

# Supercomputing Frontiers and Innovations

2024, Vol. 11, No. 3

## 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

<b>AlFaMove: Scalable Implementation of Surface Movement Method for Cluster Computing Systems</b> N.A. Olkhovsky, L.B. Sokolinsky .....	4
<b>Study of the Effectiveness of Parallel Algorithms for Modeling the Dynamics of Collisionless Galactic Systems on GPUs</b> S.S. Khrapov, A.V. Khoperskov .....	27
<b>Investigation of the Capability of Restoring Information on the Primary Particle from Cherenkov Light Generated by Extensive Air Showers Using the Lomonosov-2 Supercomputer</b> E.A. Bonvech, C.G. Azra, O.V. Cherkesova, D.V. Chernov, E.L. Entina, V.I. Galkin, V.A. Ivanov, T.A. Kolodkin, N.O. Ovcharenko, D.A. Podgrudkov, T.M. Roganova, M.D. Ziva .....	45
<b>Quantum-Chemical Study of Some Trispyrazolobenzenes and Trispyrazolo-1,3,5-triazines</b> V.M. Volokhov, V.V. Parakhin, E.S. Amosova, D.B. Lempert, Vl.V. Voevodin .....	64
<b>Wing Noise Simulation of Supersonic Business Jet in Landing Configuration</b> A.P. Duben, T.K. Kozubskaya, P.V. Rodionov .....	74
<b>Tool and Algorithm for the Determination of Aptamers in Nanopore Sequencing Data: AptaLong</b> M.A. Grigoryeva, M.G. Khrenova, M.F. Subach, Vl.V. Voevodin, M.I. Zvereva .....	93
<b>Modeling Microtubule Dynamics on Lomonosov-2 Supercomputer of Moscow State University: from Atomistic to Cellular Scale Simulations</b> N.B. Gudimchuk, V.V. Alexandrova, E.V. Ulyanov, V.A. Fedorov, E.G. Kholina, I.B. Kovalenko .....	107
<b>Numerical Analysis of OECD/NEA HYMERES Project Benchmark Tests Using CABARET-SC1 CFD Code</b> A.A. Kanaev, V.Yu. Glotov .....	117



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.

# AlFaMove: Scalable Implementation of Surface Movement Method for Cluster Computing Systems\*

Nikolay A. Olkhovsky<sup>1</sup> , Leonid B. Sokolinsky<sup>1</sup> 

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

The article presents a numerical implementation of the surface movement method for linear programming. The base of this implementation is the new AlFaMove algorithm, which builds on the surface of a feasible polytope an optimal objective path from an arbitrary boundary point to a point that is a solution to a linear programming problem. The optimal objective path is a path along the faces of the feasible polytope in the direction of maximizing the value of the objective function. To calculate the optimal movement direction, the pseudoprojection operation on a linear manifold is used. The pseudoprojection operation is a generalization of the orthogonal projection and is implemented using an iterative projection-type algorithm. The proposition is proved that, for a linear manifold that is the intersection of hyperplanes, the pseudoprojection coincides with the orthogonal projection. It is also proved that, in the case of a linear manifold, pseudoprojection makes it possible to calculate the movement vector in the direction of maximum increase of the objective function. A parallel implementation of the AlFaMove algorithm is described. The results of computational experiments on a cluster computing system are presented to demonstrate the high scalability of the proposed numerical implementation.

*Keywords:* linear programming, surface movement method, numerical implementation, AlFaMove algorithm, parallel implementation, cluster computing system, scalability evaluation.

## Introduction

The age of big data and Industry 4.0 generated large-scale linear programming (LP) problems including millions of variables and millions of constraints [4, 8, 12, 13]. In many cases, the object of linear programming is problems related to the optimization of non-stationary processes [2]. In non-stationary LP problems, the objective function and/or constraints change during the computational process. Also in this class of problems, there are applications in which it is necessary to perform optimization in real time. Highly scalable methods and parallel algorithms for linear programming are needed to solve such problems.

The simplest approach to solving non-stationary optimization problems is to consider each change as the appearance of a new optimization problem that needs to be solved from scratch [2]. However, this approach is often impractical, because solving a problem from scratch without reusing information from the past can take too long. Thus, it is desirable to have an optimization algorithm capable of continuously adapting the computation process to a changing environment, reusing information obtained in the past. This approach is applicable for real-time processes if the algorithm tracks the trajectory of the optimal point fast enough. In the case of large-scale LP problems, the latter requires the development of scalable methods and parallel algorithms for linear programming.

To date, the most popular methods for solving LP problems are the simplex method [3] and the interior-point methods [20]. These methods are capable of solving problems with tens of thousands of variables and constraints. However, the scalability of parallel algorithms based on the simplex method, in general case, is limited to 16–32 processor nodes [10]. As regards the interior-point algorithms, in general case, they are not amenable to effective parallelization. This

\*The paper is recommended for publication by the Program Committee of the International Scientific Conference “Russian Supercomputing Days 2024”.

<sup>1</sup>South Ural State University (National Research University), Chelyabinsk, Russian Federation

limits the use of these methods for solving large-scale non-stationary LP problems in real time. In accordance with this, the task of developing scalable methods and efficient parallel algorithms for linear programming on cluster computing systems remains urgent.

In recent paper [11] a theoretical description of the new surface movement method for linear programming was presented. This method builds an optimal objective path on the surface of the feasible polytope<sup>2</sup> from an arbitrary boundary point to a solution of the LP problem. The optimal objective path is a path along the faces of the feasible polytope in the direction of maximizing the value of the objective function. Algorithm 1 proposed in this paper, in Step 15, requires finding a point with the maximum value of the objective function on the boundary of a hyperdisk. At the same time, the paper does not provide a numerical algorithm that allows you to perform this step. In this article, we present and evaluate the AlFaMove algorithm, which eliminates the gap. The rest of the paper is organized as follows. Section 1 presents the theoretical background on which the surface movement method and AlFaMove algorithm are based. Section 2 is devoted to the description of the pseudoprojection operation, which allows you to find a movement vector along the optimal objective path for a linear manifold resulting from the intersection of hyperplanes. Section 3 provides a formalized description of the AlFaMove algorithm, which is a numerical implementation of the surface movement method. Section 4 describes a parallel version of the AlFaMove algorithm. Section 5 provides information on the software implementation of the AlFaMove algorithm and the results of experiments on a cluster computing system to evaluate its scalability. Conclusion summarizes the results and provides further research directions.

## 1. Theoretical Background

This section contains the necessary theoretical basis used to describe the AlFaMove algorithm. We consider a LP problem in the following form:

$$\bar{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathbb{R}^n} \{ \langle \mathbf{c}, \mathbf{x} \rangle \mid A\mathbf{x} \leq \mathbf{b} \}, \quad (1)$$

where  $\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $m > 1$ ,  $\mathbf{c} \neq \mathbf{0}$ . Here,  $\langle \cdot, \cdot \rangle$  stands for the dot product of two vectors. We assume that the constraint  $\mathbf{x} \geq \mathbf{0}$  is also included in the matrix inequality  $A\mathbf{x} \leq \mathbf{b}$  in the form of  $-\mathbf{x} \leq \mathbf{0}$ . The linear objective function of the problem (1) has the form

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle.$$

In this case, the vector  $\mathbf{c}$  is the gradient of the objective function  $f(\mathbf{x})$ .

Let  $\mathbf{a}_i \in \mathbb{R}^n$  denote a vector representing the  $i$ th row of the matrix  $A$ . We assume that  $\mathbf{a}_i \neq \mathbf{0}$  for all  $i \in \{1, \dots, m\}$ . Denote by  $\hat{H}_i$  a closed half-space defined by the inequality  $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$ , and by  $H_i$  – the hyperplane bounding it:

$$\hat{H}_i = \{ \mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i \}; \quad (2)$$

$$H_i = \{ \mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i \}. \quad (3)$$

Let us define a feasible polytope

$$M = \bigcap_{i \in \mathcal{P}} \hat{H}_i, \quad (4)$$

<sup>2</sup>The feasible polytope is the feasible region of the LP problem.

representing the feasible region of LP Problem (1). Note that  $M$ , in this case, is a closed convex set. We assume that  $M$  is bounded, and  $M \neq \emptyset$ , i.e., LP Problem (1) has a solution.

Let us define a recessive half-space [17].

**Definition 1.** The half-space  $\hat{H}_i$  is called recessive if

$$\forall \mathbf{x} \in H_i, \forall \lambda > 0 : \mathbf{x} + \lambda \mathbf{c} \notin \hat{H}_i. \quad (5)$$

The geometric meaning of this definition is that a ray outgoing in the direction of the vector  $\mathbf{c}$  from any point of the hyperplane bounding the recessive half-space has no points in common with this half-space, except for the beginning one. It is known [17] that the following condition is necessary and sufficient for the half-space  $\hat{H}_i$  to be recessive:

$$\langle \mathbf{a}_i, \mathbf{c} \rangle > 0.$$

Denote

$$\mathcal{I} = \{i \in \{1, \dots, m\} \mid \langle \mathbf{a}_i, \mathbf{c} \rangle > 0\}, \quad (6)$$

i.e.,  $\mathcal{I}$  represents a set of indexes for which the half-space  $\hat{H}_i$  is recessive. Since the feasible polytope  $M$  is a bounded set, we have

$$\mathcal{I} \neq \emptyset.$$

Define

$$\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i. \quad (7)$$

Obviously,  $\hat{M}$  is a convex, closed, unbounded polytope. We will call it recessive. Let us denote by  $\Gamma(M)$  the set of boundary points of the feasible polytope  $M$ , and by  $\Gamma(\hat{M})$  the set of boundary points of the recessive polytope  $\hat{M}$ <sup>3</sup>. According to Proposition 3 in [17] we have

$$\bar{\mathbf{x}} \in \Gamma(\hat{M}),$$

i.e., a solution to LP problem (1) lies on the boundary of the recessive polytope  $\hat{M}$ .

Following [9], we can define an orthogonal projection onto a hyperplane.

**Definition 2.** Let be given the hyperplane  $H = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}, \mathbf{x} \rangle = b\}$ . The orthogonal projection  $\pi_H(\mathbf{v})$  of a point  $\mathbf{v} \in \mathbb{R}^n$  onto the hyperplane  $H$  is defined by the equation

$$\pi_H(\mathbf{v}) = \mathbf{v} - \frac{\langle \mathbf{a}, \mathbf{v} \rangle - b}{\|\mathbf{a}\|^2} \mathbf{a}. \quad (8)$$

The following proposition provides a way to calculate the optimal path on a hyperplane.

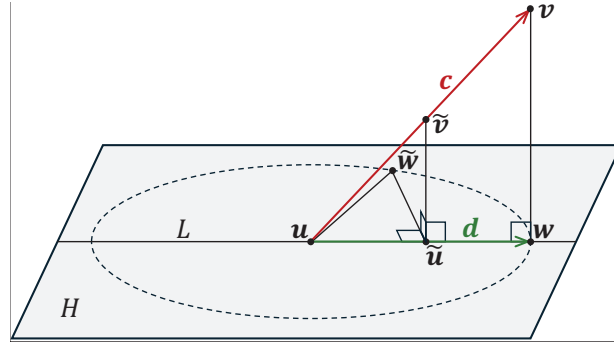
**Proposition 1.** Let be given a hyperplane  $H$  with the normal  $\mathbf{a} \in \mathbb{R}^n$ , which including the point  $\mathbf{u} \in \mathbb{R}^n$ :

$$H = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}, \mathbf{x} \rangle = \langle \mathbf{a}, \mathbf{u} \rangle\}. \quad (9)$$

Let a linear function  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  with gradient  $\mathbf{c} \in \mathbb{R}^n$  be defined:

$$f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle. \quad (10)$$

<sup>3</sup>A boundary point of a set  $\hat{M} \subset \mathbb{R}^n$  is a point in  $\mathbb{R}^n$  for which any open neighborhood of it in  $\mathbb{R}^n$  has a nonempty intersection with both the set  $\hat{M}$  and its complement.



**Figure 1.** Illustration to proof of Proposition 1

(the dashed line denotes the hyper-circle of radius  $\|\mathbf{w} - \mathbf{u}\|$  centered at the point  $\mathbf{u}$ )

Let the vectors  $\mathbf{a}$  and  $\mathbf{c}$  be linearly independent (not collinear, and there is no zero vector among them). Denote

$$\mathbf{v} = \mathbf{u} + \mathbf{c}. \quad (11)$$

Build the orthogonal projection  $\pi_H(\mathbf{v})$  of point  $\mathbf{v}$  onto the hyperplane  $H$ :

$$\mathbf{w} = \pi_H(\mathbf{v}). \quad (12)$$

Then the vector  $\mathbf{d} = \mathbf{w} - \mathbf{u}$  uniquely determines the direction of maximum increase of the linear function  $f(\mathbf{x})$  defined by equation (10).

*Proof.* Assume the opposite is true: there exists a point  $\tilde{\mathbf{w}} \in H$  such that

$$\langle \mathbf{c}, \tilde{\mathbf{w}} \rangle \geq \langle \mathbf{c}, \mathbf{w} \rangle, \quad (13)$$

$\|\tilde{\mathbf{w}} - \mathbf{u}\| = \|\mathbf{w} - \mathbf{u}\|$ , and  $\tilde{\mathbf{w}} \neq \mathbf{w}$  (see Fig. 1). Here and further on,  $\|\cdot\|$  denotes the Euclidean norm. Calculate  $\langle \mathbf{c}, \mathbf{w} \rangle$ . According to the definition 2, the orthogonal projection  $\pi_H(\mathbf{v})$  of point  $\mathbf{v}$  onto the hyperplane  $H$  defined by equation (9) is calculated as follows:

$$\mathbf{w} = \mathbf{v} - \frac{\langle \mathbf{a}, \mathbf{v} - \mathbf{u} \rangle}{\|\mathbf{a}\|^2} \mathbf{a}.$$

Substituting the right side of equation (11) instead of  $\mathbf{v}$ , we obtain

$$\mathbf{w} = \mathbf{u} + \mathbf{c} - \frac{\langle \mathbf{a}, \mathbf{c} \rangle}{\|\mathbf{a}\|^2} \mathbf{a}. \quad (14)$$

Using (14), we figure out

$$\langle \mathbf{c}, \mathbf{w} \rangle = \langle \mathbf{c}, \mathbf{u} \rangle + \|\mathbf{c}\|^2 - \frac{\langle \mathbf{a}, \mathbf{c} \rangle^2}{\|\mathbf{a}\|^2}. \quad (15)$$

Since  $\mathbf{a}$  and  $\mathbf{c}$  are linearly independent, in accordance with the Cauchy–Bunyakovsky–Schwarz inequality we have

$$\langle \mathbf{a}, \mathbf{c} \rangle^2 < \|\mathbf{a}\|^2 \cdot \|\mathbf{c}\|^2.$$

This implies

$$\|\mathbf{c}\|^2 - \frac{\langle \mathbf{a}, \mathbf{c} \rangle^2}{\|\mathbf{a}\|^2} > 0. \quad (16)$$

Now calculate  $\langle \mathbf{c}, \tilde{\mathbf{w}} \rangle$ . Let  $\tilde{\mathbf{u}} = \pi_L(\tilde{\mathbf{w}})$  be the orthogonal projection of point  $\tilde{\mathbf{w}}$  onto the line  $L$  passing through the points  $\mathbf{u}$  and  $\mathbf{w}$ . By construction, there is a number  $\delta$  satisfying the condition

$$-1 \leq \delta < 1 \tag{17}$$

such that

$$\tilde{\mathbf{u}} = \mathbf{u} + \delta(\mathbf{w} - \mathbf{u}).$$

Define

$$\tilde{\mathbf{v}} = \mathbf{u} + \delta(\mathbf{v} - \mathbf{u}). \tag{18}$$

Then, the point  $\tilde{\mathbf{u}}$  is the orthogonal projection of the point  $\tilde{\mathbf{v}}$  onto the hyperplane  $H$  defined by equation (9), and can be calculated as follows:

$$\tilde{\mathbf{u}} = \tilde{\mathbf{v}} - \frac{\langle \mathbf{a}, \tilde{\mathbf{v}} - \mathbf{u} \rangle}{\|\mathbf{a}\|^2} \mathbf{a}.$$

Substituting the right side of equation (18) instead of  $\tilde{\mathbf{v}}$ , we obtain the following equation from here:

$$\tilde{\mathbf{u}} = \mathbf{u} + \delta \left( \mathbf{v} - \mathbf{u} - \frac{\langle \mathbf{a}, \mathbf{v} - \mathbf{u} \rangle}{\|\mathbf{a}\|^2} \mathbf{a} \right).$$

Using (11), we make the replacement  $\mathbf{v} - \mathbf{u} = \mathbf{c}$ :

$$\tilde{\mathbf{u}} = \mathbf{u} + \delta \left( \mathbf{c} - \frac{\langle \mathbf{a}, \mathbf{c} \rangle}{\|\mathbf{a}\|^2} \mathbf{a} \right). \tag{19}$$

Obviously

$$\tilde{\mathbf{w}} = (\tilde{\mathbf{w}} - \tilde{\mathbf{u}}) + \tilde{\mathbf{u}}. \tag{20}$$

Replace the second summand in (20) with the right-hand side of equation (19):

$$\tilde{\mathbf{w}} = (\tilde{\mathbf{w}} - \tilde{\mathbf{u}}) + \mathbf{u} + \delta \left( \mathbf{c} - \frac{\langle \mathbf{a}, \mathbf{c} \rangle}{\|\mathbf{a}\|^2} \mathbf{a} \right).$$

Using (1), we obtain

$$\langle \mathbf{c}, \tilde{\mathbf{w}} \rangle = \langle \mathbf{c}, \tilde{\mathbf{w}} - \tilde{\mathbf{u}} \rangle + \langle \mathbf{c}, \mathbf{u} \rangle + \delta \left( \|\mathbf{c}\|^2 - \frac{\langle \mathbf{a}, \mathbf{c} \rangle^2}{\|\mathbf{a}\|^2} \right).$$

By construction, vector  $\tilde{\mathbf{w}} - \tilde{\mathbf{u}}$  is orthogonal to vector  $\mathbf{v} - \mathbf{u} = \mathbf{c}$ . Therefore,  $\langle \mathbf{c}, \tilde{\mathbf{w}} - \tilde{\mathbf{u}} \rangle = 0$ . Thus, equation (1) is transformed to the form

$$\langle \mathbf{c}, \tilde{\mathbf{w}} \rangle = \langle \mathbf{c}, \mathbf{u} \rangle + \delta \left( \|\mathbf{c}\|^2 - \frac{\langle \mathbf{a}, \mathbf{c} \rangle^2}{\|\mathbf{a}\|^2} \right). \tag{21}$$

Comparing (15) and (21), and taking into account (16) and (17), we obtain

$$\langle \mathbf{c}, \tilde{\mathbf{w}} \rangle < \langle \mathbf{c}, \mathbf{w} \rangle,$$

which contradicts(13). □



Returning to the LP (1) problem, we can say the following. Let  $\mathbf{u} \in M \cap \Gamma(\hat{M})$ , and there is a single recessive hyperplane  $H_{i'}$  ( $i' \in \mathcal{I}$ ) such that  $\mathbf{u} \in H_{i'}$ . In this case, the vector  $\mathbf{d}$ , which determines the direction of the optimal objective path at the point  $\mathbf{u}$ , in accordance with Proposition 1, can be calculated as follows:

$$\mathbf{d} = \mathbf{c} - \frac{\langle \mathbf{a}_{i'}, \mathbf{u} + \mathbf{c} \rangle - b_{i'}}{\|\mathbf{a}_{i'}\|^2} \mathbf{a}_{i'}. \quad (22)$$

In the next section, we consider the case when two or more hyperplanes pass through the point  $\mathbf{u}$ .

## 2. Pseudoprojecting onto Linear Manifold

Let  $\mathcal{J} \subseteq \{1, \dots, m\}$ ,  $\mathcal{J} \neq \emptyset$ , and  $\bigcap_{i \in \mathcal{J}} H_i \neq \emptyset$ . In this case, the set of indices  $\mathcal{J}$  defines a linear manifold  $L$  in the space  $\mathbb{R}^n$ :

$$L = \bigcap_{i \in \mathcal{J}} H_i. \quad (23)$$

Denote by  $k_L$  the dimension of the linear manifold  $L$ . For  $0 < k_L < n - 1$ , the manifold  $L$  is not a hyperplane, and, to determine the movement vector  $\mathbf{d}$  along this manifold in the direction of maximum increase in the objective function value, the equation (22) cannot be used, since such a linear manifold cannot be defined by a single linear equation in the space  $\mathbb{R}^n$ . However, we can find the specified vector  $\mathbf{d}$  using the pseudoprojection operation [17]. Define the projection mapping  $\varphi(\cdot)$ :

$$\varphi(\mathbf{x}) = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \pi_{H_i}(\mathbf{x}). \quad (24)$$

It is known [18] that the mapping  $\varphi(\mathbf{x})$  is a continuous  $L$ -Fejér mapping, and the sequence of points

$$\left\{ \mathbf{x}_k = \varphi^k(\mathbf{x}_0) \right\}_{k=1}^{\infty} \quad (25)$$

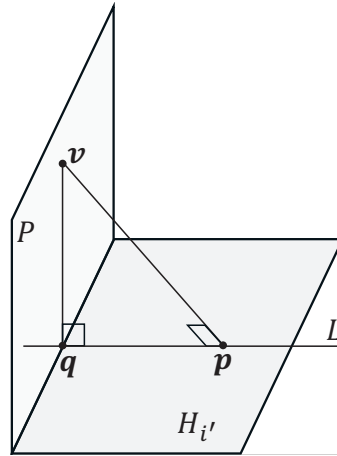
generated by this mapping converges to a point belonging to  $L$ :

$$\mathbf{x}_k \rightarrow \tilde{\mathbf{x}} \in L.$$

Using the mapping  $\varphi(\cdot)$ , let us define the pseudoprojection on a linear manifold formed by the intersection of hyperplanes.

**Definition 3.** Let  $\mathcal{J} \subseteq \{1, \dots, m\}$ ,  $\mathcal{J} \neq \emptyset$ ,  $\bigcap_{i \in \mathcal{J}} H_i \neq \emptyset$ , and  $\varphi(\cdot)$  be the projection mapping defined by equation (24). The pseudoprojection  $\rho_{\mathcal{J}}(\mathbf{x})$  of the point  $\mathbf{x} \in \mathbb{R}^n$  onto the linear manifold  $L = \bigcap_{i \in \mathcal{J}} H_i$  is the limit point of the sequence (25):

$$\lim_{k \rightarrow \infty} \left\| \rho_{\mathcal{J}}(\mathbf{x}) - \varphi^k(\mathbf{x}) \right\| = 0.$$



**Figure 2.** Illustration to proof of Lemma 1

An important feature of the pseudoprojection onto a linear manifold is that the pseudoprojection coincides with the orthogonal projection in this case. To prove this fact, we need the following lemma.

