastic wave propagation in homogeneous medium. Integration area was split into 20 tiles, each tile was described by a single grid. Each tile contained three nested grids with half the spacing, within one of which were also three nested grids with half the spacing (i.e., quarter the spacing of the main grid). An illustration of grid arrangement is presented in Fig. 2.

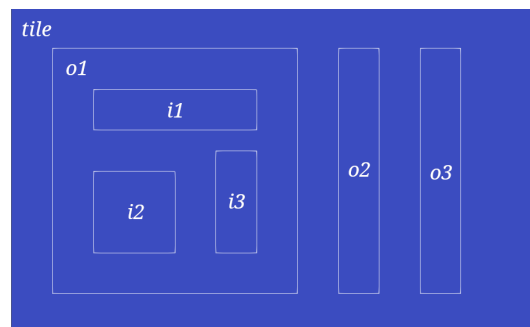


Figure 2. Problem statement illustration

Table 1. Grid sizes

Grid	Spacing (meters)	Size (nodes)
tile	2	260×160
o1	1	240×240
o2	1	40×240
o3	1	40×240
i1	0.5	320×80
i2	0.5	160×160
i3	0.5	80×200

3.2. Testing Environment

Performance testing was conducted on a server with 256Gb DDR4 RAM and two 12-core Intel Xeon processors (hyperthreading enabled), running Linux 6.8.0 kernel and Ubuntu 24.04.

Two compilers were used for testing: g++ 13.2.0 from the GNU Compiler Collection and icpx 2024.2.0 from the Intel OneAPI kit (based on clang++ from the LLVM project). These compilers use different OpenMP runtimes, and we were interested in checking whether it will have a significant impact on parallelization.

We did not use any manual CPU affinity configuration and allowed system scheduler and OpenMP runtimes to select processors and cores for threads according to their algorithms.

3.3. Testing Results

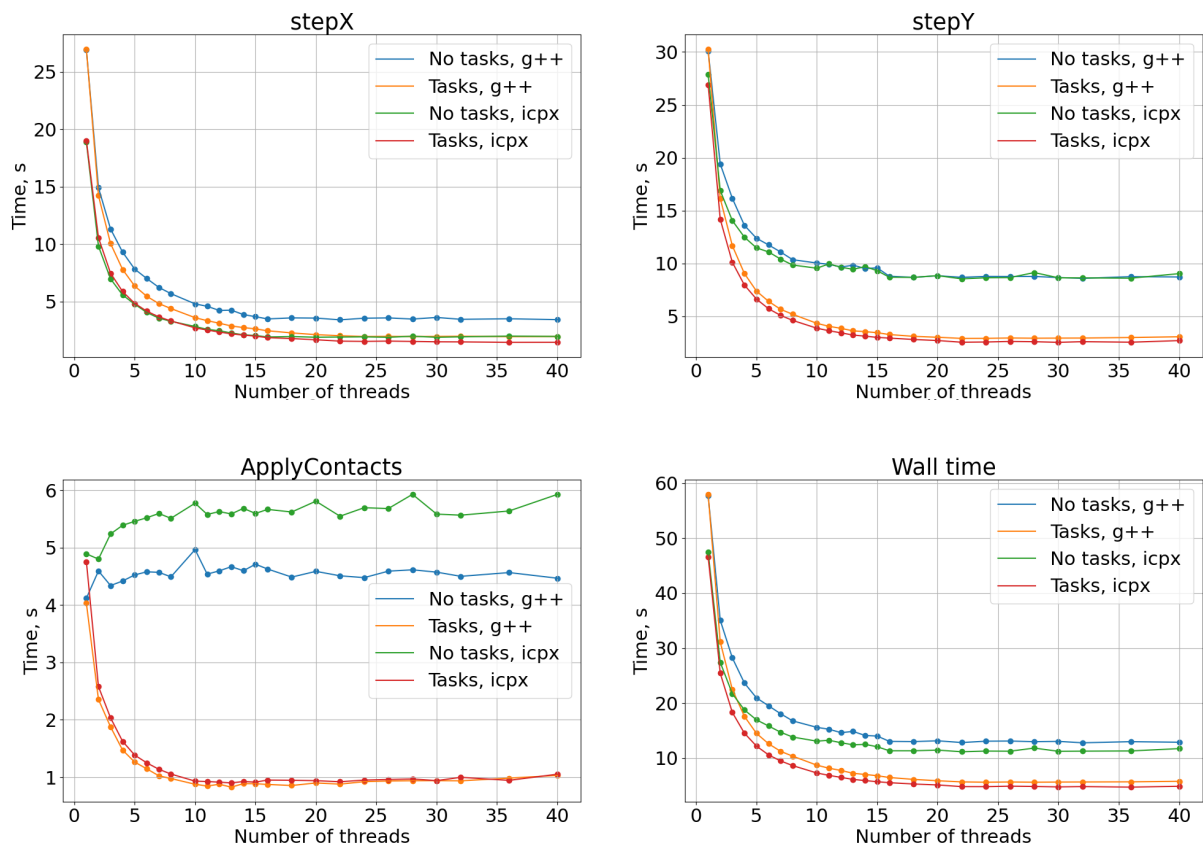


Figure 3. Testing result plots

We have conducted the testing varying the number of threads used by the program from 1 to 40. We have also tried using more threads, but the results were noisy and unreliable. Final time for each number of threads was calculated as an average of 3 runs to accommodate for possible external condition changes like CPU clock multiplier decrement caused by overheating. Since the simulation consists of multiple repeated steps, we do not need many runs for averaging the time and preventing the impact of short-term fluctuations.

We have measured the durations of different simulation stages separately, as well as the total simulation time, the plots are presented in Fig. 3. As we can see, task-based parallelism shows similar results to the simple `parallel for` of `stepX`. On `stepY` it significantly outperforms the naïve version, and has roughly the same performance as the `stepX`. `ApplyContacts` stage was

not parallelized before, and plots show that while it can benefit from parallel implementation, the number of threads it can use efficiently is low, and the time save is also relatively small.

We can also see that `icpx`-compiled solver performs slightly better than `g++`-compiled, but they do not have any significant difference in parallelism efficiency. From this we assume that different OpenMP runtimes do not have any significant impact on the performance of our application, and the difference in performance is caused by `icpx` doing optimization and vectorization slightly better.

3.4. Performance Analysis

To identify the bottlenecks, we have conducted an additional performance analysis. For this purpose we have disabled the hyper threading and ran a simulation with 20 threads limitation with an increased number of time steps.

According to the Intel VTune [1] statistics, serial execution time is approximately 9% of the total simulation duration, therefore most performance-critical operations are already executed in parallel regions, and the performance cannot be significantly improved by parallelizing serial regions.

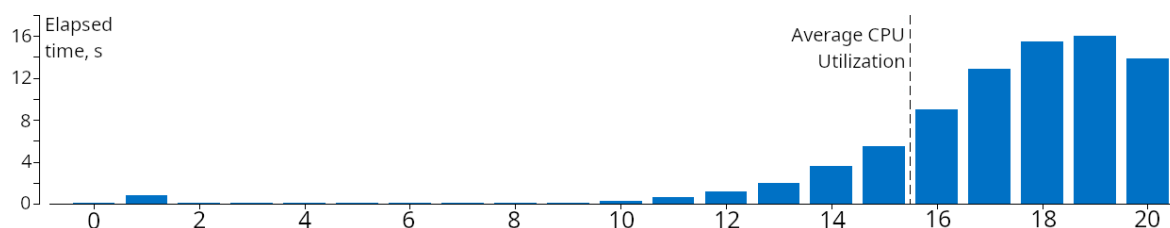


Figure 4. Thread activity histogram

Figure 4 shows the histogram of thread activity. From this histogram we can see that thread workload imbalance is noticeable, but not critical. This means that our heuristics might need some tuning in future, but this is not the main performance bottleneck. Intel VTune estimates the potential performance gain here to be around 8%.

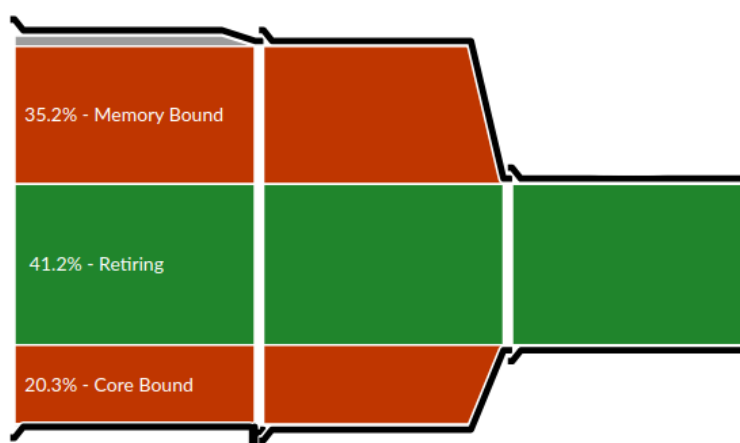


Figure 5. Microarchitecture exploration

Microarchitecture analysis shows (Fig. 5), that there are many backend-bound pipeline slots. Since the simulation consists of different stages, it is important to explore the bottlenecks in each of them. Table 2 contains the detailed data.