**Lemma 1.** Let the hyperplane  $H_{i'}$  and the linear manifold  $L$  belonging to this hyperplane be given in the space  $\mathbb{R}^n$ :

$$\begin{aligned} H_{i'} &= \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_{i'}, \mathbf{x} \rangle = b_{i'}\}; \\ L &= \bigcap_{i \in \mathcal{J}} H_i; \\ i' &\in \mathcal{J}. \end{aligned}$$

Denote by  $P$  the linear manifold that is the orthogonal complement to  $L$ :

$$P = L^\perp. \quad (26)$$

Then for any point  $\mathbf{v}$  belonging to the linear manifold  $P$ , its orthogonal projection  $\pi_{H_{i'}}(\mathbf{v})$  onto the hyperplane  $H_{i'}$  also belongs to the linear manifold  $P$ :

$$\forall \mathbf{v} \in P : \pi_{H_{i'}}(\mathbf{v}) \in P.$$

*Proof.* Denote by  $\mathbf{p}$  the orthogonal projection of the point  $\mathbf{v}$  onto the hyperplane  $H_{i'}$ :

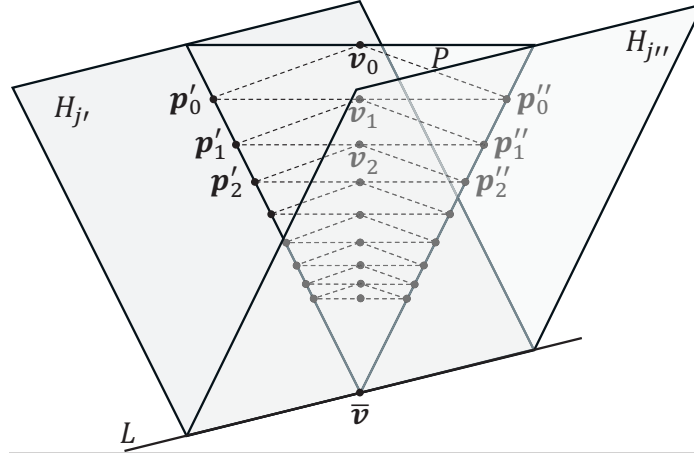
$$\mathbf{p} = \pi_{H_{i'}}(\mathbf{v}). \quad (27)$$

Without loss of generality, we can assume that  $\mathbf{p} \in L$  (see Fig. 2). Suppose that the point  $\mathbf{p}$  does not belong to the linear manifold  $P$ . Let us by  $\mathbf{q}$  denote the intersection point of a linear manifold  $L$  with its orthogonal complement  $P$ :

$$\mathbf{q} = L \cap P.$$

Since  $P$  is the orthogonal complement to the linear manifold  $L$ , the point  $\mathbf{q}$  is the orthogonal projection of the point  $\mathbf{p}$  onto the linear manifold  $P$ :

$$\mathbf{q} = \pi_P(\mathbf{p}). \quad (28)$$



**Figure 3.** Illustration to proof of Proposition 2 for  $n = 3$

Consider the triangle  $\Delta(\mathbf{v}, \mathbf{p}, \mathbf{q})$ . By virtue of (27), the angle  $\angle \mathbf{p}$  with the vertex at the point  $\mathbf{p}$  is right. But this is only possible if  $\mathbf{p} = \mathbf{q}$ , that is  $\mathbf{p} \in P$ .  $\square$

The following proposition proves that a pseudoprojection on a linear manifold coincides with an orthogonal projection.

**Proposition 2.** Let the following conditions hold:

$$\mathcal{J} \subseteq \{1, \dots, m\}, \tag{29}$$

$$L = \bigcap_{i \in \mathcal{J}} H_i, \quad L \neq \emptyset; \tag{30}$$

where  $H_i = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i\}$ . Denote by  $\pi_L(\mathbf{x})$  the orthogonal projection of the point  $\mathbf{x} \in \mathbb{R}^n$  onto the linear manifold  $L$ . Then,

$$\rho_L(\mathbf{x}) = \pi_L(\mathbf{x}),$$

i.e., the pseudoprojection onto the linear manifold  $L$  coincides with the orthogonal projection.

*Proof.* Fix an arbitrary point  $\mathbf{v}_0 \in \mathbb{R}^n$ . Consider the linear manifold  $P$  containing the point  $\mathbf{v}_0$  and being the orthogonal complement to the linear manifold  $L$ :

$$\mathbf{v}_0 \in P = L^\perp. \tag{31}$$

Denote by  $\bar{\mathbf{v}}$  the intersection point of the linear manifold  $L$  with its orthogonal complement  $P$ :

$$L \cap P = \{\bar{\mathbf{v}}\}$$

(see Fig. 3). Make the orthogonal projection of the point  $\mathbf{v}_0$  onto the hyperplane  $H_j$  for an arbitrary  $j \in \mathcal{J}$ :

$$\mathbf{p}_0 = \pi_{H_j}(\mathbf{v}_0).$$

According to Lemma 1, we have

$$\pi_{H_j}(\mathbf{v}_0) \in P.$$

It follows from this and from (24) that

$$\mathbf{v}_1 = \varphi(\mathbf{v}_0) \in P.$$

This means that the sequence

$$\left\{ \mathbf{v}_k = \varphi^k(\mathbf{v}_0) \right\}_{k=1}^{\infty}$$

converges to the point  $\bar{\mathbf{v}}$  of the intersection of the linear manifold  $L$  with the linear manifold  $P$ , i.e.,  $\rho_L(\mathbf{v}_0) = \bar{\mathbf{v}}$ . On the other hand, by virtue of (31), we have  $\pi_L(\mathbf{v}_0) = \bar{\mathbf{v}}$ . Therefore,

$$\forall \mathbf{x} \in \mathbb{R}^n : \rho_L(\mathbf{x}) = \pi_L(\mathbf{x}).$$

The proposition is proven. □

The procedure for approximate computation of a pseudoprojection on a linear manifold is presented in the form of Algorithm 1. Let us briefly comment on the steps of this algorithm.

---

**Algorithm 1** Computing of pseudoprojection  $\rho_{\mathcal{J}}(\mathbf{x})$

---

**Require:**  $H_i = \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i\}$ ;  $\mathcal{J} \subseteq \{1, \dots, m\}$ ;  $\mathcal{J} \neq \emptyset$ ;  $\bigcap_{i \in \mathcal{J}} H_i \neq \emptyset$

```

1: function  $\rho_{\mathcal{J}}(\mathbf{x})$ 
2:    $k := 0$ 
3:    $\mathbf{x}_0 := \mathbf{x}$ 
4:   repeat
5:      $\Sigma := 0$ 
6:     for  $i \in \mathcal{J}$  do
7:        $\Sigma := \Sigma + (\langle \mathbf{a}_i, \mathbf{x}_k \rangle - b_i) \mathbf{a}_i / \|\mathbf{a}_i\|^2$ 
8:     end for
9:      $\mathbf{x}_{(k+1)} := \mathbf{x}_k - \Sigma / |\mathcal{J}|$ 
10:     $\xi_{max} := 0$  ▷ Maximum residual
11:    for  $i \in \mathcal{J}$  do
12:       $\xi_i := \|\langle \mathbf{a}_i, \mathbf{x}_{k+1} \rangle - b_i\|$ 
13:      if  $\xi_i > \xi_{max}$  then
14:         $\xi_{max} := \xi_i$ 
15:      end if
16:    end for
17:     $k := k + 1$ 
18:  until  $\xi_{max} < \epsilon_{\xi}$  ▷ Small parameter  $\epsilon_{\xi} > 0$ 
19:  return  $\mathbf{x}_k$ 
20: end function

```

---

Step 2 sets the iteration counter  $k$  to zero. Step 3 sets the initial approximation  $\mathbf{x}_0$ . Step 4 begins the iterative loop of calculating the pseudoprojection. Steps 5–8 calculate the sum from the right side of equation (24) with the current approximation  $\mathbf{x}_k$ . Step 9 finds the next approximation  $\mathbf{x}_{k+1}$ . Steps 10–16 calculate the maximum residual  $\xi$  for the next approximation with respect to all hyperplanes  $H_i$  involved in the computation. Step 17 increases the iteration counter by one. Step 19 returns the new approximation  $\mathbf{x}_k$  as a result.

---

**Algorithm 2** Calculation of movement vector  $\bar{\mathbf{d}} = \mathbf{D}(\mathbf{u})$

---

**Require:**  $H_i = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i\}$ ;  $\mathbf{u} \in \Gamma(M)$

```

1: function  $\mathbf{D}(\mathbf{u})$ 
2:    $\mathcal{U} := \emptyset$  ▷  $\mathcal{U}$  – set of indices of hyperplanes  $H_i$  passing through point  $\mathbf{u}$ 
3:   for  $i = 1 \dots m$  do
4:     if  $\langle \mathbf{a}_i, \mathbf{u} \rangle = b_i$  then
5:        $\mathcal{U} := \mathcal{U} \cup \{i\}$ 
6:     end if
7:   end for
8:    $\bar{\mathbf{d}} := \mathbf{0}$ 
9:    $f := -\infty$  ▷  $f$  – value of objective function  $f(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle$ 
10:   $\mathbf{e}_c := \mathbf{c} / \|\mathbf{c}\|$ 
11:   $\mathbf{v} := \mathbf{u} + \delta \mathbf{e}_c$  ▷ Large parameter  $\delta > 0$ 
12:  for  $\mathcal{J} \in \mathcal{P}(\mathcal{U}) \setminus \emptyset$  do ▷  $\mathcal{P}(\mathcal{U})$  – set of all subsets of the set  $\mathcal{U}$ 
13:     $\mathbf{w} := \rho_{\mathcal{J}}(\mathbf{v})$ 
14:     $\mathbf{d} := \mathbf{w} - \mathbf{u}$ 
15:     $\mathbf{e}_d := \mathbf{d} / \|\mathbf{d}\|$ 
16:    if  $(\mathbf{u} + \tau \mathbf{e}_d) \in M$  then ▷ Small parameter  $\tau > 0$ 
17:      if  $\langle \mathbf{c}, \mathbf{u} + \tau \mathbf{e}_d \rangle > f$  then
18:         $f := \langle \mathbf{c}, \mathbf{u} + \tau \mathbf{e}_d \rangle$ 
19:         $\bar{\mathbf{d}} := \mathbf{d}$ 
20:      end if
21:    end if
22:  end for
23:  return  $\bar{\mathbf{d}}$ 
24: end function

```

---

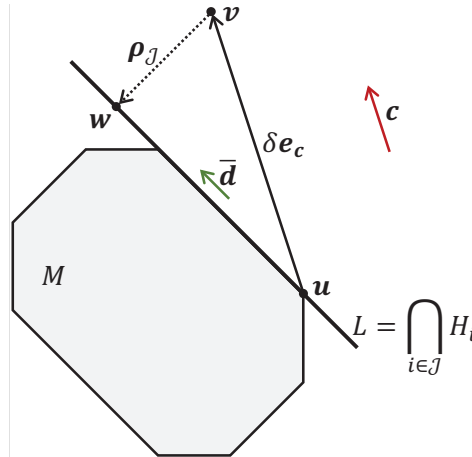
### 3. AlFaMove – Along Faces Movement Algorithm

In this section, we describe the new AlFaMove (Along Faces Movement) algorithm, which is a numerical implementation of the surface movement method [11]. The AlFaMove algorithm builds a path on the surface of the feasible polytope from an arbitrary boundary point  $\mathbf{u}_0 \in M \cap \Gamma(\hat{M})$  to a point  $\bar{\mathbf{x}}$  that is a solution to LP problem (1). Moving along the faces of an feasible polytope is performed in the direction of maximizing the value of the objective function. The path built as a result of such movement is called the optimal objective path.

The basis of the AlFaMove algorithm is the procedure  $\mathbf{D}(\mathbf{u})$ , which calculates at the boundary point  $\mathbf{u}$  the movement vector  $\bar{\mathbf{d}}$  along the face of the feasible polytope  $M$  in the direction of maximum increase in the value of the objective function. Procedure  $\mathbf{D}(\mathbf{u})$  is presented in the form of Algorithm 2. Geometrical representation of the operation of this algorithm is shown in Fig. 4. Let us briefly comment on the steps of Algorithm 2. Steps 2–7 construct the set  $\mathcal{U}$ , which includes the indices of all hyperplanes  $H_i$  passing through the point  $\mathbf{u}$ . Step 8 resets the direction vector  $\bar{\mathbf{d}}$ . In Step 9, the infinitesimal number<sup>4</sup> is assigned to the variable  $f$ , which stores the value of the objective function. Step 10 calculates the unit vector  $\mathbf{e}_c$  parallel to the vector  $\mathbf{c}$ . In

---

<sup>4</sup>In the case of double-precision floating-point format that occupies 64 bits in computer memory, the infinitesimal number is  $-1 \cdot 10^{308}$ .



**Figure 4.** Geometrical representation of Algorithm 2

Step 11, the point  $\mathbf{v}$  is constructed by adding the vector  $\delta \mathbf{e}_c$  to the vector  $\mathbf{u}$  (see Fig. 4). Here,  $\delta$  is a “large” positive parameter: the greater the  $\delta$ , the more accurately the direction vector  $\bar{\mathbf{d}}$  will be calculated. However, when the  $\delta$  parameter is increased, the time for calculating the pseudoprojection  $\rho_{\mathcal{J}}(\mathbf{v})$  in Step 13 will also increase. The **for** loop in Step 12 iterates through all possible combinations of indices of the hyperplanes passing through the point  $\mathbf{u}$ . Each such combination  $\mathcal{J}$  corresponds to the linear manifold  $L = \bigcap_{i \in \mathcal{J}} H_i$ , which also passes through the point  $\mathbf{u}$ . Step 13 calculates the point  $\mathbf{w}$  by pseudoprojecting point  $\mathbf{v}$  onto the linear manifold corresponding to the combination  $\mathcal{J}$ . Step 14 calculates, for the current linear manifold, the vector  $\mathbf{d}$ , which determines the direction of maximum increase in the values of the objective function. Step 15 calculates the unit vector  $\mathbf{e}_d$  parallel to the vector  $\mathbf{d}$ . Step 16 checks that the small movement from point  $\mathbf{u}$  in the direction  $\mathbf{d}$  does not exceed the boundaries of the feasible polytope. Step 17, in turn, checks if the value of the objective function at the point  $(\mathbf{u} + \mathbf{e}_d)$  is greater than the maximum value obtained in previous iterations of the **for** loop. If so, then the last value is stored as the maximum (Step 18), and the last direction is assigned to vector  $\bar{\mathbf{d}}$  (Step 19). After all possible combinations have been checked, the vector  $\bar{\mathbf{d}}$  is returned as a result (Step 23). If none of the combinations passed the check in steps 16–17, the zero vector will be returned as a result. This means that any movement from point  $\mathbf{u}$  along the surface of the feasible polytope does not lead to an increase in the value of the objective function.

Now, everything is ready to describe the AlFaMove algorithm that solves the LP problem (1). The Algorithm 1 from the paper [11] will serve as a basis for us. The implementation of the AlFaMove algorithm in pseudocode is presented in the form of Algorithm 3. Let us comment on the steps of this algorithm. Step 1 reads the initial approximation  $\mathbf{u}_0$ . This can be an arbitrary boundary point of the recessive polytope  $\hat{M}$ , satisfying the following condition:

$$\mathbf{u}_0 \in M \cap \Gamma(\hat{M}).$$

This condition is checked in Step 2. To obtain a suitable initial approximation, an algorithm can be used that implements the Quest stage of the apex method [17]. Step 3 calculates the initial movement vector  $\mathbf{d}_0$ . To do this, the function  $\mathbf{D}(\cdot)$  is used, implemented in the Algorithm 2. It is assumed that  $\mathbf{d}_0 \neq \mathbf{0}$ <sup>5</sup>. This condition is controlled in Step 4. Step 5 sets the iteration counter  $k$  to zero. Step 6 begins the **repeat/until** loop, which performs movement along the

<sup>5</sup>The equality of the vector  $\mathbf{d}_0$  to the zero vector means that the point  $\mathbf{u}_0$  is a solution to LP problem (1).

---

**Algorithm 3** AlFaMove

---

**Require:**  $\hat{H}_i = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i\}$ ;  $M = \bigcap_{i=1}^m \hat{H}_i$ ;  $\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i$ ;  $i \in \mathcal{I} \Leftrightarrow \langle \mathbf{a}_i, \mathbf{c} \rangle > 0$

```

1: input  $\mathbf{u}_0$ 
2: assert  $\mathbf{u}_0 \in M \cap \Gamma(\hat{M})$ 
3:  $\mathbf{d}_0 := \mathbf{D}(\mathbf{u}_0)$ 
4: assert  $\mathbf{d}_0 \neq \mathbf{0}$ 
5:  $k := 0$ 
6: repeat
7:    $\mathbf{u}_{k+1} := \boldsymbol{\mu}(\mathbf{u}_k, \mathbf{d}_k)$ 
8:    $\mathbf{d}_{k+1} := \mathbf{D}(\mathbf{u}_{k+1})$ 
9:    $k := k + 1$ 
10: until  $\mathbf{d}_k = \mathbf{0}$ 
11: output  $\mathbf{u}_k$  ▷ Solution to LP problem (1)
12: stop

```

---

faces of the feasible polytope until the movement vector  $\mathbf{d}_k$  becomes equal to the zero vector. In this case, the last approximation  $\mathbf{u}_k$  is a solution to LP problem (1). Step 7 calculates the next approximation  $\mathbf{u}_{k+1}$  using the vector function  $\boldsymbol{\mu}$ , the definition of which will be given below. Step 8 calculates the movement vector  $\mathbf{d}_{k+1}$  for the next approximation  $\mathbf{u}_{k+1}$ . Step 9 increases the iteration counter  $k$  by one. If the last movement vector is equal to the zero vector, then the **repeat/until** loop is terminated at Step 10, after that, Step 11 outputs the coordinates of the point  $\mathbf{u}_k$  as a solution to the LP problem (1). Step 12 terminates the AlFaMove algorithm.

The vector function  $\boldsymbol{\mu}(\cdot)$  used in Step 7 of Algorithm 3 is defined as follows. Denote

$$\mathcal{Q} = \{i \in \{1, \dots, m\} \mid \langle \mathbf{a}_i, \mathbf{u} \rangle < b_i \wedge \langle \mathbf{a}_i, \mathbf{d} \rangle > 0\}. \quad (32)$$

Then

$$\boldsymbol{\mu}(\mathbf{u}, \mathbf{d}) = \arg \min_{i \in \mathcal{Q}} \{\|\mathbf{u} - \mathbf{x}\| \mid \mathbf{x} = \gamma_i(\mathbf{u}, \mathbf{d})\}. \quad (33)$$

Here,  $\gamma_i(\mathbf{u}, \mathbf{d})$  denotes a vector function that calculates the oblique projection of the point  $\mathbf{u}$  onto the hyperplane  $H_i$  relative to the vector  $\mathbf{d}$ :

$$\gamma_i(\mathbf{u}, \mathbf{d}) = \mathbf{u} - \frac{\langle \mathbf{a}_i, \mathbf{u} \rangle - b_i}{\langle \mathbf{a}_i, \mathbf{d} \rangle} \mathbf{d}.$$

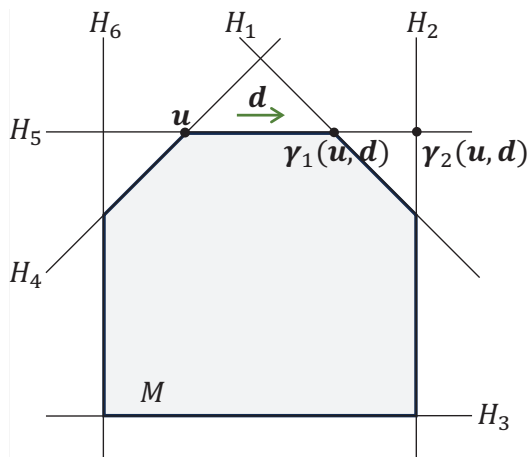
Figure 5 illustrates the action of the function  $\boldsymbol{\mu}$ . The figure suggests that the hyperplanes  $H_4$  and  $H_5$  do not satisfy the inequality  $\langle \mathbf{a}_i, \mathbf{u} \rangle < b_i$  in equation (32). Hyperplanes  $H_3$  and  $H_6$  do not satisfy the inequality  $\langle \mathbf{a}_i, \mathbf{d} \rangle > 0$  in equation (32). Thus,  $\mathcal{Q} = \{1, 2\}$ . Since

$$\|\mathbf{u} - \gamma_1(\mathbf{u}, \mathbf{d})\| < \|\mathbf{u} - \gamma_2(\mathbf{u}, \mathbf{d})\|,$$

then  $\boldsymbol{\mu}(\mathbf{u}, \mathbf{d}) = \gamma_1(\mathbf{u}, \mathbf{d})$ .

The following theorem ensures the convergence of Algorithm 3 to a solution of LP problem (1) in a finite number of iterations.

**Theorem 1.** (Convergence of AlFaMove algorithm) Let the feasible polytope  $M$  of LP problem (1) be a bounded nonempty set. Let  $\bar{\mathbf{x}}$  be a solution to LP problem (1). Then, the sequence



**Figure 5.** Action of function  $\mu$ :  
 $\mu(\mathbf{u}, \mathbf{d}) = \gamma_1(\mathbf{u}, \mathbf{d})$

of approximations  $\{\mathbf{u}_k\}_{k=1}^K$  generated by Algorithm 3, is finite ( $K < +\infty$ ), and,  $\langle \mathbf{c}, \mathbf{u}_K \rangle = \langle \mathbf{c}, \bar{\mathbf{x}} \rangle$ , i.e.,  $\mathbf{u}_K$  is a solution to LP problem (1).

*Proof.* Denote by  $\mathbf{d}_{AlFaMove}$  the vector  $\mathbf{d}_{k+1}$ , calculated in step 8 of Algorithm 3. In accordance with steps 13 and 14 of Algorithm 2, the following equation holds:

$$\mathbf{d}_{AlFaMove} = \boldsymbol{\rho}_{\mathcal{J}}(\mathbf{v}) - \mathbf{u}.$$

According to Proposition 2, it follows that

$$\mathbf{d}_{AlFaMove} = \boldsymbol{\pi}_{\mathcal{J}}(\mathbf{v}) - \mathbf{u}, \quad (34)$$

where  $\boldsymbol{\pi}_{\mathcal{J}}(\mathbf{v})$  denotes the orthogonal projection of the point  $\mathbf{v}$  onto the linear manifold  $L = \bigcap_{i \in \mathcal{J}} H_i$ . According to Proposition 1, this means that the vector  $\mathbf{d}_{AlFaMove}$  uniquely determines the direction of maximum increase in the objective function value of LP problem (1). And this, in turn, means that Algorithm 3 is a numerical implementation of the surface movement method [11], i.e., the approximation sequences of  $\{\mathbf{u}_{AlFaMove}^k\}$  and  $\{\mathbf{u}_{SMM}^k\}$ , generated respectively by Algorithm 3 from this article and Algorithm 1 from [11], coincide. Thus, the convergence of the AlFaMove algorithm directly follows from the convergence of the surface movement algorithm, ensured by Theorem 1 from article [11].  $\square$

## 4. Parallel Version of AlFaMove Algorithm

The most compute-intensive operation in the AlFaMove algorithm (Algorithm 3) is the operation  $\mathbf{D}(\cdot)$ , which calculates the direction vector at Step 8 in the **repeat/until** loop. When solving large-scale LP problems, it takes more than 90% of the processor time. This is explained by the fact that the vector function  $\mathbf{D}(\cdot)$ , implemented as Algorithm 2, uses, at Step 13, the pseudoprojection operation  $\boldsymbol{\rho}_{\mathcal{J}}(\cdot)$ <sup>6</sup>, which repeatedly applies the mapping  $\varphi(\cdot)$  defined by equation (24) to the starting point  $\mathbf{v}$ . This iterative method uses the orthogonal projection of a point onto a hyperplane as an elementary operation and belongs to the class of projection methods. It is known that in the case of large-scale LP problems, the projection method may require

<sup>6</sup>See Definition 3 and Algorithm 1.



significant time costs [6]. In addition, it should be noted that Algorithm 2 at Step 12 iterates through all non-empty subsets of the set  $\mathcal{U}$ , which includes the indices of hyperplanes passing through the point  $\mathbf{u}$ . For example, if 30 hyperplanes pass through a point, then we will have  $2^{30} - 1 = 1\,073\,741\,823$  non-empty subsets. To iterate through such a number of subsets, we will need the power of a supercomputer. Therefore, we have developed a parallel version of the AlFaMove algorithm, presented as Algorithm 4. Parallel Algorithm 4 is based on the BSF par-

---

**Algorithm 4** Parallel version of AlFaMove algorithm

---

<b>master</b>	<b><i>l</i>th worker (<math>l = 0, \dots, L - 1</math>)</b>
<pre> 1: <b>input</b> <math>n, m, A, \mathbf{b}, \mathbf{u}_0</math> 2: <math>k := 0</math> 3: <b>repeat</b> 4:   <b>Broadcast</b> <math>\mathbf{u}_k</math> 5: 6: 7: 8: 9: 10: 11: 12: 13: 14: 15: 16:   <b>Gather</b> <math>\mathcal{L}_{reduce}</math> 17:   <math>(\mathbf{d}_k, f_k) := Reduce(\oplus, \mathcal{L}_{reduce})</math> 18:   <b>if</b> <math>\mathbf{d}_k = \mathbf{0}</math> <b>then</b> 19:     <math>exit := \mathbf{true}</math> 20:   <b>else</b> 21:     <math>\mathbf{u}_{k+1} := \mu(\mathbf{u}_k, \mathbf{d}_k)</math> 22:     <math>k := k + 1</math> 23:     <math>exit := \mathbf{false}</math> 24:   <b>end if</b> 25:   <b>Broadcast</b> <math>exit</math> 26: <b>until</b> <math>exit</math> 27: <b>output</b> <math>\mathbf{u}_k, f_k</math> 28: <b>stop</b> </pre>	<pre> 1: <b>input</b> <math>n, m, A, \mathbf{b}, \mathbf{c}</math> 2: 3: <b>repeat</b> 4:   <b>RecvFromMaster</b> <math>\mathbf{u}_k</math> 5:   <math>\mathcal{U} := []</math> 6:   <b>for</b> <math>i = 1 \dots m</math> <b>do</b> 7:     <b>if</b> <math>\langle \mathbf{a}_i, \mathbf{u}_k \rangle = b_i</math> <b>then</b> 8:       <math>\mathcal{U} := \mathcal{U} \uplus [i]</math> 9:     <b>end if</b> 10:  <b>end for</b> 11:  <math>K := 2^{ \mathcal{U} } - 1</math> 12:  <math>L := \text{NumberOfWorkers}</math> 13:  <math>\mathcal{L}_{map(l)} := [lK/L, \dots, (l+1)K/L - 1]</math> 14:  <math>\mathcal{L}_{reduce(l)} := Map(\mathbb{F}_{\mathbf{u}_k}, \mathcal{L}_{map(l)})</math> 15:  <math>(\mathbf{d}_l, f_l) := Reduce(\oplus, \mathcal{L}_{reduce(l)})</math> 16:  <b>SendToMaster</b> <math>(\mathbf{d}_l, f_l)</math> 17: 18: 19: 20: 21: 22: 23: 24: 25:  <b>RecvFromMaster</b> <math>exit</math> 26: <b>until</b> <math>exit</math> 27: 28: <b>stop</b> </pre>

---

allel computation model [14] designed for cluster computing systems. The BSF model uses the master–worker parallelization scheme and requires the representation of the algorithm in the form of operations on lists using higher-order functions *Map* and *Reduce*. In Algorithm 4, the higher-order function *Map* takes, as the second parameter, the list  $\mathcal{L}_{map} = [1, \dots, K]$  containing the ordinal numbers of all subsets of the set  $\mathcal{U}$ , with the exception of the empty set. Here,

$K = 2^{|\mathcal{U}|} - 1$ . As the first parameter,  $Map$  takes the parameterized function

$$F_{\mathbf{u}} : \{1, \dots, K\} \rightarrow \mathbb{R}^n \times \mathbb{R},$$

which is defined as follows:

$$\begin{aligned} F_{\mathbf{u}}(j) &= (\mathbf{d}_j, f_j); \\ \mathbf{d}_j &= \begin{cases} \mathbf{e}_d, & \text{if } (\mathbf{u} + \tau \mathbf{e}_d) \in M \wedge \langle \mathbf{c}, \mathbf{w} \rangle > \langle \mathbf{c}, \mathbf{u} \rangle; \\ \mathbf{0}, & \text{if } (\mathbf{u} + \tau \mathbf{e}_d) \notin M \vee \langle \mathbf{c}, \mathbf{w} \rangle \leq \langle \mathbf{c}, \mathbf{u} \rangle; \end{cases} \\ f_j &= \begin{cases} \langle \mathbf{c}, \mathbf{u} + \mathbf{e}_d \rangle, & \text{if } (\mathbf{u} + \tau \mathbf{e}_d) \in M \wedge \langle \mathbf{c}, \mathbf{w} \rangle > \langle \mathbf{c}, \mathbf{u} \rangle; \\ -\infty, & \text{if } (\mathbf{u} + \tau \mathbf{e}_d) \notin M \vee \langle \mathbf{c}, \mathbf{w} \rangle \leq \langle \mathbf{c}, \mathbf{u} \rangle, \end{cases} \end{aligned} \quad (35)$$

where

$$\mathbf{w} = \rho_{\sigma(j)}(\mathbf{u} + \delta \mathbf{c} / \|\mathbf{c}\|), \quad (36)$$

and

$$\mathbf{e}_d = \frac{\mathbf{w} - \mathbf{u}}{\|\mathbf{w} - \mathbf{u}\|}.$$

The semantics of the function  $F_{\mathbf{u}}(\cdot)$  is uniquely determined by Algorithm 2. The function  $\sigma(\cdot)$  used in equation (36) maps the natural number  $j \in \{1, \dots, K\}$  to the  $j$ th subset of the set that includes all the elements of the list  $\mathcal{U}$ . To do this, the number  $j$  is converted to a binary representation consisting of  $|\mathcal{U}|$  bits. Each bit corresponds to the hyperplane index from the list  $\mathcal{U}$  in natural order. If the bit contains 1, then the corresponding index is included in the subset  $\sigma(j)$ . If the bit contains 0, then the corresponding index is not included. For example, let the hyperplanes  $H_2, H_4, H_7, H_9$  pass through the point  $\mathbf{u}$ . In this case,  $\mathcal{U} = [2, 4, 7, 9]$  and  $K = 2^4 - 1 = 15$ , i.e., 15 different non-empty subsets can be formed from the set of elements of the list  $\mathcal{U}$ . For instance, let us find the fifth subset. The function  $\sigma(\cdot)$  converts the number 5 into the binary representation of 4 bits 0101 and returns the subset  $\{4, 9\}$  as a result. In such a way, the higher-order function  $Map(F_{\mathbf{u}}, \mathcal{L}_{map})$  converts the list  $\mathcal{L}_{map}$  of ordinal numbers of subsets into the list of pairs  $(\mathbf{d}_j, f_j)$ :

$$Map(F_{\mathbf{u}}, \mathcal{L}_{map}) = [F_{\mathbf{u}}(1), \dots, F_{\mathbf{u}}(K)] = [(\mathbf{d}_1, f_1), \dots, (\mathbf{d}_K, f_K)].$$

Here,  $\mathbf{d}_j$  ( $j = 1, \dots, K$ ) is the movement unit vector, and  $f_j$  is the value of the objective function, which is reached at the point  $\mathbf{u} + \mathbf{d}_j$ .

Denote by  $\mathcal{L}_{reduce}$  the list of pairs generated by the higher-order function  $Map$ :

$$\mathcal{L}_{reduce} = Map(F_{\mathbf{u}}, \mathcal{L}_{map}) = [(\mathbf{d}_1, f_1), \dots, (\mathbf{d}_K, f_K)].$$

Define the binary associative operation

$$\oplus : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n \times \mathbb{R},$$

which is the first parameter of the higher-order function  $Reduce$ :

$$(\mathbf{d}', f') \oplus (\mathbf{d}'', f'') = \begin{cases} (\mathbf{d}', f'), & \text{if } f' \geq f''; \\ (\mathbf{d}'', f''), & \text{if } f' < f''. \end{cases} \quad (37)$$

Higher-order function *Reduce* reduces the list  $\mathcal{L}_{reduce}$  to a single pair by sequentially applying the operation  $\oplus$  to all elements of the list:

$$Reduce(\oplus, \mathcal{L}_{reduce}) = (\mathbf{d}_1, f_1) \oplus \dots \oplus (\mathbf{d}_K, f_K) = (\mathbf{d}_{j'}, f_{j'}),$$

where, according to (37)

$$j' = \arg \max_{1 \leq j \leq K} f_j.$$

Parallel Algorithm 4 uses the master–worker approach and includes  $L + 1$  process: one process is the master and  $L$  processes are the workers. The master process performs general computing management, distributes work between worker processes, receives results from them and generates the final result. For simplicity, we assume that the subset number  $K$  is a multiple of the number of workers  $L$ . In Step 1, the master reads the initial data of the LP problem and the coordinates of the starting point  $\mathbf{u}_0$ . In step 2, the master sets the iteration counter  $k$  to zero. Steps 3–26 implement the main loop **repeat/until** calculating the solution to LP problem (1). In Step 4, the master broadcasts the current approximation  $\mathbf{u}_k$  to all workers. In Step 16, the master receives from the workers the partial results, which are reduced to the single pair  $(\mathbf{d}_k, f_k)$  in Step 17. If the condition  $\mathbf{d}_k = \mathbf{0}$  is met in Step 18, then a solution is found (we assume that  $\mathbf{d}_0 \neq \mathbf{0}$ ). In this case, the master assigns the value **true** to the Boolean variable *exit* in Step 19. If  $\mathbf{d}_k \neq \mathbf{0}$ , then the master calculates the next approximation  $\mathbf{u}_{k+1}$  in Step 21, increases the iteration counter  $k$  by one in Step 22, and assigns the value **false** to the Boolean variable *exit* in Step 23. In Step 25, the master broadcasts the value of the Boolean variable *exit* to all workers. If the Boolean variable *exit* takes the value **true**, then the **repeat/until** loop ends in Step 26. In Step 27, the master outputs the last approximation  $\mathbf{u}_k$  as a result, and the quantity  $f_k$  as the optimal value of the objective function. Step 28 terminates the master process.

All workers execute the same code, but on different data. In Step 1, the  $l$ th worker ( $l = 1, \dots, L$ ) reads the initial data of the LP problem. The **repeat/until** loop of the worker (steps 3–26) corresponds to the **repeat/until** loop of the master. In Step 4, the worker receives the current approximation  $\mathbf{u}_k$  from the master. After that, the worker forms its own sublist  $\mathcal{L}_{map(l)}$  of the subset ordinal numbers to be processed (steps 5–13). The sublists of different workers do not overlap:

$$l' \neq l'' \Leftrightarrow \mathcal{L}_{map(l')} \neq \mathcal{L}_{map(l'')}, \quad (38)$$

and their concatenation gives a complete list:

$$\mathcal{L}_{map} = \mathcal{L}_{map(0)} \# \dots \# \mathcal{L}_{map(L-1)}. \quad (39)$$

In Step 14, the worker calls the higher-order function *Map*, which forms the sublist of pairs  $\mathcal{L}_{reduce(l)}$ , applying the parameterized function  $F_{\mathbf{u}_k}$ , defined by the equations (35), to all elements of the sublist  $\mathcal{L}_{map(l)}$ . In Step 15, the higher-order function *Reduce* transforms this list into the single pair  $(\mathbf{d}_l, f_l)$  by sequentially applying the binary operation  $\oplus$ , defined by the equation (37), to all elements of the sublist  $\mathcal{L}_{reduce(l)}$ . The result is sent to the master in Step 16. In Step 25, the worker receives the value of the Boolean variable *exit* from the master. If this variable takes the value **true**, then the worker process is terminated. Otherwise, the **repeat/until** loop continues to run. The exchange operators **Broadcast**, **Gather**, **RecvFromMaster** and **SendToMaster** perform implicit synchronization of the master process and worker processes.

## 5. Computational Experiments

We implemented the parallel version of the ALFaMove algorithm in C++ using the BSF-skeleton [15], which is based on the BSF parallel computation model [14]. The BSF-skeleton encapsulates all aspects related to parallelizing a program based on the MPI library. The source codes of the parallel implementation of the ALFaMove algorithm are freely available in the GitHub repository at <https://github.com/leonid-sokolinsky/ALFaMove>. The developed program has been tested on a large number of LP problems from various sources. All these problems in MTX format [1] are available at <https://github.com/leonid-sokolinsky/Set-of-LP-Problems>. As tests, we also used synthetic problems obtained using the random problem generator LP FRaGenLP [16]. These problems are available at <https://github.com/leonid-sokolinsky/Set-of-LP-Problems/tree/main/Rnd-LP>. We were unable to test the ALFaMove implementation on problems from the Netlib-LP repository [5], since, in all these problems, the number of hyperplanes passing through the starting point  $\mathbf{u}_0$  exceeded the number 30, which corresponds to the number of possible combinations equal to 1 073 741 824. The C++ compilers available to us do not accept arrays of such sizes.

Using the developed program, we evaluated the scalability of the ALFaMove algorithm. In these experiments, we used the parameterized LP problem called ‘‘cut-off vertex hypercube’’, for which the space dimension  $n$  is a parameter. The constraints of this problem contain the following  $2n + 1$  inequalities:

$$\left\{ \begin{array}{rcl} x_1 & & \leq 200 \\ & x_2 & \leq 200 \\ & & \vdots \\ & & x_n & \leq 200 \\ x_1 + x_2 + \dots + x_n & \leq & 200(n-1) + 100 \\ x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \end{array} \right. \quad (40)$$

The gradient of the objective function is given by the vector

$$\mathbf{c} = (1, 2, \dots, n). \quad (41)$$

It is necessary to find the maximum of the objective function. The problem has the unique solution at the point  $(100, 200, \dots, 200)$  with the maximum value of the objective function equal to  $100(n^2 + n - 1)$ . For an arbitrary  $n$ , this problem can be obtained in MTX format using the FRaGenLP generator, if the number of random inequalities is set to 0. With various  $n$ , these LP problems are available at <https://github.com/leonid-sokolinsky/Set-of-LP-Problems/tree/main/Rnd-LP> under the names `lp_rnd<n>-0`, where the dimension of the space is specified as `<n>`.

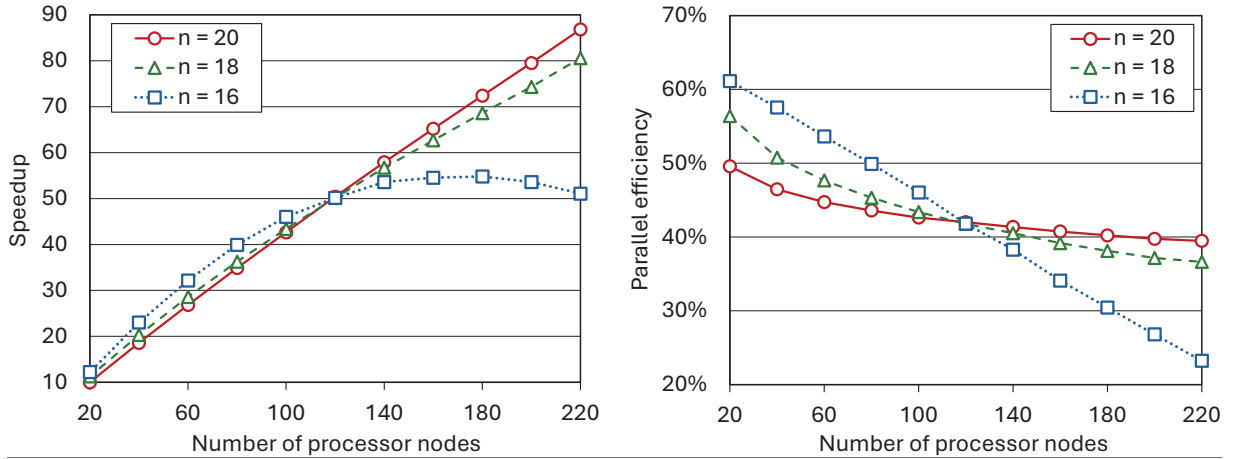
Computational experiments were carried out on the supercomputer ‘‘Lomonosov-2’’ [19], whose specifications are shown in Tab. 1.

All computations were performed with double precision, at which a floating-point number occupies 64 bits in computer memory.

In the first series of experiments, the dependence of the speedup and parallel efficiency of the ALFaMove algorithm on the number of processor nodes for the cut-off vertex hypercube problem was investigated. The results of these experiments are shown in Fig. 6. The speedup  $\alpha(L)$  was defined as the ratio of the time  $T(1)$  of solving a problem in the configuration with

**Table 1.** Specifications of “Lomonosov-2” computing cluster

Parameter	Value
Number of processor nodes	1487
Processor	Intel Haswell-EP E5-2697v3, 2.6 GHz, 14 cores
Memory per node	64 GB
Main network	InfiniBand FDR
Control network	Gigabit Ethernet
Operating system	Linux CentOS 7



**Figure 6.** Speedup and parallel efficiency of the AlFaMove algorithm,  $n$  – number of variables in LP problem (40)

a master node and a single worker node to the time  $T(L)$  of solving the same problem in the configuration with a master node and  $L$  worker nodes:

$$\alpha(L) = \frac{T(1)}{T(L)}.$$

Parallel efficiency  $\beta(L)$  was calculated using the equation

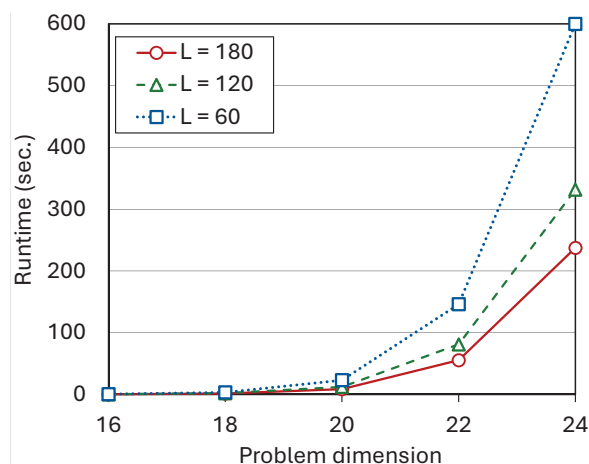
$$\beta(L) = \frac{T(1)}{L \cdot T(L)}.$$

Computations were performed for the dimensions 16, 18 and 20. The number of constraints was 33, 37 and 41 respectively. In all cases, the vertex of the feasible polytope with the following coordinates was chosen as the initial point:

$$x_1 = 0, \quad \dots \quad x_{n/2} = 0, \quad x_{n/2+1} = 200, \quad \dots \quad x_n = 200. \quad (42)$$

The experiments demonstrated good scalability of the AlFaMove algorithm on the cut-off vertex hypercube problem, starting from the dimension  $n = 18$ . In this case, the algorithm demonstrated speedup close to linear. At smaller dimensions, the cost of exchanges and latency begin to dominate the computational costs, which leads to a significant decrease in the algorithm scalability boundary<sup>7</sup>. For  $n = 16$ , this boundary was equal to 180 nodes. Experiments also shown

<sup>7</sup>The scalability boundary refers to the maximum number of processor nodes, up to which the speedup increases.



**Figure 7.** The dependence of AlFaMove runtime on the problem dimension on various multiprocessor configurations ( $L$  – number of processor nodes)

that with an increase in the problem dimension, the parallel efficiency on a small number of processor nodes (less than 120) decreases. However, with a larger number of processor nodes, the opposite trend is observed. So, for the dimension  $n = 16$ , the parallel efficiency was 61% on 20 processor nodes, after which it decreased to 23% on 220 nodes. At the same time, for  $n = 20$ , the parallel efficiency was equal to 50% and 40%, respectively.

In the next series of experiments, the dependence of runtime on the dimension of the cut-off vertex hypercube problem was investigated for various multiprocessor configurations with the number of processor nodes  $L = 60$ ,  $L = 120$  and  $L = 180$ . The results of these experiments are shown in Fig. 7. The dimension ranged from  $n = 16$  to  $n = 24$  in increments of 2. For the dimension  $n = 24$ , each of the lists  $\mathcal{L}_{map}$  and  $\mathcal{L}_{reduce}$  included 16 777 215 elements. This is the maximum size allowed by the compiler used. The vertex of the feasible polytope with coordinates (42) was always chosen as the initial point. In all the studied configurations, the experiments showed an exponential increase in the runtime with an increase in the problem dimension. However, configurations with a large number of processor nodes demonstrated considerably shorter running time of the AlFaMove algorithm.