Table 2. Microarchitecture metrics of different operations

	OmegaX	OmegaY	reconstruct	incrementX	incrementY
Retiring	7.10%	16.70%	68.70%	53.30%	61.40%
Frontend Bound	0.20%	1.40%	0.50%	0.80%	0.90%
Bad speculation	1.10%	1.70%	0.00%	6.00%	3.40%
Backend Bound	91.50%	80.20%	30.80%	39.90%	34.30%
Memory	74.10%	61.10%	9.30%	11.40%	7.10%
L1 Bound	14.70%	0.00%	-	-	-
L2 Bound	39.20%	33.90%	-	-	-
L3 Bound	0.00%	6.40%	-	-	-
DRAM Bound	15.80%	30.70%	-	-	-
Bandwith	41.90%	80.40%	-	-	-
Latency	49.80%	11.60%	-	-	-
Store Bound	0.00%	1.10%	-	-	-
Core	17.40%	19.10%	21.50%	28.50%	27.20%
Divider	0.00%	0.80%	0.30%	20.00%	27.90%
Port utilization	16.40%	20.30%	49.10%	32.80%	41.20%
0-port	66.60%	68.80%	21.60%	12.30%	14.30%
1-port	19.50%	15.40%	24.70%	16.50%	10.90%
2-port	13.90%	7.00%	24.10%	17.10%	26.70%
≥ 3 -port	7.50%	16.30%	25.10%	48.10%	51.50%

From this table we can see that **OmegaX** and **OmegaY** (transformation phase) are memory-bound, while **reconstruct**, **incrementX** and **incrementY** are utilizing the microarchitecture efficiently, but could benefit from better vectorization.

NUMA remote accesses stand for approximately 44% of all memory access operations, but NUMA access latency does not play a significant role in overall solver performance.

Practical Example

Using the software we developed, we have simulated the elastic wave propagation in a fractured layer. The simulated area size was 3000×500 meters. The area contained 30 fractures, 80 to 82 meters long, all at the same depth, but varying angles. Apart from fractures, the medium was isotropic with $C_p = 2698$ m/s, $C_s = 1730$ m/s and $\rho = 2259$ kg/m³.

The main grid size was 15360×2560 nodes with a grid step 0.196 m along both axes. Each fracture was modelled using an additional 417×9 nodes grid.

The time step of the simulation was $5 \cdot 10^{-5}$ s, and a total of 12000 time steps were modelled. The wave originated from a point source near the middle of the area. Ricker wavelet was used as a source function.

Figure 6 shows the wavefield images 50, 100, 150, 200, 250, 300 and 400 ms after the source activation.

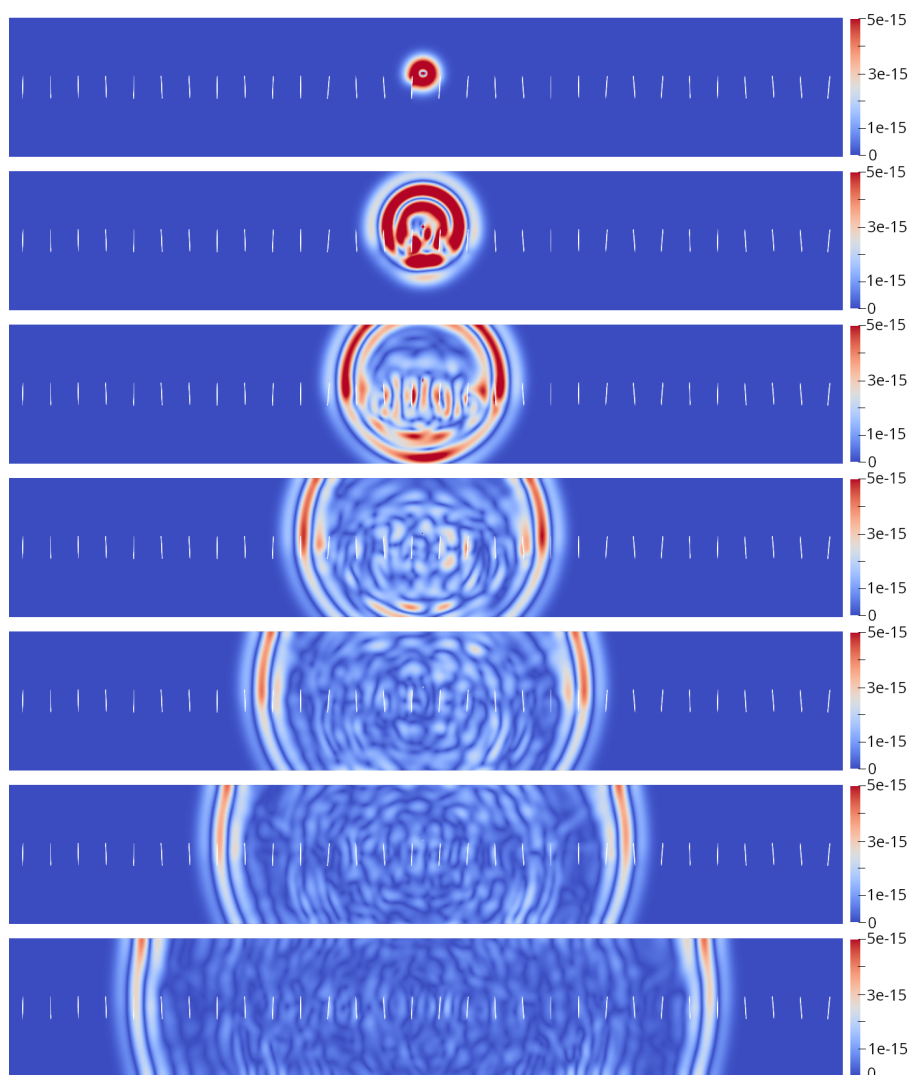


Figure 6. Wavefield snapshot at different time moments

Since the main grid has significantly more nodes than all crack grids combined, we did not observe any substantial reduction of the total simulation time. Nevertheless, we have confirmed that the new implementation does not introduce any computational errors or performance degradation. We have also observed the performance improvement in the crack and contact simulation phases, as it was expected.

Conclusion

This paper proposes a novel algorithm for the parallelisation of dynamic wave processes in heterogeneous media. The grid-characteristic method was employed as a numerical technique for the resolution of the resulting system of partial differential equations. The distinctive feature of this approach was the utilization of overset chimera meshes. The aforementioned factors contributed to the complexity of parallelisation, given the unstructured nature of the links between the computational meshes and the variability in mesh size. The proposed algorithm is based on the OpenMP task parallelisation mechanism and has been implemented and tested. The results of this testing have demonstrated the efficacy of the proposed algorithm. Work is currently underway to extend the algorithm to three-dimensional scenarios.

Acknowledgements

The work was carried out within the framework of the state task of the NRC “Kurchatov Institute” – SRISA on the topic No. FNEF-2024-0002 “Mathematical modeling of multi-scale dynamic processes and virtual environment systems” (1023032900401-5-1.2.1).