In the third series of experiments, we investigated the behavior of the AlFaMove algorithm on the Klee–Minty cube. The feasible region of this problem is a hypercube with perturbed corners defined by the following inequalities:

$$\begin{cases} x_1 & \leq 5 \\ 4x_1 + x_2 & \leq 25 \\ 8x_1 + 4x_2 + x_3 & \leq 125 \\ & \vdots \\ 2^n x_1 + 2^{n-1} x_2 + \dots + 4x_{n-1} + x_n & \leq 5^n \\ x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0. \end{cases}$$

The gradient of the objective function is given by the vector

$$\mathbf{c} = (2^{n-1}, 2^{n-2}, \dots, 2, 1).$$

It is necessary to find the maximum of the objective function. The problem has the unique solution at the point  $(0, \dots, 0, 5^n)$  with the maximum value of the objective function equal

to  $5^n$ . In article [7], Victor Klee and George Minty showed that the classical simplex method, starting at  $\mathbf{x} = \mathbf{0}$ , goes through all  $2^n$  hypercube vertices performing  $2^n - 1$  iterations in solving this problem. It is known that many optimization algorithms for linear programming exhibit poor performance when applied to the Klee–Minty cube. We applied the AlFaMove algorithm to Klee–Minty cubes of dimension from 5 to 9. The experimental results presented in Tab. 2 show that the AlFaMove algorithm found a solution in  $2n - 1$  iterations in all cases, while the simplex method performed  $2^n - 1$  iterations.

**Table 2.** Experiments with Klee–Minty cubes

Dimension $n$	AlFaMove				Simplex
	Scalability boundary	Time (sec.)	Relative error $\delta$	Iteration number	Iteration number
5	10	0.2	$0.9 \cdot 10^{-12}$	9	31
6	15	2	$0.2 \cdot 10^{-12}$	11	63
7	20	13	$0.8 \cdot 10^{-11}$	13	127
8	25	126	$0.8 \cdot 10^{-11}$	15	255
9	30	1445	$0.2 \cdot 10^{-10}$	17	511

The relative error was calculated by the equation

$$\delta = \left| \frac{f_{exact} - f_{approx}}{f_{exact}} \right|,$$

where  $f_{exact}$  is the exact maximum value of the objective function,  $f_{approx}$  is the value calculated by the AlFaMove algorithm. The iterations of the simplex method were calculated using the online calculator available at <https://www.pmc calculators.com/simplex-method-calculator>. Experiments also showed that in the case of Klee–Minty cubes, the scalability boundary of the AlFaMove algorithm increased linearly with increasing the problem dimension. At the same time, an exponential increase in runtime was observed.

## Conclusion

The article presents the AlFaMove algorithm, which is a numerical implementation of the surface movement method for linear programming. The key feature of this method is to find out the optimal path along the surface of the feasible polytope from the initial point to a solution of a linear programming problem. The optimal path is understood as a path along the surface of the feasible region in the direction of maximizing the values of the objective function. The scientific significance of the proposed algorithm lies in the fact that it opens up the possibility of using feed forward artificial neural networks to solve non-stationary multidimensional linear programming problems in real time. The theoretical basis of the AlFaMove algorithm is the operation of constructing the pseudoprojection onto linear manifolds that form the feasible polytope flat sides of different dimensions.

Pseudoprojection is implemented on the basis of the Fejér process and is a generalization of the concepts of orthogonal projection on a linear manifold and metric projection on a convex

set. In the case of a hyperplane, the pseudoprojection turns into the orthogonal projection. It is proved that the hyperplane path constructed by the gradient of the objective function and the orthogonal projection is optimal. The projection-type algorithm is presented for constructing a pseudoprojection onto linear manifold formed by the hyperplane intersections. It is proven that the pseudoprojection point coincides with the orthogonal projection point in this case. A formalized description of the AlFaMove algorithm, which builds the optimal path on the surface of the feasible polytope, is presented. The AlFaMove algorithm is based on the procedure for calculating the vector of movement along the face of the feasible polytope from the current approximation in the direction of maximizing the values of the objective function. A formalized description of this procedure is outlined.

Projection-type algorithms are characterized by a low rate of convergence, depending on the angles between the hyperplanes forming the linear manifold. It is also noted that the calculation of the movement vector is a combinatorial-type enumeration problem with high space and time complexity. A parallel version of the AlFaMove algorithm designed for cluster computing systems is presented. The parallel version is implemented in C++ using the BSF-skeleton based on the BSF parallel computation model. Computational experiments were conducted on a cluster computing system to evaluate the scalability of the AlFaMove algorithm. The experiments showed that a linear programming problem with 24 variables and 49 constraints demonstrates a speedup close to linear on 320 processor nodes of the cluster. Problems of a larger dimension led to a compiler error caused by exceeding the maximum acceptable size of arrays. The experiments with the Klee–Minty cube shown that the scalability boundary of the AlFaMove algorithm also increases linearly with increasing the problem dimension.

As directions for further research, we outline the following. We plan to design a new, more efficient method for constructing a path on the surface of a feasible polytope, leading to a solution of a linear programming problem. The main idea is to decrease the number of enumerating combinations of hyperplanes when determining the direction of movement. This can be achieved by restricting the paths of movement only to the edges of the polytope (segments of linear manifolds of dimension one). The problem of space complexity can be solved by using stochastic methods of choosing the movement direction.

## Acknowledgements

The research was supported by the Russian Science Foundation (project No. 23-21-00356) and carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

*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. Boisvert, R.F., Pozo, R., Remington, K.A.: The Matrix Market Exchange Formats: Initial Design. Tech. rep., NISTIR 5935. National Institute of Standards and Technology, Gaithersburg, MD (1996), <https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir5935.pdf>, accessed: 2024-08-26



2. Branke, J.: Optimization in Dynamic Environments. In: Evolutionary Optimization in Dynamic Environments. Genetic Algorithms and Evolutionary Computation, vol. 3, pp. 13–29. Springer, Boston, MA (2002). [https://doi.org/10.1007/978-1-4615-0911-0\\_2](https://doi.org/10.1007/978-1-4615-0911-0_2)
3. Dantzig, G.B.: Linear programming and extensions. Princeton university press, Princeton, N.J. (1998)
4. Fathi, M., Khakifirooz, M., Pardalos, P.M. (eds.): Optimization in Large Scale Problems: Industry 4.0 and Society 5.0 Applications. Springer, Cham, Switzerland (2019). <https://doi.org/10.1007/978-3-030-28565-4>
5. Gay, D.M.: Electronic mail distribution of linear programming test problems. Mathematical Programming Society COAL Bulletin 13, 10–12 (1985)
6. Gould, N.I.: How good are projection methods for convex feasibility problems? Computational Optimization and Applications 40(1), 1–12 (2008). <https://doi.org/10.1007/S10589-007-9073-5>
7. Klee, V., Minty, G.J.: How good is the simplex algorithm? In: Shisha, O. (ed.) Inequalities - III. Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, Sept. 1-9, 1969. pp. 159–175. Academic Press, New York, NY, USA (1972)
8. Kopanos, G.M., Puigjaner, L.: Solving Large-Scale Production Scheduling and Planning in the Process Industries. Springer, Cham, Switzerland (2019). <https://doi.org/10.1007/978-3-030-01183-3>
9. Maltsev, A.: The basics of linear algebra. Science. The main editorial office of the phys-math literature, Moskow (1970), (in Russian)
10. Mamalis, B., Pantziou, G.: Advances in the Parallelization of the Simplex Method. In: Zaroliagis, C., Pantziou, G., Kontogiannis, S. (eds.) Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science, vol. 9295, pp. 281–307. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-24024-4\\_17](https://doi.org/10.1007/978-3-319-24024-4_17)
11. Olkhovsky, N.A., Sokolinsky, L.B.: Surface Movement Method for Linear Programming. Lobachevskii Journal of Mathematics 45(10), (in print) (2024)
12. Schlenkrich, M., Parragh, S.N.: Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art. In: Longo, F., Affenzeller, M., Padovano, A., Shen, W. (eds.) 4th International Conference on Industry 4.0 and Smart Manufacturing. Procedia Computer Science. vol. 217, pp. 1028–1037. Elsevier (2023). <https://doi.org/10.1016/J.PROCS.2022.12.301>
13. Sokolinskaya, I.M., Sokolinsky, L.B.: On the Solution of Linear Programming Problems in the Age of Big Data. In: Sokolinsky, L., Zymbler, M. (eds.) Parallel Computational Technologies. PCT 2017. Communications in Computer and Information Science, vol. 753. pp. 86–100. Springer, Cham, Switzerland (2017). [https://doi.org/10.1007/978-3-319-67035-5\\_7](https://doi.org/10.1007/978-3-319-67035-5_7)
14. Sokolinsky, L.B.: BSF: A parallel computation model for scalability estimation of iterative numerical algorithms on cluster computing systems. Journal of Parallel and Distributed Computing 149, 193–206 (2021). <https://doi.org/10.1016/j.jpdc.2020.12.009>

15. Sokolinsky, L.B.: BSF-skeleton: A Template for Parallelization of Iterative Numerical Algorithms on Cluster Computing Systems. *MethodsX* 8, Article number 101437 (2021). <https://doi.org/10.1016/j.mex.2021.101437>
16. Sokolinsky, L.B., Sokolinskaya, I.M.: FRaGenLP: A Generator of Random Linear Programming Problems for Cluster Computing Systems. In: Sokolinsky, L., Zymbler, M. (eds.) *Parallel Computational Technologies. PCT 2021. Communications in Computer and Information Science*, vol. 1437. pp. 164–177. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-81691-9\\_12](https://doi.org/10.1007/978-3-030-81691-9_12)
17. Sokolinsky, L.B., Sokolinskaya, I.M.: Apex Method: A New Scalable Iterative Method for Linear Programming. *Mathematics* 11(7), 1–28 (2023). <https://doi.org/10.3390/MATH11071654>
18. Vasin, V.V., Eremin, I.I.: *Operators and Iterative Processes of Fejér Type. Theory and Applications. Inverse and Ill-Posed Problems Series*, Walter de Gruyter, Berlin, New York (2009). <https://doi.org/10.1515/9783110218190>
19. Voevodin, V., Antonov, A., Nkitenko, D., Shvets, P., Sobolev, S., Sidorov, I., Stefanov, K., Voevodin, V., Zhumatiy, S.: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. *Supercomputing Frontiers and Innovations* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
20. Zorkaltsev, V., Mokryi, I.: Interior point algorithms in linear optimization. *Journal of applied and industrial mathematics* 12(1), 191–199 (2018). <https://doi.org/10.1134/S1990478918010179>

# Study of the Effectiveness of Parallel Algorithms for Modeling the Dynamics of Collisionless Galactic Systems on GPUs

Sergei S. Khrapov<sup>1</sup> , Alexander V. Khoperskov<sup>1</sup> 

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

*N*-body model is a common research tool in galaxy physics and cosmology. The transition to the use of computing systems with GPUs can significantly improve the performance and quality of simulation results for gravitational systems. *N*-body – Particle-Particle algorithm is presented on a hybrid computing platform CPU + multi-GPUs. Using a direct method of calculating gravitational forces by summing the interactions of each particle with each other is resource-intensive, but provides the best accuracy in modeling dynamics at all scales. The main result is an analysis of the efficiency of parallel code depending on the number of GPUs and the choice of single and double precision floating-point arithmetics. The laws of conservation of energy, momentum and angular momentum are tested for a series of models, including major mergers of galaxies and the evolution of galactic stellar disc subject to the most severe gravitational instability. The general conclusion is that conservation laws are poorly implemented when using 4-byte numbers due to the accumulation of arithmetic errors. Calculations with 8-byte numbers ensure that the laws of conservation of momentum and angular momentum are satisfied to the limit of arithmetic accuracy without accumulating errors. The law of conservation of energy is determined primarily by the order of the numerical scheme for integrating the equations of motion. The additional reduction in the error of the conservation law of total energy due to the transition from 4-byte to 8-byte numbers is 1–2 orders of magnitude. Increasing the number of GPUs used helps improve the implementation of conservation laws due to a decrease in the number of particles per graphics processing unit.

*Keywords:* *N*-body, GPUs, OpenMP-CUDA, GPUDirect, efficiency.

## Introduction

Models of the dynamics of interacting particles are used to describe a wide variety of physical systems and processes from chemistry and plasma physics [2, 19, 20] to astrophysical objects [6, 10, 13, 14, 16, 23]. The properties of the physical medium are determined by the type of field interaction between particles. Improving the quality of modeling, the dynamics of a system requires an increase in the number of particles  $N$ , which is limited by computing resources. The *N*-body model belongs to a class of molecular dynamics models based on the motion simulations of a large number of interacting particles [9, 29]. This approach is effective for studying the dynamics of rarefied gases [7], large atomistic clusters [11], biomolecules and biological systems [2, 9]. Molecular and atomistic models use short-range potentials such as van de Waals' force or Debye screening of charges in a quasi-neutral medium [11]. The gravitational potential is always long-range and the most distant parts in a gravitationally bound system can make a significant contribution to the force [1, 5, 14, 21].

Molecular Dynamics Simulation (or *N*-body) problems are often critically dependent on the number of particles, so the new computing advantages of GPUs significantly improve modeling efficiency due to price-performance ratio [2, 6, 14]. An important excellence of computing systems with GPUs is the ability to perform massive series of simulations to build large datasets, using the appropriate subsystems of supercomputers [26].

The number of objects  $N_*$  (stars, gas clouds) in real gravitating astrophysical systems, as a rule, significantly exceeds the number of model particles  $N$ , which leads to the problem of

---

<sup>1</sup>Volgograd State University, Volgograd, Russian Federation

ensuring collisionlessness in the numerical model. For example, a typical S-galaxy with a number of stars of the order of  $N_\star \sim 10^{11}$  is a collisionless system in which the role of pair interactions is negligible compared to the influence of the mean field. Modeling with a particle number of  $N_\star/N \gg 1$  implies the use of macroparticles with a mass  $N_\star/N$  times greater than the mass of an average star. The collisionlessness of the gravitating system is ensured by using softening radius  $r_c$  at small distances to avoid pair interactions. The choice of the optimal smoothing parameter  $r_c$  depends on the number of particles, system configuration and other factors [18, 25]. The characteristic relaxation time in the stellar components of galaxies is  $\propto N/\ln(N)$ , which preserves collisionless at cosmological times [1, 5].

The traditional approach for calculating the gravitational force from  $N$  particles in an astrophysical system is based on various approximate methods, including the fast Fourier transform, various versions of TreeCode, wavelet transforms, etc. [15, 24, 27]. The use of approximate methods for calculating the gravitational potential is dictated by the desire to have as many particles  $N$  as possible, which, however, is accompanied by an increase in the error of the interaction force.

The duration of the studied evolution of galactic systems can be long and often reaches  $t^{(\max)} \sim 10$  billion years [16]. Moreover, the integration step  $\Delta t$  is limited by the inhomogeneity of the components on small scales and is within approximately  $\Delta t \sim 10^5$  years or less. Simultaneous modeling of the gas component can significantly reduce the integration step due to larger gradients of gas density distribution. Let us estimate the errors in calculating the gravitational force for an extended system of size  $r^{(\max)}$  and the minimum distance between a closely located pair of particles  $r^{(\min)}$ . Then the forces for nearby particles  $f(r^{(\min)})$  and distant particles are related as the squares of the radii and can reach  $(r^{(\min)}/r_c)^2 \simeq 10^7$  inside a typical galaxy. In the case of modeling interacting galaxies, the force ratio between the nearest particles and the most distant particles turns out to be even greater and exceeds  $10^8$ . As a result, the contribution from distant particles adds up with an error or is even lost, depending on the length of the numbers used.

The rapid increase in the performance of modern GPUs provides new opportunities for using direct methods for calculating gravitational forces, when each particle interacts with each other (Particle-Particle algorithm, PP) [1, 16, 23] and allows to perform numerous computational experiments to study galaxies with  $N > 10^6$ .

The purpose of this work is a detailed analysis of the quality of galaxy simulations within the framework of the  $N$ -body method using GPUs for a direct method of calculating gravity by summing the contributions of all particles. Conducting computational experiments under various conditions is aimed at studying the effectiveness of parallel implementations of the  $N$ -body numerical algorithm on hybrid computing platforms with multi-GPUs using single and double precision floating-point arithmetics. We focus on the accuracy of the conservation laws of momentum, angular momentum and energy of the total gravitational system, depending on the number of GPUs and the use of single or double precision arithmetic.

The article is organized as follows. Section 1 contains descriptions of the numerical algorithm for the  $N$ -body problem and the main characteristics of the models of colliding galactic systems. In Section 2, we discuss the hardware and algorithmic features of the parallel implementation of calculating gravitational forces in a system of  $N$  particles. Section 3 is devoted to the analysis of the accuracy of conservation laws in a dynamic system for various ways of organizing par-

allel computations. Finally, the conclusion summarizes the study and outlines potential future research topics.

## 1. Algorithms for Integrating Equations of Motion in the $N$ -body Model

The  $N$ -body model looks quite simple and attractive, based on a system of ordinary differential equations of motion for a large number of gravitationally interacting points

$$\frac{d^2\mathbf{r}_i}{dt^2} = \sum_{j=1}^N \mathbf{f}_{ij}, \quad \mathbf{u}_i = \frac{d\mathbf{r}_i}{dt}, \quad i = 1, 2, \dots, N, \quad (1)$$

where  $\mathbf{f}_{ij}$  is the force between the  $i$ -th particle with mass  $m_i$  and the  $j$ -th particle ( $i \neq j$ ) with mass  $m_j$ . Direct calculation of the force  $\mathbf{f}_{ij}$  involves the use of Newton's law in the form

$$\mathbf{f}_{ij} = \frac{Gm_j}{(r_{ij}^2 + r_c^2)^{3/2}} (\mathbf{r}_i - \mathbf{r}_j), \quad (2)$$

where  $G$  is the gravitational constant,  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$  is the distance between two particles,  $r_c$  is the softening radius. The value  $r_c > 0$  is required to ensure that the system is collisionless in the case of  $N \ll N_r$ .

Directly calculating all forces in (1) using (2) gives quadratic complexity  $O(N^2)$ . The approximate hierarchical TreeCode method has  $O(N \log(N))$ , increasing the error in gravitational force calculation [12, 22].

The three-stage Newton–Störmer–Verlet-leapfrog (or Kick-Drift-Kick, KDK) scheme is traditionally used for numerical integration of system (1) with some modifications [8]. Successive calculations of intermediate velocities at the first stage

$$\tilde{\mathbf{u}}_i(t + \Delta t) = \mathbf{u}_i(t) + \Delta t \sum_{j=1, j \neq i}^N \mathbf{f}_{ij}(t) \quad (3)$$

give the positions of all particles at time  $t + \Delta t$  at the second step

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \frac{\Delta t}{2} [\tilde{\mathbf{u}}_i(t + \Delta t) + \mathbf{u}_i(t)]. \quad (4)$$

Then, we calculate the velocities at time  $t + \Delta t$  in the third stage

$$\mathbf{v}_i(t + \Delta t) = \frac{\mathbf{u}_i(t) + \tilde{\mathbf{u}}_i(t + \Delta t)}{2} + \frac{\Delta t}{2} \sum_{j=1, j \neq i}^N \mathbf{f}_{ij}(t + \Delta t). \quad (5)$$

The forces at time  $t + \Delta t$  are used at the next iteration, so the main advantage of KDK is only a one-time calculation of the forces on the right side of the equations (1), which, however, provides second order accuracy  $O(\Delta t^2)$  at one time step.

We study the evolution of the initial states of gravitating systems related to severe tests in which complex flows are formed with the development of strong gravitational instability. Table 1 contains the number of particles in the disc  $N^{(d)}$ , the number of particles in the dark hot spheroidal halo  $N^{(h)}$  and the Toomre parameter  $Q_T$  in the central region of the disc, characterizing the effective temperature of the matter (particle velocity dispersion). The quantity

**Table 1.** Main parameters of the models

Model	$N^{(d)}$	$N^{(h)}$	$Q_T$	Comment
Name	Discs	Halo	$r < 0.5$	
D100	$2^{18}$	$2^{19}$	$Q_T \simeq 1.25$	Disc + halo
D101	$2^{19}$	$2^{19}$	$Q_T \simeq 0.8$	Disc + halo
D102	$2^{19}$	—	$Q_T \simeq 1$	Disc without halo
D103	$2^{19}$	—	$Q_T \simeq 0.1$	Disc without halo
D201	$2^{19}+2^{19}$	$2^{19}+2^{19}$	$Q_T \simeq 0.8$	Collision of two galaxies

$Q_T = c_r/(3.36G\sigma/\kappa)$  is traditionally used to determine the limit of gravitational stability of the stellar disc ( $c_r$  is the radial velocity dispersion,  $\sigma$  is the surface density,  $\kappa$  is the epicyclic frequency) [5]. We consider the crash tests in models D103 and D201, during which the fastest processes occur inside small sizes at large density gradients and discs destruction occurs.

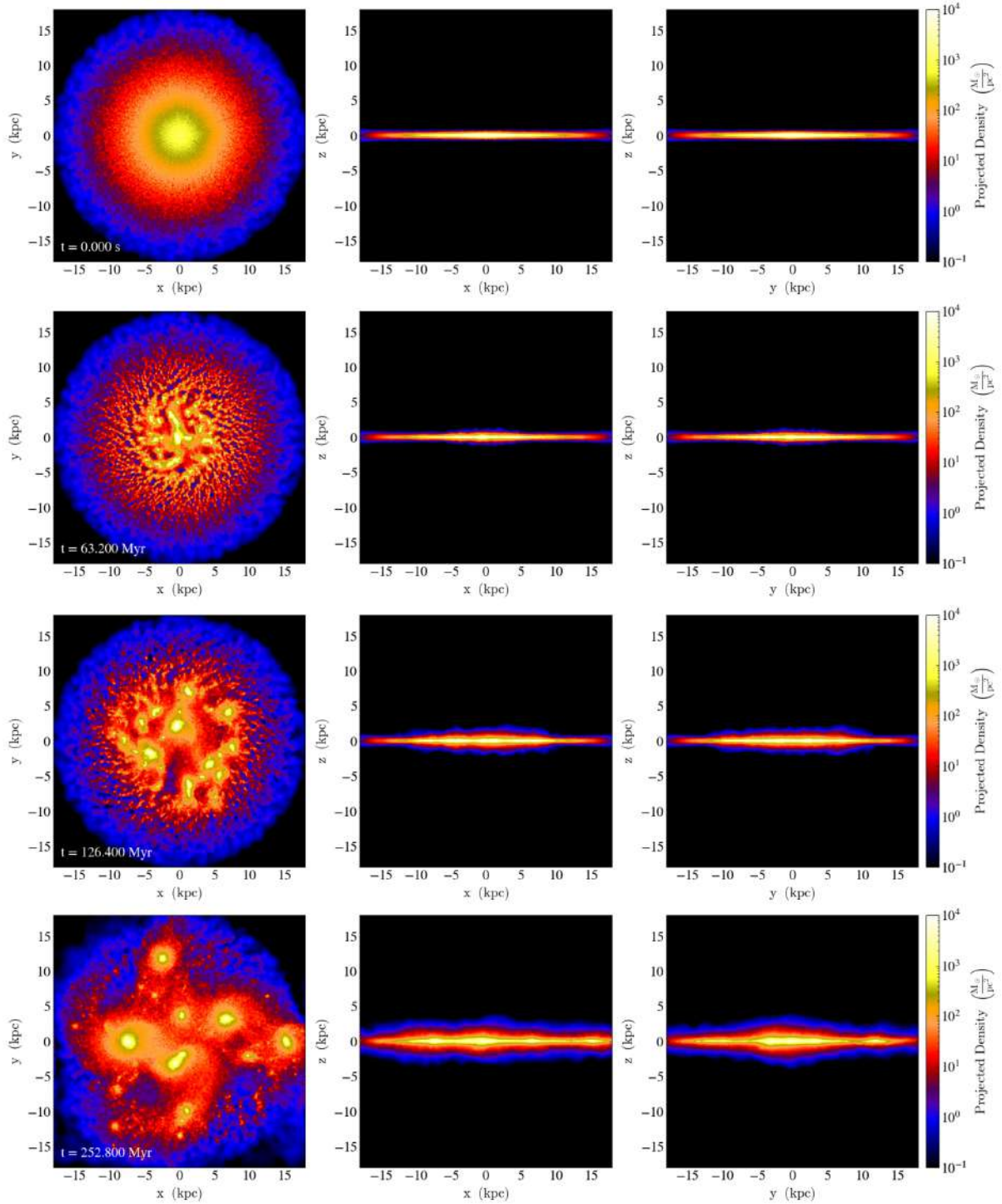
The model D201 reproduces the collision of two disc systems almost flat, leading to a large merger through the passage of two galaxies through each other. Figures 1 and 2 show the dynamics of two models in three perpendicular planes. Model D103 describes an initial very cold disc with small Toomre parameter ( $Q_T \simeq 0.1$ ), which leads to the rapid development of strong gravitational instability. This model does not contain a dark halo, which has a stabilizing effect on gravitational instability. As a result, the disc matter is divided into several massive clumps, which are slowly destroyed during the heating of the system with the formation of a hot extended disc with a significantly reconstructed radial density profile. Model D103 is not of physical interest due to the condition  $Q_T \simeq 0.1$ , but it is a good touchstone for checking calculations. The proximity of the parameter  $Q_T$  to zero leads to the development of the most powerful gravitational instability. As a result, a dynamically very cold disc breaks up into several isolated, long-lived, small, high-density clumps that actively interact with each other (see two bottom panels in Fig. 1). The rotating disc lies in the plane  $(x, y)$  in all models at the initial time.

Model D201 includes two identical galaxies embedded in a dark halo. We push them together at an angle of  $45^\circ$  (Fig. 2). This simulation of centrally colliding two-component disks with a dark, massive halo is an example of strong interaction. It ends with a large merging of the two systems. Note that the observed galaxies type Taffy for the pairs UGC12914/UGC12915, NGC7733/NGC7734 apparently goes through such a stage of evolution [3].

We use a system of dimensionless quantities to conveniently represent galactic characteristics so that all dimensionless parameters are of the order of unity under typical conditions. Conversion from standard astronomical characteristics of length ( $1 \text{ pc} \simeq 3.086 \cdot 10^{16} \text{ m}$ ), mass ( $1 M_\odot = 1.989 \cdot 10^{30} \text{ kg}$ , solar mass) to dimensionless quantities is carried out by the factors  $\ell_r = 9000 \text{ pc}$  and  $\ell_M = 3.72 \cdot 10^{10} M_\odot$ , respectively [16]. The units of time  $\ell_t$  and velocity  $\ell_V$  are equal to  $\ell_t = 63.2 \text{ Myr}$ ,  $\ell_V = 133.7 \text{ km s}^{-1}$ .

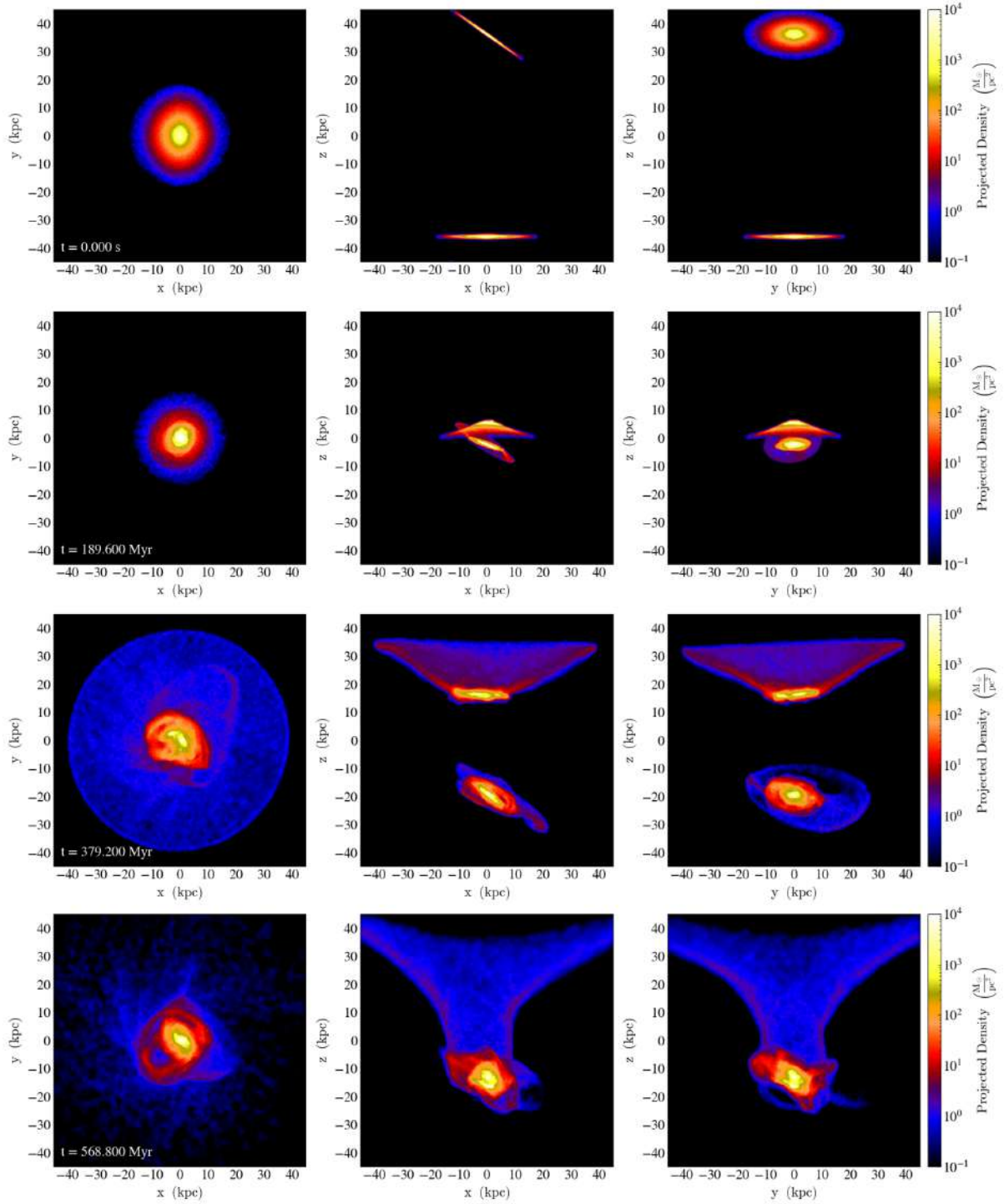
## 2. Features of Parallel Code Implementations

We used hybrid computing platforms with multiple GPUs (CPU+2GPU and CPU+4GPU). The simulations on the computing architecture CPU+2GPU were carried out on Lomonosov-2 – Volta-1 (Lomonosov Moscow State University) supercomputer: CPU (Intel Xeon Gold 6142) + 2GPU (Nvidia Tesla V100). The computing architecture CPU + 4GPU was used on VolSU –



**Figure 1.** Evolution of an isolated very cold disc without a dark halo (model D103)

Nvidia DGX-1 supercomputer: 2CPU (Intel Xeon E5-2698) + 8GPU (Nvidia Tesla V100). We parallelized our algorithm  $N$ -body – PP based on technologies OpenMP + CUDA + GPUDirect, which allow parallel calculations to be performed on one computing node with several GPUs. OpenMP technology is used to parallel run CUDA-kernel on multiple GPUs. GPUDirect technology creates a common memory address space for multi-GPUs and allows CUDA-threads to communicate directly through the NVLINK or PCIe interface, bypassing CPU memory. Figure 3 shows the principle of organizing calculations using our algorithm  $N$ -body – PP on a hybrid com-

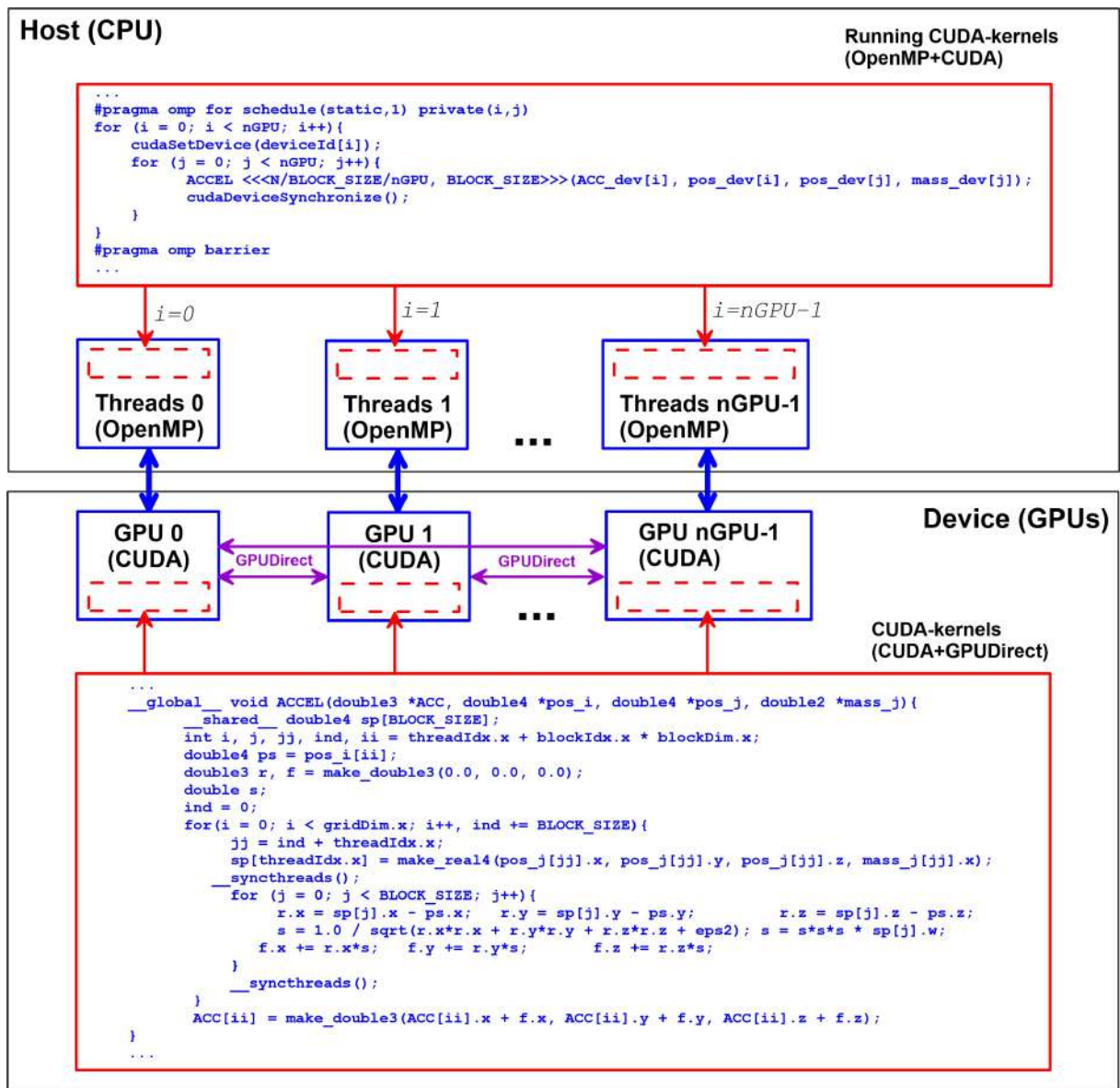


**Figure 2.** Evolution of two colliding Taffy-type disc galaxies (model D201)

puting platform CPU + multi-GPUs. Scheme in Fig. 3 also shows the most resource-intensive part of the code associated with calculating the gravitational interaction (PP).

The problem of the quality of numerical  $N$ -body models using high performance GPUs single-precision performance is relevant [6]. Figure 4 shows the results of analyzing the efficiency of code parallelization under various conditions. We carry out calculations with different numbers of GPUs:  $n_G = 1, 2, 4, n_G \text{ GPU}$ . All simulations are duplicated using 4-byte (FP32) and 8-byte (FP64) numbers.

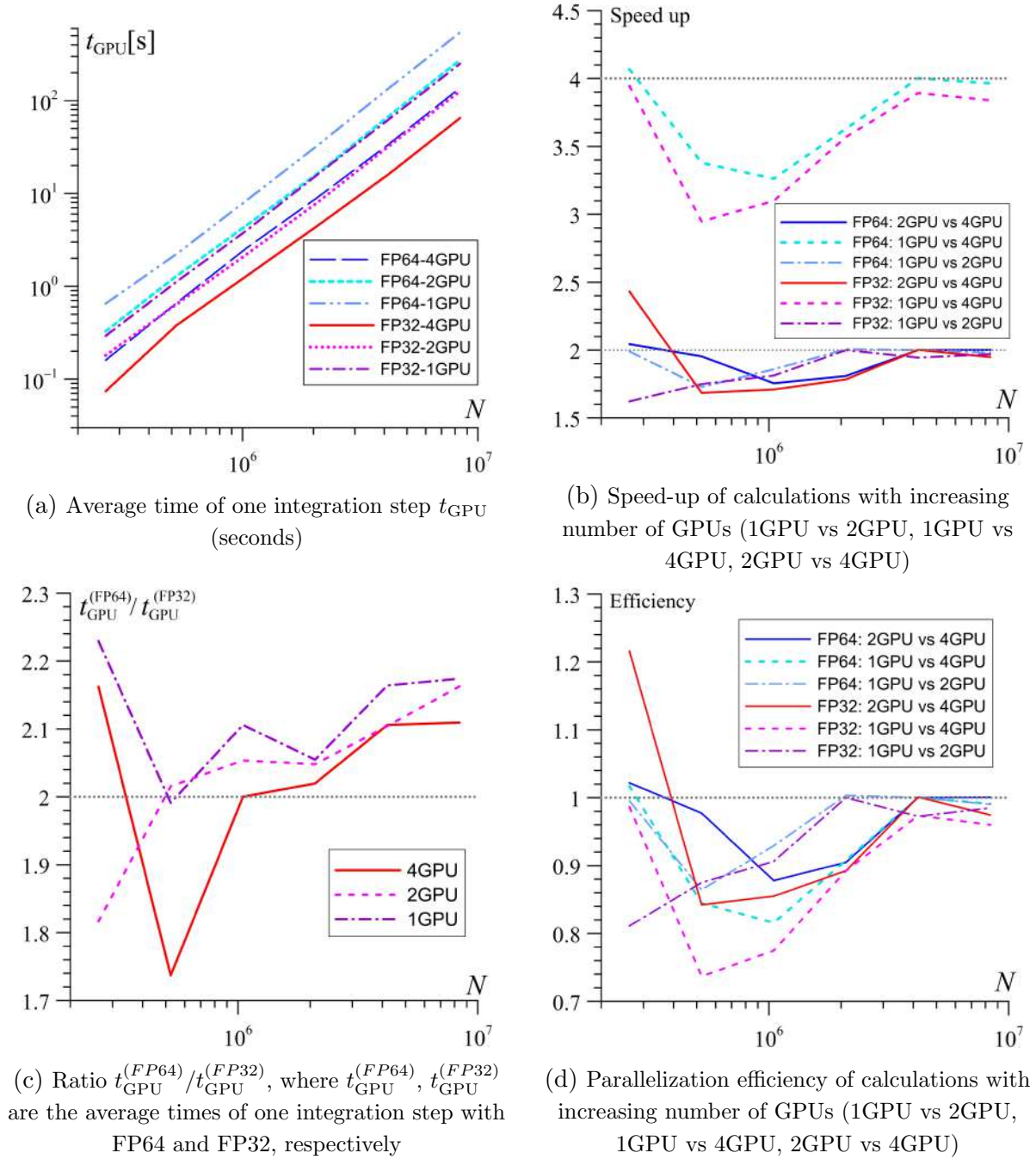




**Figure 3.** Scheme of implementation of the algorithm  $N$ -body – PP with code fragments on hybrid computing platform CPU + multi-GPUs

When analyzing the performance of parallel computing on various platforms CPU+ $n_G$ GPU, we consider only the execution time of the parallel part of our  $N$ -body–PP code (see Fig. 3), in which gravitational forces between particles are calculated by the direct method (2) and integration of the equations of motion (1) is carried out using the method (3)–(5). The time it takes to copy data from the GPU to the CPU and write it to disc is not taken into account. Figure 4a shows the dependence of the computation time for the parallel  $N$ -body – PP algorithm on the number of particles in various computational models. There is a quadratic law in the form  $t_{GPU} \propto N^2/n_c$ , where  $n_c$  is the total number of computing cores of GPU.

The Nvidia Tesla V100 GPU contains  $n_c = 2560$  cores for double precision (FP64) and  $n_c = 5120$  cores for single precision (FP32), so the computational performance for numbers FP32 is approximately 2 times faster than with FP64 (see Fig. 4a,c). Figures 4b,d demonstrate the characteristic features of parallelization of the  $N$ -body – PP algorithm on multi-GPUs. The



**Figure 4.** Dependence of the performance of multi-GPU calculations on the number of particles  $N$  in various computational models for the  $N$ -body – PP algorithm

speed-up and efficiency curves have a minimum near  $N = 2^{19}$ – $2^{20}$  when comparing different computational models. The parallelization efficiency of our algorithm on multi-GPUs tends to 1 as the number of particles increases after  $N \geq 2^{22}$ . Note the anomalous behavior of speed-up and efficiency at  $N = 2^{18}$ , which may be associated with an increase in the data transfer rate between GPUs via NVLINK interface when the data volume is less than a certain threshold value.

The performance of two hybrid computing platforms CPU+2GPU and 2CPU+8GPU was compared for our algorithm  $N$ -body–PP. Supercomputer Lomonosov-2 – Volta-1 for computing

on 1GPU and 2GPU with numbers FP64 is approximately 4–5 percent more productive than VolSU – DGX-1.

Data copy time between CPU (Device) and GPU (Host) depends on the memory bus bandwidth and the amount of data being copied. Therefore, the copying time is proportional to the number of particles  $N$ . The calculation time of gravitational forces in our  $N$ -Body-Particle-Particle algorithm is  $\propto N^2$ , so replacing the GPU-Direct technology with direct copying of data between the GPU and CPU in our code does not lead to a significant decrease in speed-up and parallelization efficiency on multi-GPU at  $N > 10^5$ . These times can be comparable only for very small  $N$ . Our estimates of the speed-up degradation for different values of  $N$  show that the speed-up of computations without using GPU-Direct is less than 1% for  $N > 2^{18}$  (Tab. 2). The use of GPU-Direct technology in numerical algorithms with lower computational complexity, for example,  $\propto N \ln N$  for treecode or  $\propto N$  in the case of hydrodynamic simulations, should lead to more significant gains in speed-up and parallelization efficiency on multi-GPUs. Direct data copying between GPU and CPU has another drawback, which is the duplication of arrays as the number of GPUs increases, since all particle positions must be stored on each GPU at each computational time. This results in an increase in memory space by a factor of  $k$  (where  $k \simeq 0.58 + 0.42 \cdot n_{\text{GPU}}$ ) on each GPU compared to GPU-Direct.