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. Intel VTune Profiler User Guide – intel.com. <https://www.intel.com/content/www/us/en/develop/documentation/vtune-help/top.html>, accessed: 2024-07-01
2. Appelö, D., Kreiss, G.: A new absorbing layer for elastic waves. *Journal of Computational Physics* 215, 642–660 (2006). <https://doi.org/10.1016/J.JCP.2005.11.006>
3. Benek, J., Buning, P., Steger, J.: A 3-D chimera grid embedding technique. *American Institute of Aeronautics and Astronautics* (1985). <https://doi.org/10.2514/6.1985-1523>
4. Benek, J.A., Steger, J.L., Dougherty, F.C., Buning, P.G.: Chimera. a grid-embedding technique (1986), <https://apps.dtic.mil/sti/citations/ADA167466>
5. Biryukov, V.A., Miryakha, V.A., Petrov, I.B., Khokhlov, N.I.: Simulation of elastic wave propagation in geological media: Intercomparison of three numerical methods. *Computational Mathematics and Mathematical Physics* 56(6), 1086–1095 (jun 2016). <https://doi.org/10.1134/S0965542516060087>
6. Board, O.A.R.: OpenMP Application Programming Interface Specification 5.2 (2021), <https://www.openmp.org/specifications/>, accessed: 2024-07-01
7. Chelnokov, F.B.: Explicit representation of grid-characteristic schemes for the elasticity equations in two- and three-dimensional space. *Mathematical modeling* 18, 96–108 (2006)
8. Favorskaya, A.V., Khokhlov, N.I., Petrov, I.B.: Grid-characteristic method on joint structured regular and curved grids for modeling coupled elastic and acoustic wave phenomena in objects of complex shape. *Lobachevskii Journal of Mathematics* 41, 512–525 (4 2020). <https://doi.org/10.1134/S1995080220040083/FIGURES/16>
9. 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, 1485–1502 (10 2018). <https://doi.org/10.1111/1365-2478.12639>
10. Furgailo, V., Ivanov, A., Khokhlov, N.: Research of Techniques to Improve the Performance of Explicit Numerical Methods on the CPU. In: 2019 Ivannikov Memorial Workshop (IVMEM). pp. 79–85. IEEE (sep 2019). <https://doi.org/10.1109/IVMEM.2019.00019>
11. Galis, M., Moczo, P., Kristek, J.: A 3-D hybrid finite-difference-finite-element viscoelastic modelling of seismic wave motion. *Geophysical Journal International* 175(1), 153–184 (oct 2008). <https://doi.org/10.1111/j.1365-246X.2008.03866.x>

12. Ivanov, A.M., Khokhlov, N.I.: Parallel implementation of the grid-characteristic method in the case of explicit contact boundaries. *Computer Research and Modeling* 10, 667–678 (2018). <https://doi.org/10.20537/2076-7633-2018-10-5-667-678>
13. Käser, M., Dumbser, M.: An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - I. The two-dimensional isotropic case with external source terms. *Geophysical Journal International* 166(2), 855–877 (aug 2006). <https://doi.org/10.1111/j.1365-246X.2006.03051.x>
14. Khokhlov, N., Petrov, I.: Application of the grid-characteristic method for solving the problems of the propagation of dynamic wave disturbances in high-performance computing systems. *Proceedings of the Institute for System Programming of the RAS* 31(6), 237–252 (2019). [https://doi.org/10.15514/ISPRAS-2019-31\(6\)-16](https://doi.org/10.15514/ISPRAS-2019-31(6)-16)
15. Khokhlov, N.I., Favorskaya, A., Furgailo, V.: Grid-Characteristic Method on Overlapping Curvilinear Meshes for Modeling Elastic Waves Scattering on Geological Fractures. *Minerals* 12(12), 1597 (dec 2022). <https://doi.org/10.3390/min12121597>
16. Kozhemyachenko, A.A., Petrov, I.B., Favorskaya, A.V., Khokhlov, N.I.: Boundary Conditions for Modeling the Impact of Wheels on Railway Track. *Computational Mathematics and Mathematical Physics* 60(9), 1539–1554 (sep 2020). <https://doi.org/10.1134/S0965542520090110>
17. LeVeque, R.J.: *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press (8 2002). <https://doi.org/10.1017/CB09780511791253>
18. Lisitsa, V., Tcheverda, V., Botter, C.: Combination of the discontinuous Galerkin method with finite differences for simulation of seismic wave propagation. *Journal of Computational Physics* 311, 142–157 (apr 2016). <https://doi.org/10.1016/j.jcp.2016.02.005>
19. Liu, W.K., Li, S., Park, H.S.: Eighty years of the finite element method: Birth, evolution, and future. *Archives of Computational Methods in Engineering* 29, 4431–4453 (6 2022). <https://doi.org/10.1007/S11831-022-09740-9>
20. Luo, S., Chen, Z.D.: A FDTD-based modal PML. *IEEE Microwave and Wireless Components Letters* 16, 528–530 (2006). <https://doi.org/10.1109/LMWC.2006.882408>
21. Magomedov, K., Kholodov, A.: The construction of difference schemes for hyperbolic equations based on characteristic relations. *USSR Computational Mathematics and Mathematical Physics* 9(2), 158–176 (1969). [https://doi.org/https://doi.org/10.1016/0041-5553\(69\)90099-8](https://doi.org/https://doi.org/10.1016/0041-5553(69)90099-8)
22. Mitskovets, I., Sagan, V., Khokhlov, N.: Parallel Modeling of Elastic Wave Propagation, with Explicit Pore Delineation Using Overset Grids Method. *Physics of Particles and Nuclei* 55(3), 516–518 (jun 2024). <https://doi.org/10.1134/S1063779624030602>
23. Mitskovets, I., Stetsyuk, V., Khokhlov, N.: Novel approach for modeling curved topography using overset grids and grid-characteristic method pp. 1–5 (1 2021). <https://doi.org/10.3997/2214-4609.202011784>
24. Novacki, W.: *Theory of Elasticity*. MIR (1975)

25. Vishnevsky, D., Lisitsa, V., Tcheverda, V.: Efficient Finite-difference Algorithm for Simulation of Seismic Waves in Models with Anisotropic Formations (apr 2012). <https://doi.org/10.3997/2214-4609.20143647>
26. Voigt, W.: Lehrbuch der kristallphysik (mit ausschluss der kristalloptik). B.G. Teubner (1910)
27. Yao, H.M., Jiang, L.: Machine-Learning-Based PML for the FDTD Method. IEEE Antennas and Wireless Propagation Letters 18, 192–196 (1 2019). <https://doi.org/10.1109/LAWP.2018.2885570>
28. Zienkiewicz, O.C., Taylor, R.L.R.L., Zhu, J.Z.: The finite element method: its basis and fundamentals. Elsevier, 7 edn. (2013)

HPC Optimization Algorithm for Assessing Haemodynamic Parameters in Synthetic Patient Cohorts

Artem V. Rogov^{1,2} , Timur M. Gamilov^{1,3} , Yaroslav Yu. Kirichenko⁴ ,
Philipp Yu. Kopylov¹ , Sergey S. Simakov^{2,3,4} 

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

A computational framework for the generation of a synthetic pulse wave database is developed. This framework demonstrates the feasibility of generating large-scale, high-fidelity virtual patient cohorts for biomedical research. Pulse waves are generated using a one-dimensional hemodynamic model of the systemic circulation coupled with a model of the left heart. Each virtual patient in the database is defined by a set of physiological parameters, including systolic and diastolic blood pressure, stroke volume, and heart rate. The parameters are optimized to match the desired outputs by solving an inverse problem using the Unscented Kalman Filter (UKF). The UKF is selected for its ability to accurately and efficiently estimate parameters in nonlinear systems. The generation of a single virtual patient requires between one and several hundred iterations of the UKF, depending on the complexity of the desired outputs. To meet the computational demands of generating a database with thousands of virtual patients, a computing cluster with 24 CPU nodes, each containing 52 cores, is employed. Two levels of parallelization are implemented, resulting in a speedup factor of 8.

Keywords: Unscented Kalman Filter, hemodynamic model, inverse problem, synthetic data base, virtual population, parallel computing, high-performance computing.

Introduction

Synthetic patient cohorts are becoming increasingly popular in clinical trials. They make it possible to generate synthetic data that is very close to real data, without having to deal with financial costs or ethical restrictions. Synthetic data can be used to design clinical trials, develop novel educational tools, or perform preliminary tests of various diagnostic techniques. This is particularly important in the age of data mining and artificial intelligence. Synthetic databases can be generated with machine learning and neural networks, or they can be used to train a new diagnostic method based on machine learning.

There are many ways to generate synthetic patient data. Some of them are based on statistical simulations [13, 14]. These methods analyse existing real patient cohorts and look for different patterns and distributions that can be used to generate a new virtual patient. Another approach is based on mathematical models of physiological processes [2, 7]. In this case, we assume that the process can be described by a system of differential and/or algebraic equations. Each virtual patient is associated with a set of input parameters. Once the input parameters are defined, we can solve the system of equations and calculate physiological characteristics: pulse waves [7], blood pressures [21], body weight changes [11], etc. It can be expensive or time-consuming to measure these characteristics in large numbers of patients. An example of this approach are synthetic pulse wave databases generated using blood flow models [7, 24].

¹World-Class Research Center “Digital Biodesign and Personalized Healthcare”, Sechenov First State Medical University, Moscow, Russian Federation

²Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

³Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russian Federation

⁴I.M. Sechenov First Moscow State Medical University (Sechenov University), Moscow, Russian Federation

Synthetic pulse wave databases consist of thousands of virtual patients with unique characteristics: blood pressure, heart rate, stroke volume, etc. Each synthetic database should represent a subset of the real world demographics. This is achieved by assigning some target characteristics to each virtual patient. Some of these characteristics can be associated with input parameters of the blood flow model, but some are associated with output values (blood pressure). As a result, one must solve an inverse problem [18] to obtain a desired virtual patient. An inverse problem is usually a computationally expensive task that requires numerous simulations to solve. A number of methods have been developed to approach an inverse problem. Some modern methods utilize physics-informed neural networks [10], which are used in 3D computations when the boundary information is difficult to model. A more classical approach involves Bayesian optimization [8] or Kalman filter [23].