**Table 2.** Average time of one integration step on multi-GPU without using GPU-Direct ( $t_{n_{\text{GPU}}}^*$ ) and using GPU-Direct ( $t_{n_{\text{GPU}}}$ ) for different  $N$

	$N = 2^{18}$	$N = 2^{19}$	$N = 2^{20}$	$N = 2^{21}$	$N = 2^{22}$	$N = 2^{23}$
$t_{2\text{GPU}}^*$ , [s]	$0.6484 \times 2^{-1}$	$0.6475 \times 2^1$	$0.5661 \times 2^3$	$0.5276 \times 2^5$	$0.5253 \times 2^7$	$0.5254 \times 2^9$
$t_{2\text{GPU}}$ , [s]	$0.6444 \times 2^{-1}$	$0.6450 \times 2^1$	$0.5648 \times 2^3$	$0.5268 \times 2^5$	$0.5249 \times 2^7$	$0.5253 \times 2^9$
$t_{4\text{GPU}}^*$ , [s]	$0.3238 \times 2^{-1}$	$0.3303 \times 2^1$	$0.3289 \times 2^3$	$0.2877 \times 2^5$	$0.2679 \times 2^7$	$0.2681 \times 2^9$
$t_{4\text{GPU}}$ , [s]	$0.3210 \times 2^{-1}$	$0.3285 \times 2^1$	$0.3280 \times 2^3$	$0.2872 \times 2^5$	$0.2678 \times 2^7$	$0.2680 \times 2^9$

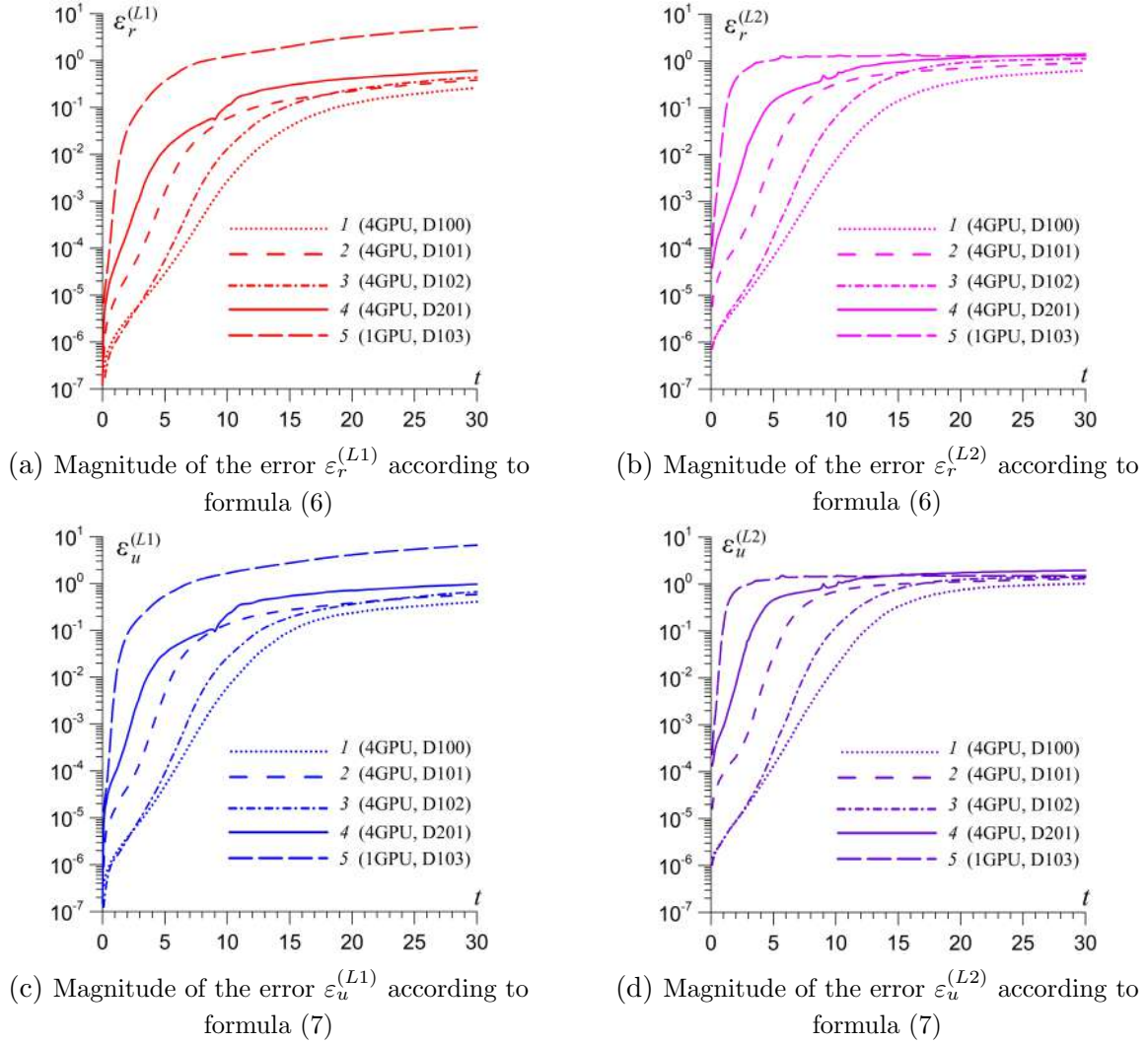
The complete set of trajectories in the phase space  $\{\mathbf{r}_i, \mathbf{u}_i\}$  ( $i = 1, 2, \dots, N$ ) depends on the length of the numbers (FP32 or FP64), all other things being equal. The average divergence of such trajectories is determined by calculating the parameters

$$\varepsilon_r^{(L1)} = \frac{1}{N} \sum_{i=1}^N \left| \mathbf{r}_i^{(FP32)} - \mathbf{r}_i^{(FP64)} \right|, \quad \varepsilon_r^{(L2)} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left| \mathbf{r}_i^{(FP32)} - \mathbf{r}_i^{(FP64)} \right|^2}, \quad (6)$$

$$\varepsilon_u^{(L1)} = \frac{1}{N} \sum_{i=1}^N \left| \mathbf{u}_i^{(FP32)} - \mathbf{u}_i^{(FP64)} \right|, \quad \varepsilon_u^{(L2)} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left| \mathbf{u}_i^{(FP32)} - \mathbf{u}_i^{(FP64)} \right|^2} \quad (7)$$

for the metrics  $L1$  and  $L2$  at each time instant  $t$  during the simulations.

Figure 5 shows the average integral divergence of trajectories in phase space in accordance with (6) and (7) in different models (see Tab. 1) on 1GPU and 4GPU. The accumulation of errors occurs primarily at the initial stage of evolution, when powerful spiral structures are formed in an isolated disk due to gravitational instability (models D100–D103). A similar dependence is obtained in the process of a large merger of two galaxies into one in the D201 model. The evolution of all models over long times ends in quasi-stationary states, when macroscopic characteristics (density, velocities, dispersions of velocity components) practically cease to change. Such quasi-stationary systems are characterized by a very slow increase in the parameters  $\varepsilon_{r,u}^{(L1)}$ ,  $\varepsilon_{r,u}^{(L2)}$  or no changes.



**Figure 5.** Dependences of errors  $\varepsilon_{r,u}^{(L1)}$ ,  $\varepsilon_{r,u}^{(L2)}$  on time for different experiments with different numbers of GPUs

A stronger initial nonstationarity of the gravitating system leads to a more rapid growth of  $\varepsilon_{r,u}^{(L1)}$ ,  $\varepsilon_{r,u}^{(L2)}$ , which stops at more high level. The smallest discrepancies between the phase trajectories are obtained in model D100, in which the initial disc is only marginally unstable and the slow formation of spiral arms of small amplitude is observed.

### 3. Problems of Fulfilling Conservation Laws

The law of conservation of energy  $E$  for a system of  $N$  interacting particles in the absence of dissipation and external forces is determined by the following expression:

$$E = \sum_{i=1}^N \frac{m_i |\mathbf{v}_i|^2}{2} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{G m_i m_j}{(r_{ij}^2 + r_c^2)^{1/2}}, \quad (8)$$

where  $m_i$  is the mass of the  $i$ -th particle,  $j \neq i$ ,  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$  is the distance between two points.

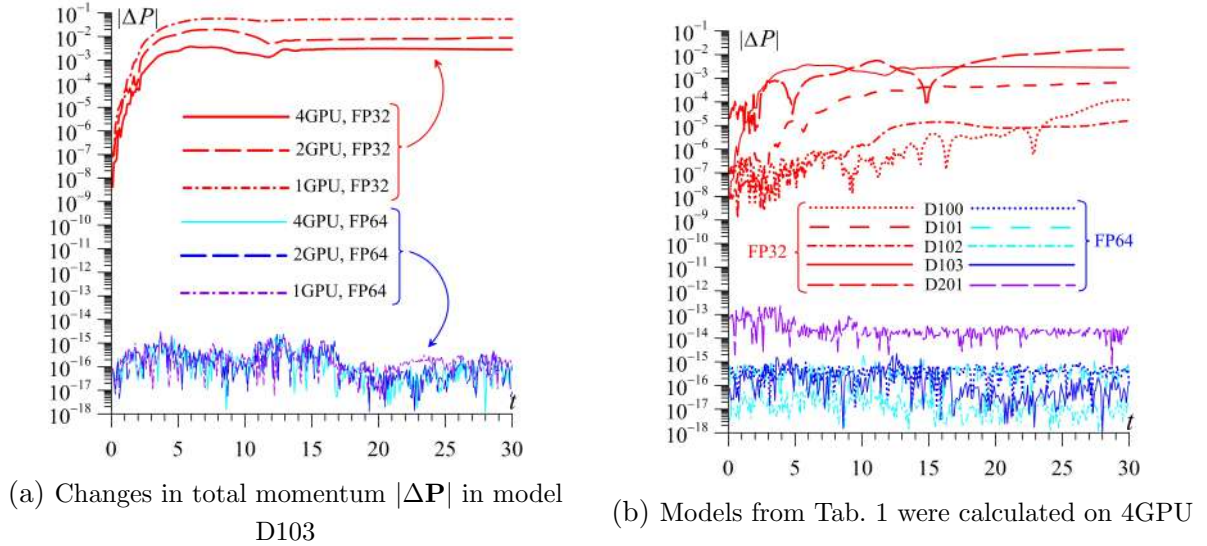
Conservation of total momentum

$$\mathbf{P} = \sum_{i=1}^N m_i \mathbf{u}_i, \quad (9)$$

and angular momentum

$$\mathbf{L} = \sum_{i=1}^N m_i [\mathbf{r}_i \times \mathbf{u}_i]_z \quad (10)$$

in explicit form does not depend on the softening radius  $r_c$ .

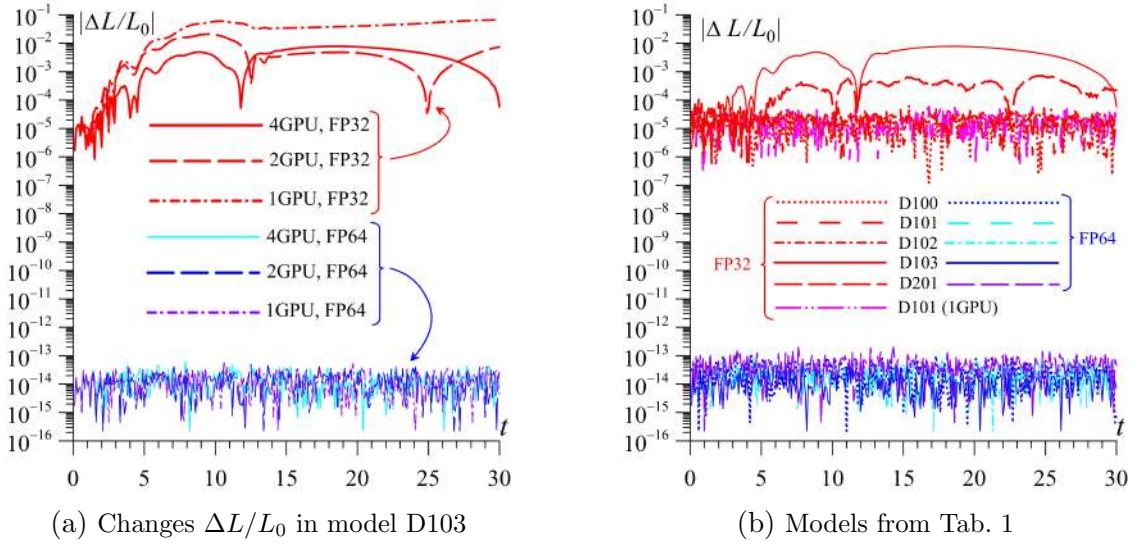


**Figure 6.** Changes in total momentum in the system  $|\Delta \mathbf{P}|$  under different conditions

Modeling of a gravitating system in accordance with (1) should ensure the fulfillment of the laws of conservation of total momentum  $\mathbf{P}$  (9), angular momentum  $\mathbf{L}$  (10) and energy (8). We consider in detail the problem of conservation of (8)–(9) when using numbers of different lengths in parallel calculations with different numbers of GPUs. Figure 6 shows the accuracy of total momentum conservation depending on the calculation conditions. Analysis of motion trajectories gives the worst results for model D103 in Fig. 5, therefore, the dependencies  $|\Delta \mathbf{P}(t)|$  for this model are constructed separately (Fig. 6b), where two features stand out. Firstly, calculations with FP32 give a rapid increase in error and  $|\Delta \mathbf{P}|$  increases by 5–6 orders of magnitude. Using FP64 keeps  $|\Delta \mathbf{P}|$  approximately at the entry level within  $10^{-17}$ – $10^{-15}$ . This finding holds true for any model and number of GPUs. The second feature is more subtle and is related to the number of GPUs used. Momentum is better preserved as the number of GPUs increases, and this effect can be significant (compare the red lines in Fig. 6a).

Analysis of the simulation results of various models from Tab. 1 confirms the inadmissibility of using FP32, which leads to large errors for  $\mathbf{P}$  (see Fig. 6b for 4GPU). The error only increases when using 2GPU or 1GPU. Higher starting level  $|\Delta \mathbf{P}| \sim 10^{-5}$  in model D201 is associated with the peculiarity of constructing the initial state for a pair of colliding galaxies. However, it is important to emphasize that calculations with FP64 keep  $|\Delta \mathbf{P}|$  within the initial limits ( $\sim 10^{-13}$ ).

Disc galactic subsystems rotate rapidly and the azimuthal velocity significantly exceeds the characteristic thermal velocities of particles in the disc. This rotation velocity is comparable to the thermal velocities of the dark matter in the halo. The total initial angular momentum of the



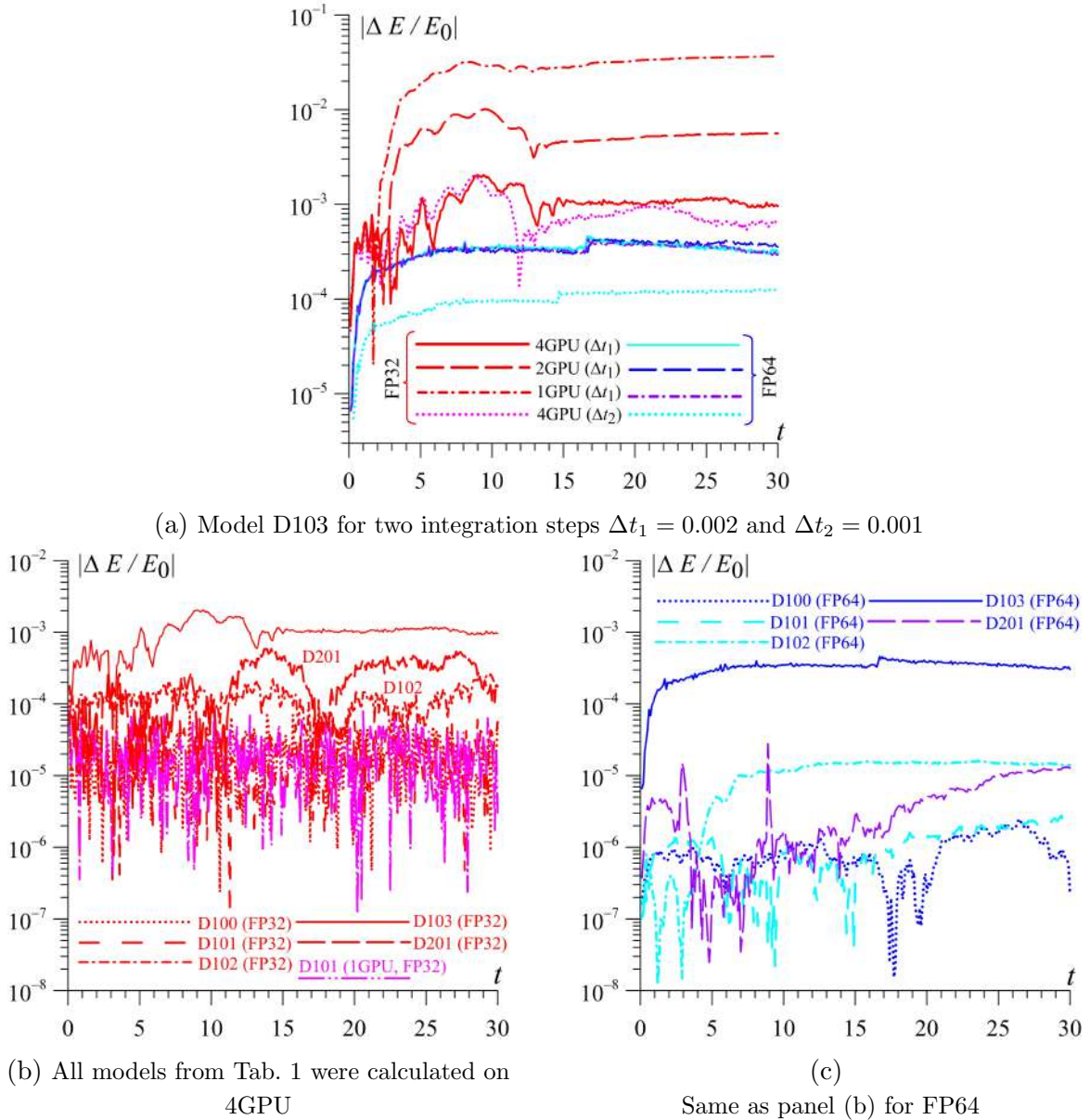
**Figure 7.** Changes in total angular momentum in the system  $\Delta L/L_0$  ( $L_0$  is the angular momentum at time  $t = 0$ )

dark halo is close to zero in our models and can then arise from tidal interactions of the halo with disturbances in the disc. Noticeable rotation of the halo can only occur in model D201 at long times after merging.

Figure 7 shows the relative changes in total angular momentum  $\Delta L/L_0$ , where  $L_0$  is the initial angular momentum. All models from Tab. 1 are calculated on 4GPU. Model D101 on 1GPU is additionally shown as a magenta curve in Fig. 7b. Conservation of angular momentum plays an important role since the disc is in the balance of primarily gravitational and centrifugal forces. Therefore, even small disturbances lead to radial imbalances of forces in the disc, which is accompanied by radial movements of the matter. Behavior of curves in Fig. 7a is generally similar to the results of calculations of  $|\Delta \mathbf{P}|$ . The evolution of  $\Delta L(t)/L_0$  for five models on 4GPU with FP32 shows that if single disc are close to the stability limit  $Q_T \simeq 1$  (models D100, D101, D102), then the relative angular momentum error does not increase and remains within  $< 10^{-4}$ . Only very cold discs or merging models give an increase in error in the case of FP32. All our models with FP64 retain angular momentum up to 13 digits.

The results of checking the law of conservation of energy are shown in Fig. 8. Total energy is less well conserved compared to momentum and angular momentum, since the velocities in the kinetic part of the energy in (8) are calculated by approximately solving the equations of motion (1). Curves  $\Delta E(t)$  in Fig. 8a describe the worst-case model D103, in which the difference in calculations using FP32 and FP64 is only 3:1 for step  $\Delta t_1 = 0.002$ . We have strong differences between the curves ( $\Delta E(t)$ ) when using 1GPU, 2GPU and 4GPU with FP32, as in the case of momentum and angular momentum in Fig. 6a, 7a. Calculations with FP64 are slightly dependent on the choice of 1GPU/2GPU/4GPU. Thus, the accumulation of error due to arithmetic rounding on short 4-byte numbers significantly depends on the number of GPUs used. Increasing the number of GPUs reduces error very effectively, bringing the results closer to DF64 calculations, which are almost independent of  $n_G$  (see Fig. 8a for calculations with FP64).

Reducing the integration step by half from  $\Delta t_1 = 0.002$  to  $\Delta t_2 = 0.001$  naturally reduces the error (compare the solid and dotted light-blue lines in Fig. 8a). This decrease is  $n_t^2 = 4$  times at the beginning of evolution in accordance with scheme (3)–(5) and then reaches 2.5 due to the

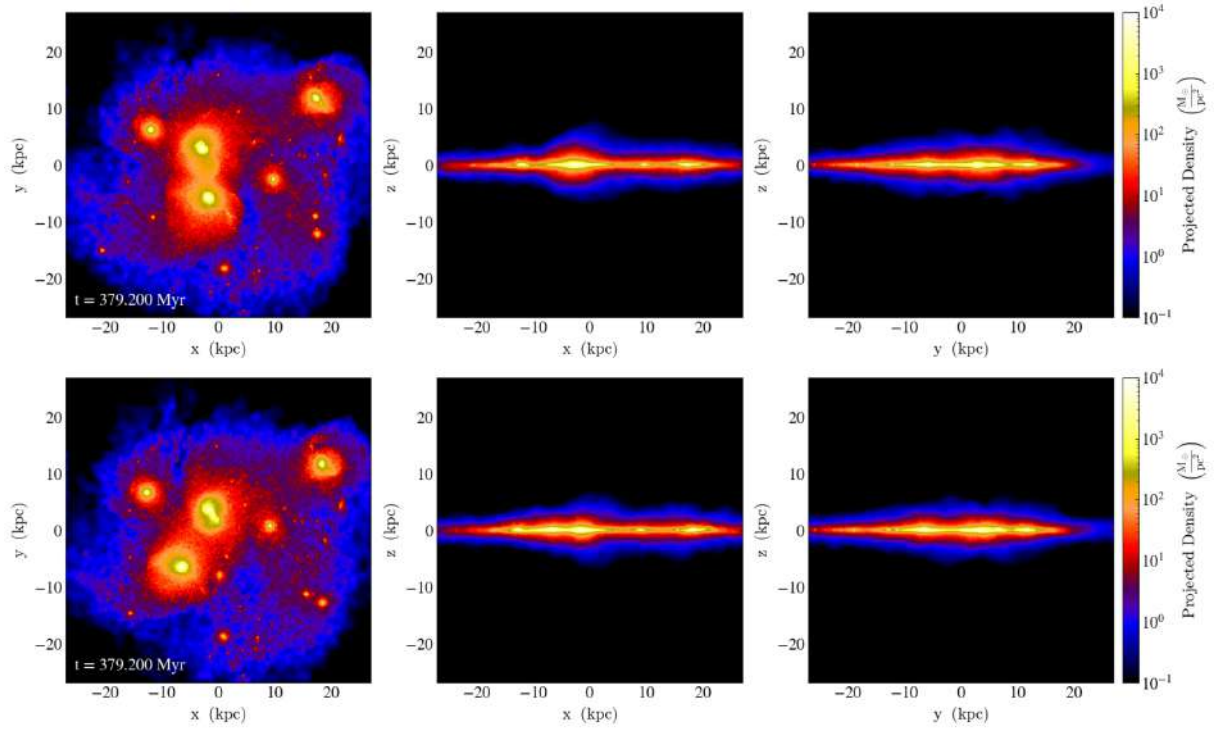


**Figure 8.** Changes in total relative energy  $\Delta E/E_0$  ( $E_0$  is the energy at time  $t = 0$ ). Model D101 on 1GPU with FP32 is shown additionally by magenta line in panel (b)

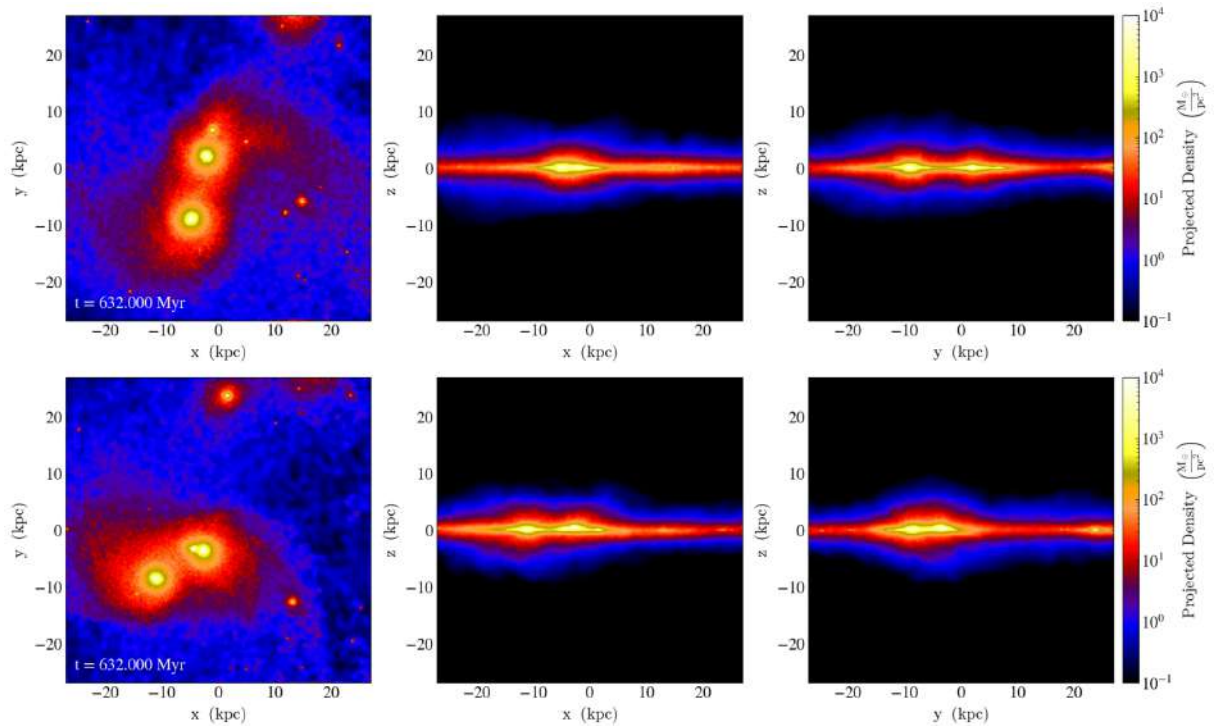
accumulation of arithmetic error in end of calculations ( $t = 30$ ). The error under consideration is determined by the order of the numerical scheme  $n_t$  and the integration step  $\Delta t$ :  $O(\Delta t^{n_t})$ . The transition from  $\Delta t_1$  to  $\Delta t_2$  in the case of 4-byte numbers almost does not reduce the relative energy error.