There are several types of algorithms used in conjunction with Kalman filtering. All of these algorithms are used to solve nonlinear optimisation problems and are primarily used in navigation systems, control systems and signal processing. The algorithm allows system feedback (e.g., noisy experimental data) to be incorporated into the estimation of system state parameters.

The Unscented Kalman Filter (UKF) is a modification of Kalman filter that has been widely used for parameter estimation in discrete-time dynamic models, particularly in biomedical applications. For example, in [6], the UKF is used to estimate terminal resistances and arterial wall parameters from experimental flow and pressure data. Similarly, [19] demonstrates the use of UKF to estimate absolute blood volume during haemodialysis based on dynamic measurements of hematocrit and ultrafiltration rates. Another related study focuses on dialysis patients with an emphasis on personalised treatment approaches [1]. In addition, in [3] the UKF is used to estimate parameters of the Windkessel model and arterial wall stiffness.

The main focus of this work is the Unscented Kalman Filter (UKF), which offers several advantages. The first advantage is the fact that the UKF considers an exact nonlinear system, unlike the modifications and other optimisation methods which use a linearised form. The second advantage is speed. To capture the current state, which is assumed to be a Gaussian random variable distribution, only a minimal set of so-called sigma points is used. The third advantage is robustness. The UKF is robust in estimating the posterior distribution. It accurately approximates both the covariance and the mean to third order precision for any type of non-linearity [23].

An overview of the relationship and differences between the UKF and other algorithms, as well as a brief description, mathematical background and fundamentals can be found in [23]. A detailed description of the UKF and a demonstration of some of its most important properties, as well as the definition of sigma points, can be found in [17].

We propose a computational framework that uses UKF to generate a virtual patient from a one-dimensional hemodynamic model. One-dimensional hemodynamic used to generate virtual patients is based on the ADAN56 structure of systemic circulation [5]. ADAN56 includes 56 major arteries of systemic circulation and elastic compartments connected to the terminal vessels to simulate peripheral arteries. The inlet of the aorta is connected to two-chamber model of the left heart that take into account the angle of valve opening.

Each virtual patient has three target output characteristics: systolic blood pressure, diastolic blood pressure and stroke volume. Target input characteristics include heart rate, left ventricular ejection time and aortic diameter. To achieve the target output characteristics, a set of 13 model parameters are adjusted. Adjusting the model parameters for each virtual patient

involves dozens and sometimes hundreds of calculations with a blood flow model. At the same time, a synthetic database should include a few thousand virtual patients. This means that the generation of a single database requires around 10^5 – 10^6 blood flow simulations of the full systemic circulation. This computationally intensive task is handled with the help of a computing cluster with 24 CPU nodes, each with 52 cores. The unscented Kalman filter allows effective parallelization of the computations. We describe two types of parallelization and estimate the reduction in computation time. After implementation of the optimizations computational time was reduced from 700 days to 10 days.

The remainder of this paper is organised as follows. Section 1.1.1 presents the description of 1D blood flow model. Section 1.1.2 describes lumped model of the left heart. Inverse problem is formulated in section 1.2. Unscented Kalman Filter and it's modification that is used to solve the inverse problem are described in sections 1.3 and 1.4. Section 2.1 presents calculation results: we estimate the amount of UKF iterations required to solve an inverse problem and estimate the speedup achieved with the help of parallelization. We discuss results and possible future work in section 2.1. The list of abbreviations used is presented in the end of the paper.

1. Methods

1.1. One-dimensional Hemodynamic Model with Lumped Heart

1.1.1. One-dimensional hemodynamic model

We generate pulse waves of virtual patients with the help of one-dimensional model of blood flow in systemic circle [9, 21]. This model represents each artery as one-dimensional elastic tube. The blood is assumed to have constant density $\rho = 1.06 \text{ g/cm}^3$ and constant viscosity $\mu = 4 \text{ mPa}\cdot\text{s}$. For each artery with index k we formulate mass and momentum conservation laws:

$$\frac{\partial A_k}{\partial t} + \frac{\partial(A_k u_k)}{\partial x} = 0, \quad (1)$$

$$\frac{\partial u_k}{\partial t} + \frac{\partial(u_k^2/2 + p_k/\rho)}{\partial x} = \psi_k, \quad (2)$$

where x is the coordinate along the artery, t is time, $A_k(t, x)$, $u_k(t, x)$, $p_k(t, x)$ are cross-sectional area, blood flow velocity (averaged over cross-section), and blood pressure, respectively. The ψ_k is the friction force:

$$\psi_k = -\frac{8\pi\mu u_k}{\rho A_k}. \quad (3)$$

The tube law closes the system (1)–(2) and describes the elastic properties of the artery:

$$p_k(S_k) = \rho c_k^2 f(A_k), \quad (4)$$

where c_k is the speed of small disturbances propagation. $f(A_k)$ is a piecewise function:

$$f(S_k) = \begin{cases} \exp(A_k/A_k^0 - 1) - 1, & A_k > A_k^0, \\ \ln(A_k/A_k^0), & A_k \leq A_k^0. \end{cases} \quad (5)$$

A_k^0 is a cross-section when the pressure is zero. A_k^0 can be derived from the diameter of the artery d_k^0 : $A_k^0 = \frac{\pi(d_k^0)^2}{4}$.

At arterial junctions points we impose mass conservation law and continuity of total pressure. Conservation of mass implies that the algebraic sum of flows in vessels at the junction is zero:

$$\sum_{k=k_1, \dots, k_M} Q_k \epsilon_k = 0, \quad (6)$$

where M is the number of arteries attached to the junction; $Q_k = u_k(t, \tilde{x}_i) A_k(t, \tilde{x}_i)$; $\tilde{x}_k = L_k$ and $\epsilon_k = 1$ for incoming arteries (L_k is the length of the artery); $\tilde{x}_k = 0$ and $\epsilon_k = -1$ for outgoing arteries.

Continuity of total pressure is expressed as:

$$p_i(A_i(t, \tilde{x}_i)) + \frac{\rho u_i^2(t, \tilde{x}_i)}{2} = p_j(A_j(t, \tilde{x}_j)) + \frac{\rho u_j^2(t, \tilde{x}_j)}{2}, \quad (7)$$

where i and j are indices of two arteries connected to the junction. Index i is fixed and index j runs through the indices of all other connected arteries, $i \neq j$.

All arteries are connected into a network that represents arterial part of systemic circulation. Lengths, diameters and the structure of the network correspond to ADAN56 model [5], which contains 56 largest arteries of systemic circulation under normal physiological conditions.

The inlet of the aorta is connected to the two-chamber lumped parameter left heart model described in section 1.1.2. The outflow conditions assume that the systemic arteries are connected to the 3-element Windkessel compartments, which represent the peripheral vasculature. Thus, at the end of each terminal vessel with an index k we impose the following condition:

$$Q_k(t, L_k) \left(1 + \frac{R_1}{R_2}\right) + C R_1 \frac{dQ_k(t, L_k)}{dt} = \frac{p_k(t, L_k) - p_{out}}{R_2} + C \frac{dp_k(t, L_k)}{dt}, \quad (8)$$

where $Q_k(t, L_k) = A_k(t, L_k) u_k(t, L_k)$; R_1, R_2, C are two resistances and compliance of Windkessel compartments; p_{out} is the outflow pressure (same for all compartments). Parameters R_1, R_2, C are individual for each terminal vessel. They are defined in ADAN56 model [5].

Equations (1) and (2) are solved in inner points of each artery with the help of explicit grid-characteristic scheme. The mass conservation law and total pressure continuity equations (7) together with compatibility conditions of the set (1)–(2) are solved with the help of Newton's method. Details of numerical implementation are presented in [20]. This approach allows us calculate values of cross-sectional area, velocity and pressure at the new time step independently in all arteries and junction points. It simplifies parallelization of calculations since each artery and each junction can be assigned to a separate thread.

1.1.2. Lumped parameter model of left heart with valve dynamics

In this part, we introduce a set of differential equations that describe the dynamics of the heart chambers, specifically the left atrium (la) and left ventricle (lv). These chambers are connected to the pulmonary veins (plv) and the aortic root. Additionally, the dynamics of the mitral valve (mv) and aortic valve (av) are also described. In contrast to predefined cardiac output [7], this model incorporates heart disease dynamics such as valve regurgitation.