Models D100, D101, D102 with typical galactic spiral patterns have an energy error approximately an order of magnitude smaller for FP32 compared to model D103, all other things being equal (Fig. 8b). Calculations with FP64 give an acceptable error already at  $\Delta t_1$  (Fig. 8c).

Accumulating errors in conservation laws are reflected in the evolution of macroscopic characteristics. Figures 9, 10 compare surface density distributions in three projections, constructed in model D103 with FP32 and FP64. There are comparable differences in velocity fields, distributions of velocity dispersion components, etc. The distributions of matter along the line of sight in Fig. 9 with FP32 (bottom) and FP64 (top) give qualitatively similar structures at



**Figure 9.** Density distributions along the line of sight, demonstrating the differences in numerical solutions in model D103 for FP64 (top) and FP32 (bottom) at  $t = 6$



**Figure 10.** As in Fig. 9 at time  $t = 10$

time  $t = 379$  Myr. However, we already see noticeable differences in the positions of the density clumps and their relative orientation. These deviations quickly increase and the picture in Fig. 10 is already qualitatively different when comparing between FP32 and FP64 at time  $t = 632$  Myr.



The differences in models D100, D101, D102 are weaker, since the amplitudes of disturbances are smaller in these models compared to D103. However, the conclusion remains that it is impossible to quantitatively study galactic systems within the framework of 4-byte arithmetic.

## Discussion and Conclusion

We analyzed the implementation of the laws of conservation of momentum, angular momentum and energy in models of the dynamics of gravitationally interacting N-bodies. Such models are a traditional tool for studying globular clusters, open clusters, galaxies and galaxy clusters [1, 13, 16, 18, 25]. The system of gravitationally interacting points simulates the movements of both stars and dark matter. Our models contain both of these components. Adding gas to the model is possible when using smoothed-particle hydrodynamics, since it allows an end-to-end method for calculating gravitational forces [17].

The direct method of calculating the gravitational force by summing the contributions of all particles from each other “Particle–Particle” provides the most accurate result for a fixed number of particles  $N$ . However, some features of the organization of parallel computing on GPUs can have a significant impact on the error in  $N$ -body modeling even for an accurate method.

There are two factors that we investigated. Firstly, this is the number of significant digits. In practice, there is a choice between 4-byte and 8-byte numbers. The efficiency of operations with numbers of different lengths is very sensitive to the microarchitecture of modern GPUs. For example, the execution time of an operation with FP32 and FP64 on the V100 GPU differs by 2 times. Similar calculations on NVIDIA RTX4090 GPU differ by almost an order of magnitude. The second factor is related to the use of different numbers of  $n_G$ , in particular, calculations on 1GPU, 2GPU and 4GPU are considered, which also affects the error of long-term modeling of complex structures. Increasing the number of  $n_G$  leads to a decrease in the number of particles processed on one GPU, which in turn reduces the accumulation of error when using numbers FP32.

Graphics cards are designed for single precision arithmetic, and implementing double precision for many types of graphics accelerators requires disproportionate time resources, as is the case, for example, with the RTX4090. The considered solution to the  $N$ -body problem on RTX4070/RTX4090 with FP32 and FP64 differs by approximately an order of magnitude in execution time. Therefore, only the NVIDIA GPU Kx0/Pascal/Volta/Ampere line provides an acceptable transition to double-precision. The area of application of GPUs with FP32 are machine learning algorithms mainly [21, 28].

The law of conservation of energy always has an error due to the approximate method of integrating the equations of motion. This error can accumulate at each subsequent integration step under conditions where the number of iterations is on the order of  $10^5$ , and contributions to the gravitational force from different particles can differ by 6 orders of magnitude or more.

In principle, we can provide the laws of conservation of total momentum  $\mathbf{P}$  and total angular momentum  $\mathbf{L}$  close to the limit of arithmetic resolution at the level of 13 digits or even better. The conservation of the value  $\mathbf{P}$  is a reflection of the accuracy of the execution of Newton’s third law  $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$ . In the case of a sequential version of the program, it is easy to achieve exact fulfillment of this condition and further reduce the number of operations by 2 times thanks to optimization of the algorithm. In the case of CUDA parallelization, the requirement to satisfy

the condition  $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$  is always accompanied by an increase in computation time due to an increase in the complexity of the algorithm [4].

Thus, we highlight three main results.

1) The parallelization efficiency of the  $N$ -body – PP algorithm on multi-GPU varies between 0.8–1.2 depending on the number of particles  $N$ , the number of GPUs and the choice of single or double precision floating-point numbers. Qualitative modeling of galaxy dynamics requires the use of a number of gravitating particles  $N > 10^6$ . The efficiency of parallelizing such numerical models on a multi-GPU tends to unity.

2) The test of the laws of conservation of energy, momentum and angular momentum for the long-term evolution of gravitating systems showed a strong dependence of errors on the digits of the floating-point numbers used. The decrease in the accuracy of conservation laws for single-precision operations is due to the accumulation of arithmetic errors due to two factors. Firstly, the sum of gravitational forces from different particles contains terms that differ by several orders of magnitude, which leads to a loss of accuracy. Effective implementation of the CUDA algorithm on high-performance GPUs requires the use of a very large number of parallel threads ( $10^5$ – $10^6$ ). The execution order of these threads is determined by the built-in CUDA scheduler and cannot be determined in the source code. Secondly, studying the dynamics of galaxies over cosmological time corresponds to more than  $10^5$  integration steps, which also requires calculations with 8-byte numbers.

3) An increase in the number of GPUs used contributes to a more accurate implementation of conservation laws in the case of 4-byte arithmetic due to a decrease in the number of particles per GPU. Conservation laws in double-precision models are always fulfilled with high accuracy and do not depend on the number of GPUs.

## Acknowledgements

This work is supported by the Russian Science Foundation (grant No. 23-71-00016, <https://rscf.ru/project/23-71-00016/>). The research also relied on the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

*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. Aarseth, S.J.: Gravitational N-Body Simulations: Tools and Algorithms. Cambridge University Press, Cambridge (2009)
2. Abraham, M.J., Murtola, T., Schulz, R., *et al.*: Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1-2, 19–25 (2015). <https://doi.org/10.1016/j.softx.2015.06.001>
3. Appleton, P.N., Emonts, B., Lisenfeld, U., *et al.*: The CO Emission in the Taffy Galaxies (UGC 12914/15) at 60 pc Resolution. I. The Battle for Star Formation in the Turbulent Taffy Bridge. Astrophysical Journal 931(2), 121 (2022). <https://doi.org/10.3847/1538-4357/ac63b2>

4. Belleman, R.G., Be dorf, J., Zwart, S.F.P.: High performance direct gravitational N-body simulations on graphics processing units II: An implementation in CUDA. *New Astronomy* 13, 103–112 (2008). <https://doi.org/10.1016/j.newast.2007.07.004>
5. Binney, J., Tremaine, S.: *Galactic Dynamics*. Princeton University Press, Princeton (2008)
6. Brasser, R., Grimm, S.L., Hatalova, P., Stadel, J.G.: Speeding up the GENGA N-body integrator on consumer-grade graphics cards. *Astronomy & Astrophysics* 678, A73 (2023). <https://doi.org/10.1051/0004-6361/202347071>
7. Bruno, D., Capitelli, M., Longo, S., *et al.*: Particle kinetic modelling of rarefied gases and plasmas. *Plasma Sources Science and Technology* 12(4), S89 (2003). <https://doi.org/10.1088/0963-0252/12/4/024>
8. Eckmann, J.P., Hassani, F.: The detection of relativistic corrections in cosmological N-body simulations. *Celestial Mechanics and Dynamical Astronomy* 132(2) (2020). <https://doi.org/10.1007/s10569-019-9943-z>
9. Fedorov, V.A., Kholina, E.G., Gudimchuk, N.B., Kovalenko, I.B.: High-performance computing of microtubule protofilament dynamics by means of all-atom molecular modeling. *Supercomputing Frontiers and Innovations* 10(4), 62–68 (2023). <https://doi.org/10.14529/jsfi230406>
10. Greengard, L.: The Numerical Solution of the N-Body Problem. *Computers in Physics* 4, 142–152 (1990). <https://doi.org/10.1063/1.4822898>
11. Grigoriev, F.V., Sulimov, V.B., Tikhonravov, A.V.: Study of thin optical films properties using high-performance atomistic simulation. *Supercomputing Frontiers and Innovations* 11(1), 97–108 (2024). <https://doi.org/10.14529/jsfi240105>
12. Hopkins, P.F., Nadler, E.O., Grudic, M.Y., *et al.*: Novel conservative methods for adaptive force softening in collisionless and multispecies N-body simulations. *Monthly Notices of the Royal Astronomical Society* 525(4), 5951–5977 (2023). <https://doi.org/10.1093/mnras/stad2548>
13. Ishchenko, M., Kovaleva, D.A., Berczik, P., *et al.*: Star-by-star dynamical evolution of the physical pair of the Collinder 135 and UBC 7 open clusters. *Astronomy & Astrophysics* 686, A225 (2024). <https://doi.org/10.1051/0004-6361/202348978>
14. Kamlah, A.W.H., Leveque, A., Spurzem, R., *et al.*: Preparing the next gravitational million-body simulations: evolution of single and binary stars in NBODY6++GPU, MOCCA, and MCLUSTER. *Monthly Notices of the Royal Astronomical Society* 511(3), 4060–4089. <https://doi.org/10.1093/mnras/stab3748>
15. Khan, R., Kandappan, V.A., Ambikasaran, S.: HODLRdD: A new black-box fast algorithm for N-body problems in d-dimensions with guaranteed error bounds: Applications to integral equations and support vector machines. *Journal of Computational Physics* 501, 112786 (2024). <https://doi.org/10.1016/j.jcp.2024.112786>
16. Khoperskov, A.V., Khrapov, S.S., Sirotin, D.S.: Formation of transitional cE/UCD galaxies through massive disc to dwarf galaxy mergers. *Galaxies* 12(1), 1 (2024). <https://doi.org/10.3390/galaxies12010001>

17. Khrapov, S.S., Khoperskov, A.V.: Retrograde infall of the intergalactic gas onto S-galaxy and activity of galactic nuclei. *Open Astronomy* 33(1), 20220231 (2024). <https://doi.org/10.1515/astro-2022-0231>
18. Khrapov, S.S., Khoperskov, A.V., Zaitseva, N.A., *et al.*: Formation of spiral dwarf galaxies: observational data and results of numerical simulation. *St. Petersburg State Polytechnical University Journal. Physics and Mathematics* 16(1.2), 395–402 (2023). <https://doi.org/10.18721/JPM.161.260>
19. Li, Y., Pinto, M.C., Holderied, F., *et al.*: Geometric Particle-In-Cell discretizations of a plasma hybrid model with kinetic ions and mass-less fluid electrons. *Journal of Computational Physics* 498, 112671 (2024). <https://doi.org/10.1016/j.jcp.2023.112671>
20. Liseykina, T.V., Dudnikova, G.I., Vshivkov, V.A., *et al.*: MHD-PIC Supercomputer Simulation of Plasma Injection into Open Magnetic Trap. *Supercomputing Frontiers and Innovations* 10(3), 11–17 (2024). <https://doi.org/10.14529/jsfi230302>
21. Navarro, C.A., Hitschfeld-Kahler, N., Mateu, L.: A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures. *Communications in Computational Physics* 15(2), 285–329 (2014). <https://doi.org/10.4208/cicp.110113.010813a>
22. Ong, B.W., Dhamankar, S.: Towards an Adaptive Treecode for N-body Problems 82, 72 (2020). <https://doi.org/10.1007/s10915-020-01177-1>
23. Rantala, A., Naab, T., Rizzuto, F.P., *et al.*: Bifrost: simulating compact subsystems in star clusters using a hierarchical fourth-order forward symplectic integrator code. *Monthly Notices of the Royal Astronomical Society* 522(4), 5180–5203 (2023). <https://doi.org/10.1093/mnras/stad1360>
24. Romeo, A.B., Horellou, C., Bergh, J.: N-body simulations with two-orders-of-magnitude higher performance using wavelets. *Monthly Notice of the Royal Astronomical Society* 342(2), 337–344 (2003). <https://doi.org/10.1046/j.1365-8711.2003.06549.x>
25. Smirnov, A.A., Sotnikova, N.Y., Koshkin, A.A.: Simulations of slow bars in anisotropic disk systems. *Astronomy Letters* 43(2), 61–74 (2017). <https://doi.org/10.1134/S1063773717020062>
26. Voevodin, V.V., Chulkevich, R.A., Kostenetskiy, P.S., *et al.*: Administration, Monitoring and Analysis of Supercomputers in Russia: a Survey of 10 HPC Centers. *Supercomputing Frontiers and Innovations* 8(3), 82–103 (2021). <https://doi.org/10.14529/jsfi210305>
27. Yokota, R., Barba, L.A.: *Treecode and Fast Multipole Method for N-Body Simulation with CUDA*. Springer (2011). <https://doi.org/10.1016/B978-0-12-384988-5.00009-7>
28. Zhang, H., Si, S., Hsieh, C.J.: Gpu-acceleration for large-scale tree boosting. Eprint arXiv 1706.08359 (2017). <https://doi.org/10.48550/arXiv.1706.08359>
29. Zhou, K., Liu, B.: *Molecular Dynamics Simulation: Fundamentals and Applications*. Elsevier, Amsterdam (2022)

# Investigation of the Capability of Restoring Information on the Primary Particle from Cherenkov Light Generated by Extensive Air Showers Using the Lomonosov-2 Supercomputer

*Elena A. Bonvech*<sup>1</sup> , *Clemence G. Azra*<sup>1,2</sup> , *Olga V. Cherkesova*<sup>1,3</sup> ,  
*Dmitriy V. Chernov*<sup>1</sup> , *Elena L. Entina*<sup>1</sup>, *Vladimir I. Galkin*<sup>1,2</sup> ,  
*Vladimir A. Ivanov*<sup>1,2</sup>, *Timofey A. Kolodkin*<sup>1,2</sup> ,  
*Natalia O. Ovcharenko*<sup>1,2</sup> , *Dmitriy A. Podgrudkov*<sup>1,2</sup> ,  
*Tatiana M. Roganova*<sup>1</sup> , *Maxim D. Ziva*<sup>1,4</sup> 

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

The new SPHERE-3 detector is under development. Its main objectives are the primary cosmic ray spectrum and chemical composition studies in the 1–1000 PeV energy range. The detector will register both reflected and direct Cherenkov light from extensive air showers. The goal of the new approach is high precision of event-by-event estimation of the primary particle parameters, especially its mass. The reflected Cherenkov light registration technique used in earlier experiments has good energy sensitivity and some mass estimation capability. Addition of direct Cherenkov light registration will allow to further advance the detector mass sensitivity. Several approaches to direct Cherenkov light registration are considered: by the main detector camera and by a dedicated direct light detector. First tests of the proposed methods are presented both for reflected and direct Cherenkov light. The detector design is tested on a large database of simulated showers. The simulation pipeline and related challenges to it are described. Also, progress in parallelization of the CORSIKA code for Cherenkov light simulations is presented.

*Keywords:* Cherenkov light, primary cosmic rays, supercomputer Lomonosov-2, extensive air showers, air-borne telescope.

## Introduction

Chemical composition of primary cosmic rays is crucial for the understanding of their sources, acceleration and transport processes. However, despite decades of studies, the precision of available data, especially at high energies, is somewhat insufficient.

Primary cosmic rays with energies above  $10^{15}$  eV are studied via indirect methods using large ground-based detector arrays using Earth's atmosphere as a natural calorimeter. Upon entry into the atmosphere a primary cosmic ray particle starts to interact with air initiating a cascade of secondary particles: an extensive air shower (EAS). EAS development in the atmosphere is accompanied by different electromagnetic phenomena, including Cherenkov light (CL) emission. In 1974 A.E. Chudakov proposed to register reflected CL from EAS [10]. In line of his idea we developed a series of SPHERE detectors [4]. The SPHERE-2 telescope operated in 2011–2013 and proved the validity and applicability of this method. Analysis of the results of the SPHERE-2 experiment gave an energy spectrum and chemical composition of primary cosmic rays in the 5–500 PeV energy range [5].

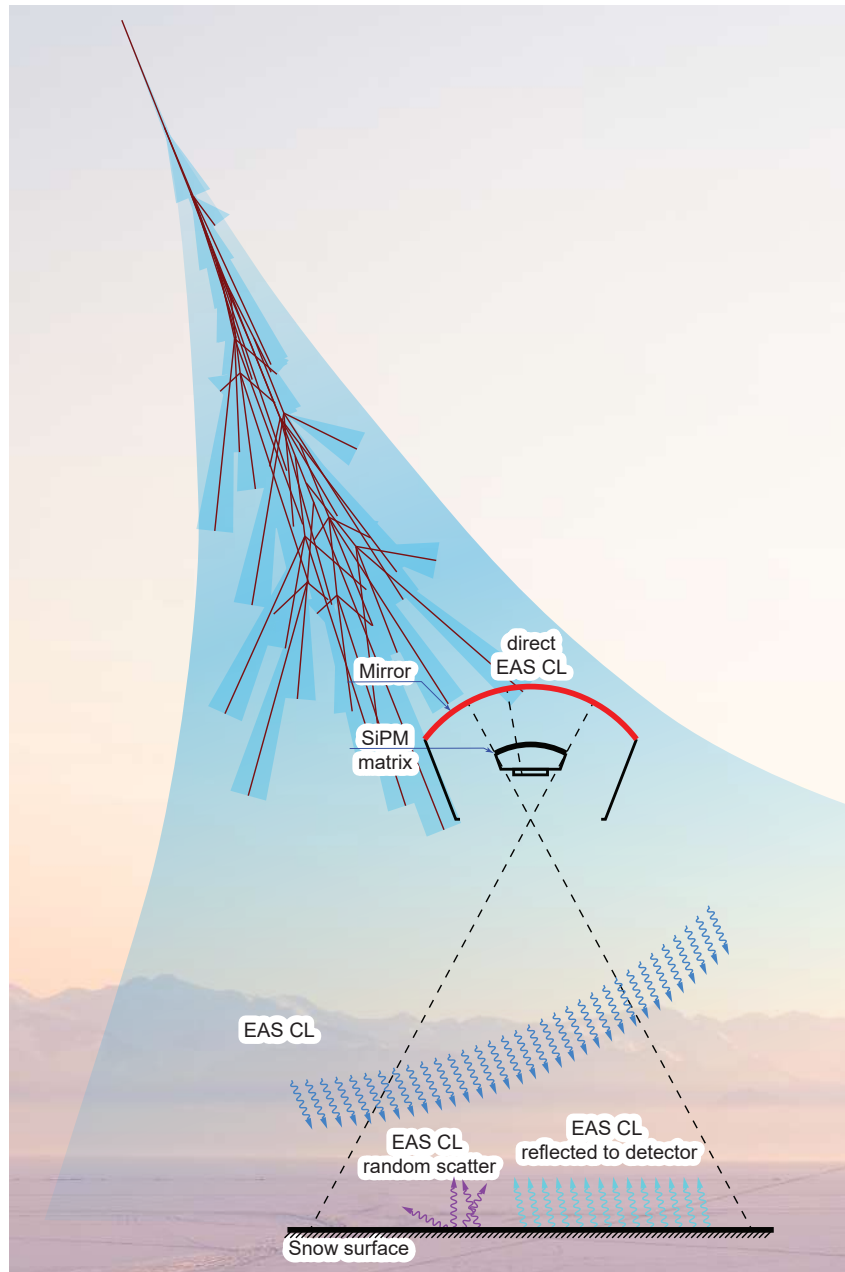
Currently, a new SPHERE-3 air-borne telescope aimed at reflected EAS CL registration is being designed. An unmanned aerial vehicle (UAV) is intended as a carrier for the detector. The proposed experimental setup is presented in Fig. 1.