The heart chambers volumes follow equation

$$I_k \frac{d^2 V_k}{dt^2} + R_k P_k \frac{dV_k}{dt} + E_k(t)(V_k - V_k^0) + P_k^0 = P_k, \quad k = la, lv, \quad (9)$$

where k is the index of chambers; I_k is the coefficient of inertia; R_k is the coefficient of hydraulic resistance; V_k is the pressure averaged over the chamber volume; P_k is the pressure averaged over the chamber volume; P_0 and V_0 are reference values of pressure and volume. $E_k(t)$ reflects variable elasticity during contraction [16].

$$E(t) = E^d + \frac{E^s - E^d}{2}e(t), \quad 0 \leq e(t) \leq 1, \quad (10)$$

where $e(t)$ represents periodic activation potential. For the left ventricle the function is:

$$e_{lv}(t) = \begin{cases} 0.5(1 - \cos(\frac{t}{T_{s1}}\pi)), & 0 \leq t < T_{s1}, \\ 0.5(1 - \cos(\frac{t-T_{s2}}{T_{s1}-T_{s2}}\pi)), & T_{s1} \leq t < T_{s2}, \\ 0, & T_{s2} \leq t \leq T. \end{cases} \quad (11)$$

$T = \frac{60}{HR}$ is the duration of cardiac cycle, which can be derived from the heart rate (HR); T_{s1} is the time to systolic peak; T_{s2} is the duration of the systole which we assume to be equal to left ventricular ejection time (LVET). For left atrium the function is:

$$e_{la}(t) = \begin{cases} 0, & 0 \leq t < T_{pb}, \\ 0.5(1 - \cos(\frac{t-T_{pb}}{T_{pw}}2\pi)), & T_{pb} \leq t < T, \end{cases} \quad (12)$$

where T_{pb} is the beginning of the atrial contraction and T_{pw} is the duration of the atrial contraction.

The mass conservation law for left atrium and left ventricle can be expressed as:

$$\begin{aligned} \frac{dV_{la}}{dt} &= Q_{plv} - Q_{mv}, \\ \frac{dV_{lv}}{dt} &= Q_{mv} - Q_{av}. \end{aligned} \quad (13)$$

The relationship between flow and pressure drop across left atrial inlet ($Q_{plv}, \Delta P_{plv}$), mitral valve ($Q_{mv}, \Delta P_{mv}$) and aortic valve ($Q_{av}, \Delta P_{av}$) is defined as:

$$\Delta P_m = L_m \frac{dQ_m}{dt} + \beta_m Q_m |Q_m|, \quad m = plv, mv, av. \quad (14)$$

Here L_m is constant, β_m depends on area of the valve according to [25, 26]:

$$\begin{aligned} \beta_{mv} &= \frac{\rho}{2B_{mv}} \left(\frac{1}{A_{mv}} - \frac{1}{A_{ref}} \right)^2, \\ \beta_{av} &= \frac{\rho}{2B_{av}} \left(\frac{1}{A_{av}} - \frac{1}{A_a} \right)^2, \\ \beta_{plv} &= const. \end{aligned} \quad (15)$$

B_{mv} and B_{av} are constant parameters; A_{mv} , A_{av} are valve effective cross-section areas (depend on the angle); A_a is cross-sectional area of aorta at the inlet of the network of arteries; $A_{ref} = 5 \text{ cm}^2$.

For each valve effective cross-section area A_v can be calculated from the maximum cross-section area A_v^{max} and opening angle θ as $A_v(\theta) = g(\theta)A_v^{max}$, where $g(\theta)$ is a smooth function

that equals 1 when the valve is fully open ($\theta = \theta_{max}$) and 0 when the valve is fully closed ($\theta = \theta_{min}$):

$$g(\theta) = \begin{cases} \frac{(1 - \cos \theta_{min})^2}{(1 - \cos \theta_{max})^2}, & \theta < \theta_{min}, \\ \frac{(1 - \cos \theta)^2}{(1 - \cos \theta_{max})^2}, & \theta_{min} \leq \theta \leq \theta_{max}, \\ 1, & \theta > \theta_{max}. \end{cases} \quad (16)$$

The opening-closing dynamics of the valves are modeled using Newton's second law, where valve motion depends on the pressure gradient, friction, and a restoring force [15]:

$$\begin{aligned} \frac{d^2\theta_{mv}}{dt^2} &= (P_{la} - P_{lv})K_{mv}^p \cos\theta_{mv} - K_{mv}^f \frac{d\theta_{mv}}{dt} - F_{mv}(\theta_{mv}), \\ \frac{d^2\theta_{av}}{dt^2} &= (P_{lv} - P_a)K_{av}^p \cos\theta_{av} - K_{av}^f \frac{d\theta_{av}}{dt} - F_{av}(\theta_{av}), \end{aligned} \quad (17)$$

where θ_{mv} and θ_{av} are opening angles of mitral and aortic valves; K_{mv}^p , K_{mv}^f , K_{av}^p , K_{av}^f are constants; P_a is the pressure in the aortic root. The virtual dumping force $F(\theta)$ for each valve prevents it from opening more than maximum angle θ_{max} and less than minimum angle θ_{min} :

$$F(\theta) = \begin{cases} 0, & \theta_{min} < \theta < \theta_{max}, \\ e^{B(\theta - \theta_{max})} - 1, & \theta > \theta_{max}, \\ -(e^{B(\theta_{min} - \theta)} - 1), & \theta_{min} < \theta, \end{cases} \quad (18)$$

where $B = 10^3$.

Pressure drop across left atrial inlet ΔP_{plv} in (14) is the difference between the pressure in pulmonary veins P_{plv} and the pressure in left atrium P_{la} . Pulmonary veins pressure P_{plv} is equal to 13 mm Hg by default but can be adjusted during parameter optimization.

1.2. Parameter Optimization Problem

The aim of the optimisation is to fit the model parameters to three key medical indicators: systolic blood pressure (SBP), diastolic blood pressure (DBP), and left ventricular stroke volume (SV). These three indicators were chosen for their clinical relevance and ease of measurement in practice [22]. SBP and DBP are associated with maximum and minimum blood pressures in the brachial artery of our arterial network. SV is the amount of blood ejected from the left ventricle into the aorta during a single cardiac cycle. The target values for SBP, DBP, and SV are indicated by the lower index t .

To achieve target values SBP_t , DBP_t and SV_t , we adjust the following set of 13 parameters: $R_{la}, R_{lv}, I_{la}, I_{lv}$ in (9); $K_{mv}^p, K_{av}^p, K_{mv}^f, K_{av}^f$ in (17); p_{out} in (8); P_{plv} ; coefficients for total resistance and total compliance of all Windkessel compartments κ_R and κ_C ; coefficient for all c_k in (4) κ_s .

The coefficient κ_s for the speed of small disturbances propagation c_k is used to change the stiffness of all arteries at once. We do not change c_k for each artery independently, we multiply c_k of all arteries by a single coefficient. Similar coefficients are introduced for windkessel resistances κ_R and windkessel compliances κ_C . These three coefficients are used to modify the elastic properties of large arteries and peripheral vessels of a virtual patient.

The loss function $\mathcal{L} = \mathcal{L}(SBP, DBP, SV)$ for optimization task is defined as:

$$\mathcal{L} = \max \left\{ \frac{|SBP - SBP_t|}{SBP_t}, \frac{|DBP - DBP_t|}{DBP_t}, \frac{|SV - SV_t|}{SV_t} \right\}. \quad (19)$$

We adjust parameters of a virtual patient until $\mathcal{L} < 1\%$.

Each virtual patient is defined by desired (target) SBP, DBP, SV, HR, LVET, and major arteries diameters. SBP, DBP, and SV were discussed above. HR corresponds to the duration of cardiac cycle in (11) and (12). LVET is equal to T_{s2} in (11). Diameters of arteries are used to calculate A_k^0 in (5). Typical diameters of major arteries are taken from clinical literature [12].

To create a virtual population spanning different age groups, typical virtual patients were developed for each decade from 20 to 80 years of age [7]. Vessel dimensions and their standard deviations (SD) in diameter in an age group were adjusted based on age-specific data [12]. For each decade diameters of arteries were varied simultaneously and each artery can have three possible diameters: mean, mean + SD, mean - SD. Predefined SBP, DBP, SV, HR, LVET, and their standard deviations for each decade follow findings from [7].

The following parameters were varied for direct generation: SBP, DBP, SV, HR, LVET, diameters of large arteries. We modify the mean values of the parameters by incorporating the standard deviation, adjusting them both upwards and downwards. Consequently, each parameter takes on three possible values: the mean, the mean plus standard deviation, and the mean minus standard deviation. This approach generates $3^6 = 729$ virtual patients for each age group.

In this work we focus on the age group of 20–30 years old. Using a baseline model for a 25-year-old patient, we optimize parameters to achieve target SBP, DBP, and SV values. Table 1 presents baseline and SD values for SBP, DBP, SV, HR, and LVET. Diameters of major arteries were modified simultaneously according to the data from [12].

Table 1. Parameter of a baseline 25 year old virtual patient [7]

Parameter	Value \pm SD
SBP, mmHg	112.3 \pm 8.7
DBP, mmHg	72.0 \pm 5.6
SV, ml	73.0 \pm 13.1
HR, beats/min	73.0 \pm 9.1
LVET, ms	283 \pm 23

1.3. The Unscented Kalman Filter for Parameter Estimation

The following section outlines the general structure of the Unscented Kalman Filter. Subsequently, it will present the specific implementation employed in this study. Time-discretized system of the one-dimensional hemodynamic model with left heart described in Section 1.1.2 can be presented as:

$$X_{n+1} = \tilde{F}(X_n, \Theta_n), \quad (20)$$

where X_n is vector, which represents state of the system at n -th iteration. X_n represents all values of pressure, flow, cross-sectional area in all point of all arteries at each moment of a cardiac cycle, along with the parameters of the Windkessel models and the lumped parameters associated with the left heart. Θ_n is a set of 13 parameters to be estimated,

$\Theta = [R_{la}, R_{lv}, I_{la}, I_{lv}, K_{mv}^p, K_{av}^p, K_{mv}^f, K_{av}^f, p_{out}, \kappa_R, \kappa_C, \kappa_s]^\top$; \tilde{F} is operator of propagation of the system on next step, which depends on underlying model equations and particular type of discretization.

$$Z_n = \tilde{H}(X_n) + \epsilon, \quad (21)$$

where Z_n is vector of measured variables at some time instants, \tilde{H} is operator, which transforms state of the system to specific measured variables, ϵ is a vector of Gaussian noise. Further in this paper it is assumed, that $Z_n = [SBP_n, DBP_n, SV_n]^\top$, and SBP_t, DBP_t, SV_t are target or required values for virtual patient. Bearing in mind that all these values can only be calculated after a complete cardiac cycle, one sampling step is considered as one cardiac cycle. For other tasks and applications, the temporal sampling interval may vary according to the specific objective and the frequency of measurements available for the system under consideration.

Algorithm of any Kalman filter can be represented as prediction-correction scheme for state (X_n, Θ_n) , where prediction step calculates X_{n+1}^- and Θ_{n+1}^- , assuming that parameters have trivial dynamic and do not change over time.

$$\begin{aligned} X_{n+1}^- &= \tilde{F}(X_n^+, \Theta_n), \\ \Theta_{n+1}^- &= \Theta_n. \end{aligned} \quad (22)$$

In the correction step the difference between the measurement and the one calculated according to the new state of the system is taken into account.

$$\begin{aligned} X_{n+1}^+ &= X_{n+1}^- + K_X(Z_{n+1} - \tilde{H}(X_{n+1}^-)), \\ \Theta_{n+1}^+ &= \Theta_{n+1}^- + K_\Theta(Z_{n+1} - \tilde{H}(X_{n+1}^-)). \end{aligned} \quad (23)$$

K_X and K_Θ are Kalman matrices defined in a way to minimize difference between actual measurement and observed state.

1.4. The Reduced-order Unscented Kalman Filter

The aim of this section is to examine the reduced-order Unscented Kalman Filter (ROUKF), which is the UKF implementation used in this study. These two methods share core principles: both filters approximate the mean and covariance of a nonlinear system by propagating a set of sigma points through the nonlinear function, unlike other versions of the Kalman filter that use a linearised system; both filters employ an identical prediction-correction strategy for state and covariance estimation.

As shown in [17], the filtering operation can only involve matrices of the same size as the unknown parameter space. Furthermore, it is shown that to estimate p parameters, only $p + 1$ sigma points are needed to capture the current state of the system. These sigma points can be defined recursively (the algorithm of this procedure is described in detail in [17]). Then, all the sigma points are integrated into a matrix I . This matrix has $p + 1$ vectors of size p , where the notation $I_{(i)}$ will represent one column of the I matrix and one concrete sigma point. The weights of the sigma points are stored in the diagonal matrix D , where the same and equal coefficients are on the diagonal $\alpha = \frac{1}{p+1}$.

The algorithm of ROUKF is presented in Algorithm 1. The stopping criterion for this algorithm can be chosen at the discretion of the researcher and is typically determined by the loss function. Once the criterion is met, the problem of selecting the parameter is considered to be solved. For more details check [4, 17].

Algorithm 1 Iterative Process of UKF

1: **Initialization step:**

2: Define covariance matrices:

$$\Sigma^\Theta = \text{diag} \{ \sigma_{\text{param},i}^2 \}, \quad i = 1, \dots, p \quad (24)$$

$$\Sigma^Z = \text{diag} \{ \sigma_{\text{obs},i}^2 \}, \quad i = 1, \dots, m \quad (25)$$

3: For the first step ($n = 0$), set the following matrices:

$$L^\Theta = \mathbb{1}, \quad L^X = \mathbf{0}, \quad U_0 = (\Sigma^\Theta)^{-1} \quad (26)$$

4: **while** loss function is higher than the threshold **do**

5: **Sigma-point Sampling:**

$$C_n = \sqrt{U_n^{-1}} \quad (\text{Cholesky factorization}) \quad (27)$$

$$X_{n,(i)}^+ = X_n^+ + L_n^X C_n^\top I_{(i)}, \quad i = 1, \dots, p+1 \quad (28)$$

$$\Theta_{n,(i)}^+ = \Theta_n^+ + L_n^\Theta C_n^\top I_{(i)}, \quad i = 1, \dots, p+1 \quad (29)$$

6: **Forward Propagation / Prediction:**

$$X_{n+1,(i)}^- = \tilde{F}(X_{n,(i)}^+, \Theta_{n,(i)}), \quad i = 1, \dots, p+1 \quad (30)$$

$$\Theta_{n+1,(i)}^- = \Theta_{n,(i)}, \quad i = 1, \dots, p+1 \quad (31)$$

$$X_{n+1}^- = \sum_{i=1}^{p+1} \alpha X_{n+1,(i)}^- \quad (32)$$

$$\Theta_{n+1}^- = \sum_{i=1}^{p+1} \alpha \Theta_{n+1,(i)}^- \quad (33)$$

7: **Calculate Innovation:**

$$\Delta = Z_{n+1} - \tilde{H}(X_{n+1}^-) \quad (34)$$

8: **Update Covariances:**

$$L_{n+1}^X = X_{n+1}^- D I^\top, \quad D = \text{diag} \{ \alpha, \dots, \alpha \}, \quad (35)$$

$$L_{n+1}^\Theta = \Theta_{n+1}^- D I^\top \quad (36)$$

$$L_{n+1}^\Delta = \Delta_{n+1} D I^\top \quad (37)$$

$$U_{n+1} = I D I^\top + (L_{n+1}^\Delta)^\top (\Sigma^Z)^{-1} L_{n+1}^\Delta \quad (38)$$

9: **Correction:**

$$\Delta^{\text{sum}} = \sum_{i=1}^{p+1} \alpha \Delta_{(i)} \quad (\text{weighted sum of columns}) \quad (39)$$

$$X_{n+1}^+ = X_{n+1}^- - L_{n+1}^X U_{n+1}^{-1} (L_{n+1}^\Delta)^\top (\Sigma^Z)^{-1} \Delta^{\text{sum}} \quad (40)$$

$$\Theta_{n+1}^+ = \Theta_{n+1}^- - L_{n+1}^\Theta U_{n+1}^{-1} (L_{n+1}^\Delta)^\top (\Sigma^Z)^{-1} \Delta^{\text{sum}} \quad (41)$$

10: Set iteration number $n = n + 1$

11: **end while**

It is worth noting that the algorithmic implementation of the above problem has a number of peculiarities. The target parameters are $Z_n = [SBP_n, DBP_n, SV_n]^\top$. In order to calculate SBP_n , DBP_n , and SV_n , it is necessary to perform several cardiac cycles calculations until the values of SBP, DBP, and SV become constant from one cycle to another. The calculations are carried out until the periodic mode is established. For the purposes of this paper, we assume that a sufficient number of cardiac cycles is 9. Consequently, the determination of the set of states $X_{n+1,(i)}^-$ via the propagation operator \tilde{F} is a computationally demanding task. It is necessary to compute the problem of fluid flow through a network of pipes $p + 1$ times before it can be determined.

Furthermore, the vector of observations or the vector of target parameters Z_n remains constant, as the required values are invariant. This provides an alternative view of the algorithm. In more typical cases, the target vector will change over time, reflecting the evolution of the system.

The primary objective of the ROUKF is to overcome the limitations of the UKF. The UKF may require the implementation of costly matrix operations such as factorisation and inverse matrix computation on matrices of considerable dimension, equivalent to the number of state variables. Furthermore, if n represents the number of state variables, it propagates $2n + 1$ sigma points through the nonlinear system. Unlike the UKF, the ROUKF focuses on a lower dimensional state, thereby reducing the number of sigma points and consequently the computational complexity.

2. Results

2.1. Parallelization of Virtual Patient Generation Process

The process of selecting the appropriate parameters to solve the inverse problem using the Kalman filter is time consuming. To complete a single iteration of the Kalman filter algorithm, the problem must be computed 13 times until a solution is found. In the absence of parallelization, the completion of a single calculation requires around 320 seconds. It is now necessary to estimate the time required to generate the entire virtual population of 25 year olds. As previously stated, a mean of 13 iterations is required for each update of the parameters (for a single iteration of UKF). The average number of iterations required for the entire process is several dozen, with an estimated mean of 20. The total number of objects in the population is 729, thus the generation will take more than 700 days, which is an unacceptable timespan. In this section, the methodology used to parallelize the generation of virtual populations is described in detail.

The first type of parallelization is used in the calculation of the result of the forward propagation operator, when the resulting vector X_n of SBP , DBP , SV is obtained from the set of parameters Θ_n . The processes of calculating the values at the internal points of the vessels and at the points where the vessels are connected are both subject to parallelization.

The high-performance computational cluster of Sechenov University, comprising 24 CPU nodes with Intel(R) Xeon(R) Gold 6230R processors at 2.10GHz and 52 cores each, was employed for the computation. The cluster exhibited a peak performance of 84 tflops and a total distributed RAM capacity of 10.5 TB. The optimal number of cores for the calculation was determined by empirical experiments. Figure 1 illustrates the effect of core allocation on the speedup of the calculation of the vector of resulting values for a single virtual patient. An examination of the

data in the graph shows that it is pointless to exceed 15 cores for the specified calculation. In addition, there is no significant improvement in speed.

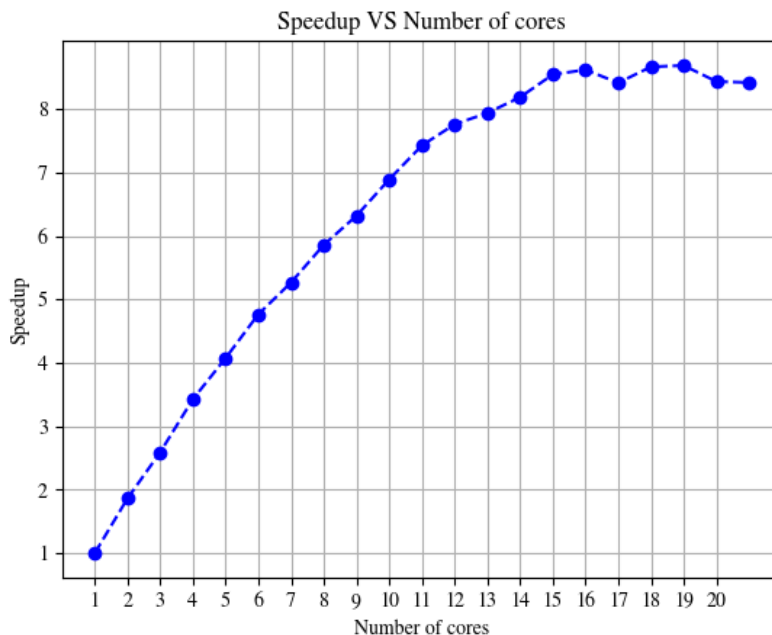


Figure 1. Dependence of calculation speedup on the number of cores

The second type of parallelization results from the fact that the parameter optimisation task is configured using a configuration file. This file contains a description of the geometric properties of the vessels as well as the values of the vascular and cardiac parameters. Therefore, it was convenient to organise the start of the parameter selection tasks by fixing the values of LVET and vessel diameters. Accordingly, once the configuration files had been fixed and all possible combinations of the corresponding parameters had been collected, 9 different tasks were identified. In other words, all virtual objects were divided into 9 different subsamples corresponding to different diameters of large vessels and LVET. These subsamples can be run in parallel on a cluster.

It should be noted that an additional way of speeding up the calculations is to distribute 13 independent tasks to different nodes of the cluster. These 13 tasks must be computed during the forward propagation step of the UKF algorithm. Due to the workload of the cluster, this feature is not implemented, but may be added in the future.

In consideration of the two aforementioned parallelizations, it can be estimated that the calculation of the entire virtual population will require approximately 9–10 days, which is consistent with the observed results.

The following three plots on Fig. 2 illustrate the convergence rate characteristic of the calculation of the required parameters for the whole sample. All cases can be classified into three categories based on their convergence characteristics: (A) fast convergence (1–2 filter steps, Fig. 2a), (B) “medium convergence” (several tens of filter steps, Fig. 2b), and (C) slow convergence (100 or more filter steps, Fig. 2c). The parameters of these cases are listed in Tab. 2.

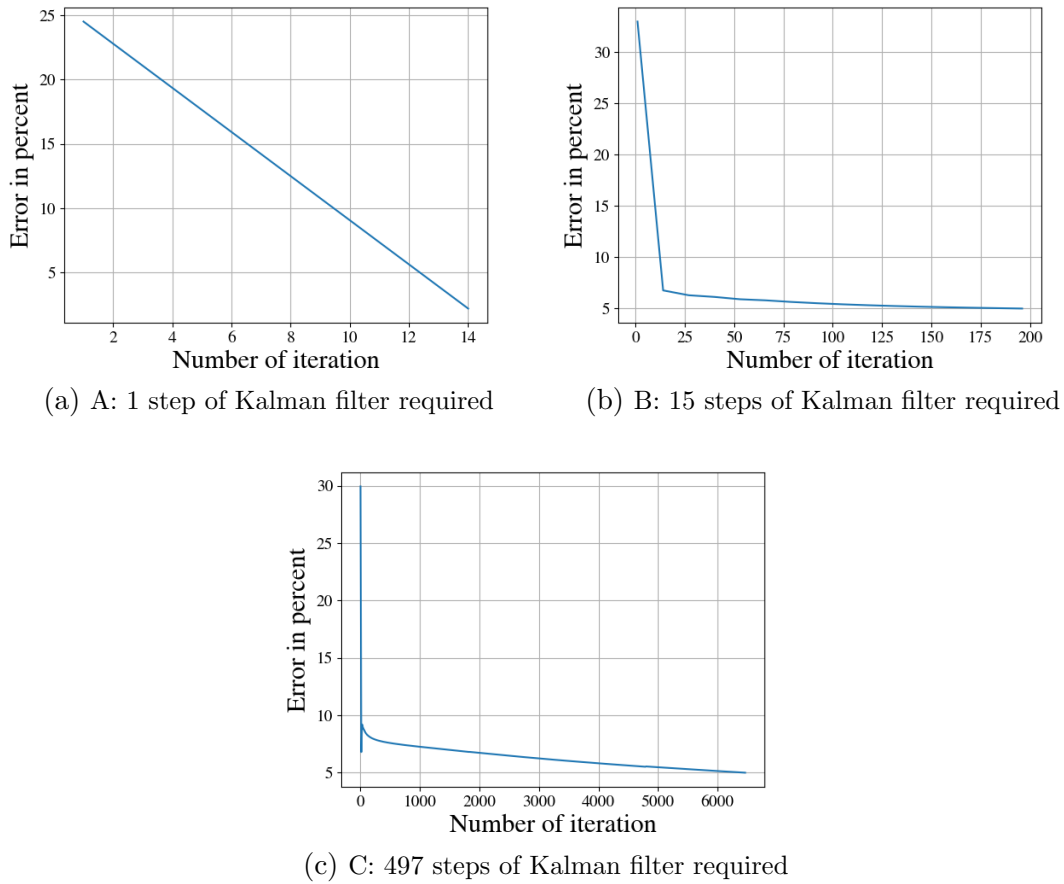


Figure 2. Convergence rate of representative cases

Table 2. Parameters of cases presented in Fig. 2

Varying parameters	Case A	Case B	Case C
SBP, mmHg	121.0	112.3	112.3
DBP, mmHg	72.0	77.6	77.6
SV, ml	66.8	79.9	79.9
HR, beats/min	73.0	63.9	73.0
Diameters of major arteries	Decreased	Decreased	Decreased
LVET, ms	260	306	260

Conclusion

In this paper, we have shown how the generation of synthetic patient cohorts can be facilitated using computer clusters. The process is automatic once the target values are defined. We have generated a synthetic pulse wave database based on a one-dimensional blood flow model of the systemic arterial circulation with a two-chambered lumped heart. This method can be used to study the effect of different cardiovascular pathologies on pulse wave shapes. The introduction of a lumped heart model with leaflet dynamics allows us to simulate databases with valvular disease.

By parallelizing and optimising computations, we can calculate hundreds of virtual patients in 10 days instead of months. In our example, we have simulated pulse waves for a healthy

synthetic population. The introduction of any cardiovascular pathology, such as aortic stenosis, would require a separate synthetic cohort of thousands of virtual patients. Ideally, we need a separate cohort for each degree of pathology and for each combination of pathologies. The amount of computation required for such a task is impossible without high performance optimisations.

We have used parallelization on two levels: parallelization of different branches and junctions of the blood flow model and parallelization of different virtual patients. It is possible to implement another level of parallelization: the distribution of the “forward propagation” of the UKF algorithm. This would allow us to distribute 13 independent tasks to different nodes of the cluster. This was not necessary due to the limitations of our cluster, but it provides opportunities for further optimisation if more computing power is available.

Acknowledgements

This work was financed by the Ministry of Science and Higher Education of the Russian Federation within the framework of state support for the creation and development of World-Class Research Centers “Digital biodesign and personalized healthcare” № 075-15-2022-304.

Abbreviations

The following abbreviations are used in this manuscript:

CPU	Central processing unit
DBP	Diastolic blood pressure
HPC	High performance computing
HR	Heart rate
LVET	Left ventricular ejection time
ROUKF	Reduced-order Unscented Kalman Filter
SD	Standard deviation
SBP	Systolic blood pressure
SV	Stroke volume
UKF	Unscented Kalman Filter

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. Abohtyra, R.M., Vincent, T.L.: Nonlinear parameter and state estimation approach in end-stage kidney disease patients. bioRxiv (2023). <https://doi.org/10.1101/2022.04.02.486844>
2. Allen, R.J., Rieger, T.R., Musante, C.J.: Efficient generation and selection of virtual populations in quantitative systems pharmacology models. CPT Pharmacometrics Syst Pharmacol 5(3), 140–146 (2016). <https://doi.org/10.1002/psp4.12063>
3. Arthurs, C., Xiao, N., Moireau, P., *et al.*: A flexible framework for sequential estimation of model parameters in computational hemodynamics. Advances in Modeling and Simulation

- in Engineering Sciences 7, 48 (2020). <https://doi.org/10.1186/s40323-020-00186-x>
4. Bertoglio, C., Moireau, P., Gerbeau, J.F.: Sequential parameter estimation for fluidstructure problems: Application to hemodynamics. *International Journal for Numerical Methods in Biomedical Engineering* 28(4), 434–455 (2012). <https://doi.org/10.1002/cnm.1476>
 5. Boileau, E., Nithiarasu, P., Blanco, P.J., *et al.*: A benchmark study of numerical schemes for one-dimensional arterial blood flow modelling. *International Journal for Numerical Methods in Biomedical Engineering* 31(10) (2015). <https://doi.org/10.1002/cnm.2732>
 6. Caiazzo, A., Caforio, F., Montecinos, G., *et al.*: Assessment of reduced-order unscented Kalman filter for parameter identification in 1-dimensional blood flow models using experimental data. *International Journal for Numerical Methods in Biomedical Engineering* 33(8), e2843 (2017). <https://doi.org/10.1002/cnm.2843>
 7. Charlton, P.H., Harana, J.M., Vennin, S., *et al.*: Modeling arterial pulse waves in healthy aging: a database for in silico evaluation of hemodynamics and pulse wave indexes. *American Journal of Physiology - Heart and Circulatory Physiology* 317(5), H1062–H1085 (2019). <https://doi.org/10.1152/ajpheart.00218.2019>
 8. Frazier, P.I.: A tutorial on bayesian optimization (2018)
 9. Gamilov, T., Danilov, A., Chomakhidze, P., *et al.*: Computational analysis of hemodynamic indices in multivessel coronary artery disease in the presence of myocardial perfusion dysfunction. *Computation* 12(6) (2024). <https://doi.org/10.3390/computation12060110>
 10. Garay, J., Dunstan, J., Uribe, S., Costabal, F.: Physics-informed neural networks for blood flow inverse problems (08 2023). <https://doi.org/10.48550/arXiv.2308.00927>
 11. Hall, K.: Predicting metabolic adaptation, body weight change, and energy intake in humans. *Am J Physiol Endocrinol Metab* 298(3), E449–66 (2009). <https://doi.org/10.1152/ajpendo.00559.2009>
 12. Hickson, S.S., Butlin, M., Graves, M., *et al.*: The relationship of age with regional aortic stiffness and diameter. *JACC.Cardiovascular Imaging* 3(12), 1247–1255 (2010). <https://doi.org/10.1016/j.jcmg.2010.09.016>
 13. Jeanson, F., Farkouh, M.E., Godoy, L.C., *et al.*: Medical calculators derived synthetic cohorts: a novel method for generating synthetic patient data. *Scientific Reports* 14, 11437 (2024). <https://doi.org/10.1038/s41598-024-61721-z>
 14. Khorchani, T., Gadiya, Y., Witt, G., *et al.*: SASC: A simple approach to synthetic cohorts for generating longitudinal observational patient cohorts from COVID-19 clinical data. *Patterns (N Y)* 3(4), 100453 (2022). <https://doi.org/10.1016/j.patter.2022.100453>
 15. Korakianitis, T., Shi, Y.: A concentrated parameter model for the human cardiovascular system including heart valve dynamics and atrioventricular interaction. *Medical Engineering & Physics* 28(7), 613–628 (2006). <https://doi.org/10.1016/j.medengphy.2005.10.004>
 16. Korakianitis, T., Shi, Y.: Numerical simulation of cardiovascular dynamics with healthy and diseased heart valves. *Journal of Biomechanics* 39(11), 1964–1982 (2006). <https://doi.org/10.1016/j.jbiomech.2005.06.016>

17. Moireau, P., Chapelle, D.: Reduced-order Unscented Kalman Filtering with application to parameter identification in large-dimensional systems. *ESAIM: Control, Optimisation and Calculus of Variations* 17(2), 380–405 (2011). <https://doi.org/10.1051/cocv/2010006>
18. Nolte, D., Bertoglio, C.: Inverse problems in blood flow modeling: A review. *International Journal for Numerical Methods in Biomedical Engineering* 38(8), e3613 (2022). <https://doi.org/10.1002/cnm.3613>
19. Rammah Abohtyra, T.V., Schneditz, D.: Magnitude and precision of absolute blood volume estimated during hemodialysis. *Renal Failure* 46(2), 2377781 (2024). <https://doi.org/10.1080/0886022X.2024.2377781>
20. Simakov, S., Gamilov, T., Soe, Y.N.: Computational study of blood flow in lower extremities under intense physical load. *Russian Journal of Numerical Analysis and Mathematical Modelling* 28(5), 485–504 (2013). <https://doi.org/10.1515/rnam-2013-0027>
21. Simakov, S., Timofeev, A., Gamilov, T., *et al.*: Analysis of the impact of left ventricular assist devices on the systemic circulation. *Russian Journal of Numerical Analysis and Mathematical Modelling* 35(5), 295–314 (2020). <https://doi.org/10.1515/rnam-2020-0025>
22. Stamler, J., Stamler, R., Neaton, J.D.: Blood pressure, systolic, and diastolic, and cardiovascular risks: Us population data. *Archives of Internal Medicine* 153, 598–615 (1993). <https://doi.org/10.1001/archinte.153.5.598>
23. Wan, E.A., Merwe, R.V.D.: The unscented Kalman filter for nonlinear estimation. In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. pp. 153–158. Lake Louise, AB, Canada (2000). <https://doi.org/10.1109/ASSPCC.2000.882463>
24. Wang, T., Jin, W., Liang, F., Alastruey, J.: Machine learning-based pulse wave analysis for early detection of abdominal aortic aneurysms using in silico pulse waves. *Symmetry* 13(5) (2021). <https://doi.org/10.3390/sym13050804>
25. Young, D.F., Tsai, F.Y.: Flow characteristics in models of arterial stenoses. I. Steady flow. *Journal of Biomechanics* 6(4), 395–410 (1973). [https://doi.org/10.1016/0021-9290\(73\)90099-7](https://doi.org/10.1016/0021-9290(73)90099-7)
26. Young, D.F., Tsai, F.Y.: Flow characteristics in models of arterial stenoses. II. Unsteady flow. *Journal of Biomechanics* 6(5), 547–559 (1973). [https://doi.org/10.1016/0021-9290\(73\)90012-2](https://doi.org/10.1016/0021-9290(73)90012-2)