<sup>1</sup>Skobeltsyn Institute for Nuclear Physics, Lomonosov Moscow State University, Moscow, Russian Federation

<sup>2</sup>Lomonosov Moscow State University, Physics Department, Moscow, Russian Federation

<sup>3</sup>Lomonosov Moscow State University, Department of Cosmic Research, Moscow, Russian Federation

<sup>4</sup>Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Moscow, Russian Federation



**Figure 1.** SPHERE-3 experiment scheme

The article is organized as follows. Section 1 is devoted to the SPHERE-3 detector description. The registration technique advantages, challenges and limitations are discussed. In Section 2 we present the simulations pipeline that allowed us to obtain data described and analysed later in the text. Section 3 reports on current progress in creation of parallel CORSIKA version that works with CL. In Section 4 we present first approaches for arrival direction and shower axis location estimation methods using reflected and direct EAS CL (independently from each other at this stage). Section 5 is devoted to current progress in our approaches to EAS primary particle energy and mass estimation, again independently, based on direct CL data and indirect CL data. Conclusion summarizes presented results and points directions for further work.

## 1. SPHERE-3 Telescope

The SPHERE-3 detector shares the same optical scheme with its predecessors: a Schmidt camera. The main mirror of the detector will be approximately 2.2 m in diameter and most probably segmented (there are some limitations in the different coating technologies). A corrector plate, absent in the previous SPHERE detectors, this time will be added, since the detector's light sensitive camera is planned to have a much higher resolution.

Substitution of the previous detectors photo-multiplier tubes by silicone photo-multipliers (SiPM) will allow to gain higher resolution and sensitivity of the detector camera, with lower mass and power consumption. The camera will consist of 379 optical modules with 7 SiPM each, 2653 SiPM in total. Higher resolution will give better precision for the reflected EAS CL distribution registration. Lower mass of the sensitive camera will allow a larger detector to be lifted on the UAV, besides, this less power consumption will allow more measuring channels, since each measuring channel with its electronics consumes power, as well as higher overall autonomy of the detector.

The planned flight altitudes are from 500 to over 2000 m. However, the exact measurement program will be corrected after more studies on the best flight parameters are done, since data registered at each altitude can yield information on a different energy region. The weather is also a factor to be accounted for.

### 1.1. Direct Cherenkov Light: EAS Primary Particle Parameters Estimation

Since earlier in the SPHERE project only the reflected EAS CL was registered some comments on why it was decided to include direct CL and what is to be achieved from analyzing its properties are needed.

All present day EAS detectors register only direct CL (if they register CL at all). SPHERE experiments were an exception, thus requiring us to develop new methods and approaches for data analysis. For direct EAS CL registration with ground-based detectors there exists an established set of methods with known capabilities and virtually no possibility to add some radically new approach. The primary particle energy is usually determined using some form of total CL photon count or by using photon density measures at some distance from the shower axis (150–200 m) [8, 16], where fluctuations are considered to be relatively small. The EAS arrival direction is reconstructed using delays between CL registration by several stations of the detector array or, alternatively, CL angular distribution can be studied at high resolution [6]. In this case, the maximum (or center of mass) of this distribution will indicate the shower arrival direction. However, this approach exhibits a systematic shift depending on the distance to the shower axis, shower zenith angle, primary particle energy and particle mass. But, the most difficult problem is the primary particle mass estimation. The generally used approach, first, requires to estimate the depth of the shower maximum  $X_{max}$  derived from the steepness of the lateral distribution function (LDF) for CL photons or from the CL pulse width at significant distances (above 300 m) from the shower axis. Then, using  $X_{max}$  and basing on the average model distributions with respect to primary particle energy and shower arrival direction the primary particle mass is defined. The pros and cons of this approach are analyzed in detail in earlier publications [11, 12]. Our current approach estimates the primary particle mass from the parameters of the CL distributions (spacial [11, 13] and angular [9]), making the resulting

criteria more integral in a sense that they utilize the whole distribution rather than its small part, with a low dependence on the used interaction model as a bonus.

The idea to introduce direct CL registration to the SPHERE-3 telescope came from two major sources: its accidental registration by the SPHERE-2 telescope [7] and transition from a balloon to an UAV as a carrier. *A priori* the idea does not look promising, since the limitations of the UAV platform are quite severe. However, first estimations show that even extremely small (compared to imaging air Cherenkov telescopes) direct CL detectors allow to obtain independent estimations of primary particle parameters, which is, arrival direction and mass. Especially interesting becomes the possibility of simultaneous registration of both direct and reflected CL from the same EAS allowing for two independent estimations of the shower primary parameters.

A direct CL detector alone cannot provide data for LDF analysis, only angular and temporal information is available for registration. But, the analysis of temporal data on EAS CL requires the detector to have nanosecond resolution and precise external knowledge on the shower axis location. And even fulfilment of these requirement does not guarantee any viable precision of primary mass reconstruction. On the other hand, it is known, that the CL angular distribution does hold information on the EAS longitudinal development [6], what makes it sensitive to the primary particle mass. As it will be shown below, even basic parameters of the angular distribution are sufficient for this task and allow to estimate the EAS arrival direction, but for best results they should be combined with reflected CL data. This brings us back to the idea of simultaneous registration of direct and reflected EAS CL. The SPHERE-3 detector construction and measurements strategy should be optimized so that most of the registered showers will contain data on both direct and reflected CL.

## 1.2. Limitations on Detector Location Relative to the Shower for Direct Cherenkov Light Registration

From the telescope target energy range of 1–1000 PeV and reasonable mass and size of the direct CL detector (a kilogram in mass and around 100 cm<sup>2</sup> area) come the limitations on shower core distance. At large distances, there will be not enough photons to reliably register and then reliably reconstruct the relevant distribution parameters. For the lower end of the target energy range (about 1–3 PeV) the distance to the shower core is estimated to be 100–200 m. Arrival direction estimates are possible in a broader range, but mass estimation at lower distances requires higher detector resolution as the shower angular images become smaller and axially symmetrical, while at larger distances the detector area should be bigger in order to collect enough photons.

## 1.3. Limitations of Detector Altitude for Direct Cherenkov Light Registration

The direct CL flux at a fixed distance from the shower core depends on the altitude: roughly weakening twice going from 500 to 2000 m. In the same manner, the size of the CL image shrinks with altitude. This leads to a higher energy threshold and higher required angular resolution of the detector to provide data for reliable primary mass estimation. Calculations show that the altitudes between 500 and 1500 m are the most favourable for our detector design. The expected fraction of EAS with both direct and reflected CL data will be around 0.3.

The properties of the angular distributions from direct CL obtained using the CORSIKA code, as well as the characteristics of images from a toy-model of a direct CL detector consisting



of a lens and a large area position-sensitive sensor were studied. The task was to separate EAS by their primary mass and estimate their arrival direction. Each of the tasks was solved separately for the angular distributions and for detector images. The careful analysis of model angular distributions allows to set the upper limits on the accuracy of parameter estimates in each of the two tasks, which will then be needed for the detector design. Solving these problems for a toy-model detector brings the accuracy closer to real values.

## 2. Calculations Pipeline

The design and optimization of primary cosmic rays optical detectors is a computationally heavy and complex procedure. This task includes not only optimization of the optical part (a telescope) that needs to meet certain criteria such as area, available materials and some specific needs that are significantly different from other optical instruments, but also includes simulation of EAS development, registration process, electronics response and data analysis procedure. And only in the end the final result can be evaluated to some extent. For common optics, there are parameters of the image that are crucial for later analysis. In case of EAS studies these parameters are relatively unknown. It is always a decision between larger entry window and larger field of view, between better resolution and limitations of light sensors, between more pixels and limitations on how much data can be digitized, stored and later processed. All this makes it virtually impossible to divide the optimization task into separate stages.

Thus, the detector optimization loop, therefore, includes a full Monte Carlo simulation of the EAS development process. The number of particles traced in the simulation is proportional to the primary particle energy. A single shower from a very high energy particle may take weeks to simulate (however, this is rarely done for obvious reasons). But, even with relatively low energies detector optimization requires to account variations in EAS development conditions (the atmosphere is not constant and weather changes affect the simulation outcome) and different high energy hadron interaction models. In a perfect situation, an optimized detector should not be sensitive to the model change, however, this is impossible to achieve by construction changes only. A combination of detector construction and data analysis procedures may solve such a task. As it was shown for the SPHERE-2 reflected CL data, there is a way to lower the influence of the nuclear interactions model uncertainty for this detector by careful data processing [17].

On top of a long optimization loop that includes EAS development, light collection simulation, data handling and evaluation of the obtained results (which usually are based on statistical methods) there is another simulation step in the middle that introduces additional uncertainties: the light registration process itself. The expected number of photons reaching the detector in the end is relatively small, the light registration process in the sensor should also be a full Monte Carlo simulation, since fluctuations are high. These simulations should account for the sensitive element operation. Both PMTs from the earlier experiments and SiPMs in the current detector rely on cascade amplification from a single photoelectron to the final anode current pulse. Beside this, there are side-effects of amplification, digitization, background and various sorts of interferences (however, the last are easier to exclude from the optimization process and try to avoid or to compensate *post factum* in real data).

The electronics design for the detector is also not set in stone and allows for parameter selection such as amplification and dynamic range, digitization frequency and event record length (and how deep a buffer can be). These decisions may affect the trigger system logic and detector energy thresholds. The overall detector optimization procedure may even include a task to find

the minimal viable parameters of the electronic systems to reach the overall detector goals, which are the primary cosmic rays energy spectrum and mass composition studied with reasonable event statistics behind each data point.

This gargantuan problem even for only direct or only reflected CL requires a careful approach and tremendous computational resources unavailable outside the supercomputer framework. Trying to optimize for both simultaneously just adds complexity to the task. On the bright side, we can directly estimate the best possible results obtainable with the detector if we exclude the background and electronics from consideration. This will allow to check if it is possible to obtain certain results given the natural fluctuations in EAS development. Inclusion of the background and electronics will affect the results, but we will have a starting point in detector shape and key parameters to track. Also, the first approximation for data analysis can be obtained from this data.

Also, in the search for such analytical methods and approaches neural networks may come in handy. In the recent years, neural networks find more and more use in scientific research. However, their introduction to a new field goes with comparison to traditional methods and techniques, since they are quite sensitive to noise in the data (and even to the type of noise) while not providing transparent (for the scientific community who are yet not quite accustomed to them) indications and measures of success or failure. They also require careful data preprocessing and large statistics to train upon, what brings back the computational burden (plus the computational resources for neural network training).

To approach the solution of this complex problem we decided to unify and standardize approaches and procedures. The SPHERE-3 software combines several modelling and data processing stages into a single calculations pipeline. Since the main aim of the current stage of SPHERE project is the experiment and detector design optimization such approach allows smooth recalculation from any point (except, probably, the very first step as it is very time-consuming).



**Figure 2.** SPHERE calculation pipeline

The first step in the calculations was EAS simulation using the CORSIKA 7.5600 code [14] with two different models of high-energy hadron interactions QGSJET01 [15] and QGSJETII-04 [18]. At this step, a set of simulation parameters was chosen, that included energy, zenith angle, primary particle type, atmosphere model. Simulation results were saved for each simulated EAS as a separate spatio-temporal distribution on the snow and a set of angular distributions for different observation altitudes and shower core distances. These simulated EAS formed a data base that was used for all consecutive simulation steps.

Next, in order to boost the statistics and save time each simulated shower was “cloned” 100 times with random shifts in axis location relative to the detector to simulate photons reaching the telescope entry window. Photon arrival times are updated respectively to the travelled distances. This step allows to use more data from the shower on later steps and to study the precision of the registration system and reconstruction procedures.

The next step involves tracing individual photons inside the detector from the entry window to the registration surface of the light sensitive elements. This step was done using Geant4 [1–3]. Detector elements, due to their complexity, were modelled as independent parts, saved as STL

files and imported into Geant4 using CADMesh [20]. This approach simplifies the modelling process, keeping the geometry precise and allowing flexibility in detector geometry manipulations, since no code needs to be updated, only the geometry files. Geant4 in its run transports the photons inside the detector with account for all reflections and scattering until the photon is absorbed (on the light sensitive surface or on some not absolutely reflective surface) or leaves the detector back through the entry window. Photons that hit the sensitive surface are recorded with their arrival time and other properties.

Next comes the electronics response simulation (if needed). At this step, the list of photons at the detector SiPM matrix is padded (if necessary) with background photons and converted to the output of the data acquisition system (DAQ) readout. Since the detector is at the earliest stages of development, no real electronics exist. But, an early prototype of the electronic system was tested. A small scale SiPM matrix was tested as part of the Small Imaging Telescope in Tunka Valley [19] and a DAQ prototype was designed and tested in the lab. The results of these tests were used for approximation of the SPHERE-3 electronics properties (amplification, SiPM behaviour, amplification effects on SiPM output pulse profile etc.). Optionally, other electronics effects can be applied such as non-linearity, baseline fluctuations, noise, buffer shift etc. The output of this stage closely approximates the expected output of a real detector and can be used to test various approaches for trigger system design, data processing techniques, etc.

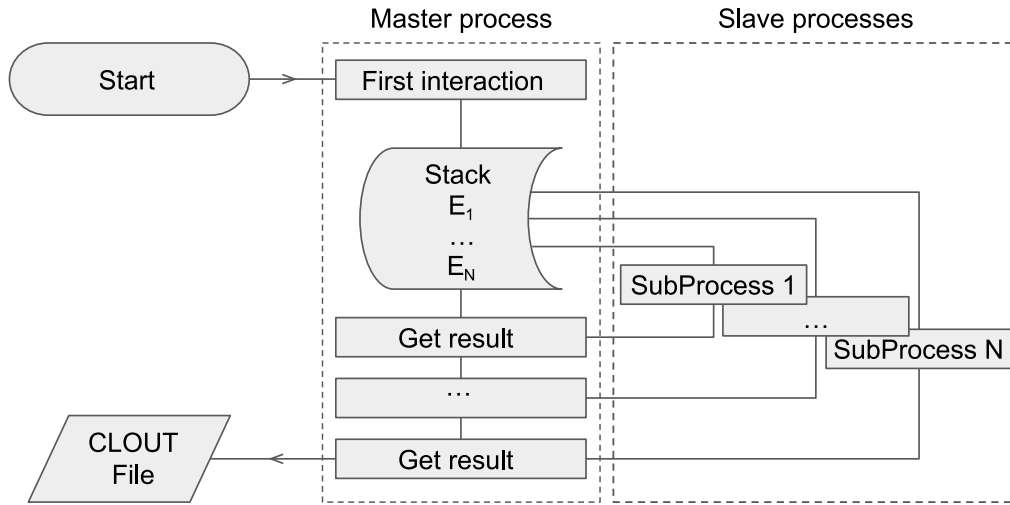
For convenience, all of the above mentioned steps were tied together into a unified pipeline using a Python application. This application allowed to select the desired simulation parameters (energy, zenith angle, nuclei, detector position, background etc.) and run the required simulation steps documenting the results in the process. Also, at any moment later, the pipeline may be expanded with any number of new steps allowing to include it naturally into the detector optimization loop.

### 3. Parallelized CORSIKA with Cherenkov Light Generation

The foundation of the simulations for the SPHERE project is a CORSIKA code (the standard for EAS simulations). While an option to compile a version of CORSIKA for parallel systems exists, it will only generate charged particles, and the authors of the CORSIKA code do not plan on making a parallel version with the CL option due to lack of resources. A single simulation of EAS with CL for a primary particle of 100 PeV will require at least 37 hours to complete, if all goes fast. If not (this is a Monte Carlo simulation through and through), it will take the simulation almost to the limit of 48 hours set for a job on most of the supercomputer systems. Thus, the required simulation precision makes it impossible to reach the top of the desired energy range (1–1000 PeV). In this situation, in order to perform simulations for even higher energies there are two options: to run simulations on local machines with no limitations or to create a in-house parallel version of CORSIKA using MPI libraries.

This first option severely limits the amount of showers that can be simulated at a reasonable time. Therefore, we started to develop a parallel version of the CORSIKA code. Since there was no aim at a generalized universal conversion of the code which would be compatible with other options of the software (and there are numerous options that provide required flexibility to the code) and there was no need to keep some of the CORSIKA base functionality (like charged particles distribution over the observation level, individual particle parameters, shower profile statistics etc.), only the required data on CL was to be kept and stored in form of angular-spatio-temporal distributions at certain levels. Along with the CORSIKA code native logic of keeping

a sort of a particles calculation queue (in the form of a particle stack) and physics that insure that the first interactions produce particles with highest energies, the parallelization scheme is rather simple (see Fig. 3).



**Figure 3.** Parallel CORSIKA operation scheme

Thus, a single job contains a separate task (Start block) that prepares every required bit of data for later processes (atmospheric parameters, cross-sections, particle tables and alike) and parameters of the event to simulate (primary particle, its start location, first interaction point, momentum etc.). All of this info is stored in a file. At the next stage, the master-process reads this file and starts tracking the first particle through the atmosphere where through the first interactions a stack of particles to be traced is formed. At the same time, a set of slave-processes is started that reads the same data file as the master-process and awaits for the data on the particles to trace from the master-process. The master-process distributes the stack of secondary high-energy particles and goes into listening mode. The slave-processes upon receiving their portions of the particles stack perform subshower simulations for those particles and send the collected data on CL photons to the master-process, which, in turn, aggregates the received data. When all slave-processes finish their tasks the master-process produces a file (CLOUT) and finishes the job.

At the moment the task of CORSIKA parallelization is at the stage of collecting the data required for subshower simulations to run and particle stack distribution algorithm design.

So far the base EAS CL distributions set from CORSIKA includes two separate runs (slightly different versions and distributions parameters) over 6 and 4 nuclei (first and second runs respectively) initiating showers with 3 different energies (however, more are to be simulated once the parallel CORSIKA version will be ready) in 4 different atmosphere models, 2 nuclear interaction models and 6 zenith angles. With each set of parameters 100 showers were modelled (total of 144 000 EAS) that alone took more that 140 000 node-hours to compute on the Lomonosov-2 supercomputer [21] (roughly an node-hour per EAS, but these results are for low energy EAS). At least about the same time will be needed to finish the CORSIKA simulation runs to get the full required set of energies.

But, even on this set of modelled EAS CL distributions the first results using basic approaches and previously tested methods were obtained.

## 4. Axis and Direction

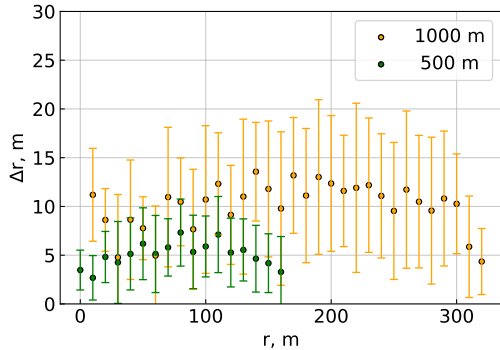
Axis location and shower arrival direction are the basic information that is reconstructed in EAS data analysis. The arrival direction can be estimated using both direct and reflected CL data.

### 4.1. Axis and Direction by Reflected Cherenkov Light

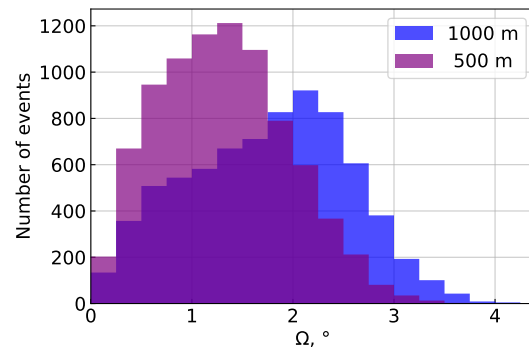
Reconstruction procedures were developed and tested on model events (without electronics response and background) from primary protons, nitrogen and iron nuclei with an energy of 10 PeV that arrived at  $15^\circ$ . Two detector flight altitudes were analysed: 500 and 1000 m.

Axis location was reconstructed from reflected CL data, at this stage, as a simple maximum search. During the first stage, an event from EAS is located within the recorded data frame (same as for the previous detector in the series, SPHERE-2) as a maximum in the time series of the total signal across all pixels. The assumption here is that while the amount of EAS CL photons is low relative to the expected background, they arrive in tight pulse and locally greatly exceed the average background values.

In the located event window individual pixels are analyzed, maximums in their time series are taken as data points (time and value). Across all values the pixel with the maximum value is identified. The data is then re-projected onto the snow and pixels are assigned coordinates with respect to the detector flight altitude. For higher precision weighted results from adjacent pixels are used. This is a simple procedure, but gives a relatively good precision – around 5 m for detector flight altitude of 500 m and around 10 m for 1000 m (see Fig. 4).



**Figure 4.** Axis location precision for different flight altitudes,  $\Delta r$  represents the distance between reconstructed and true axis locations



**Figure 5.** Arrival direction reconstruction precision using reflected CL data for different flight altitudes.  $\Omega$  is the angle between the reconstructed and true EAS arrival directions

Shower arrival direction was reconstructed from the same data pulses. Since the shower CL component is a thin (only few meters thick in its central part) slightly curved disk that effectively intersects the observation level, the light falls onto the ground as a thin line, travelling with a certain speed across the observation field. Time delays between the pulse's appearance in each pixel (corrected by the optical path length to the detector) form a shower front in the  $(x, y, ct)$

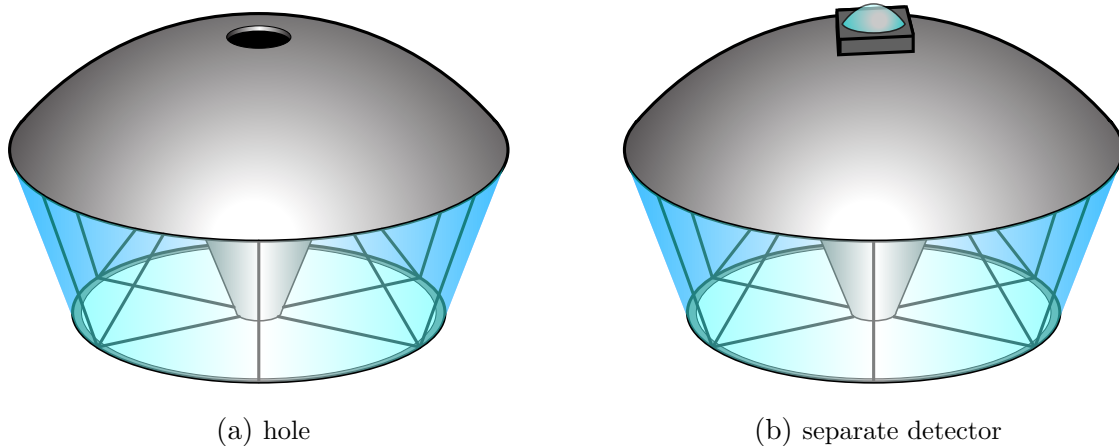
coordinates. This front was fitted by a parabolic function (in a EAS reference frame it forms a paraboloid of revolution around the shower axis):

$$t_i = a_0 + a_1 R(\phi, \theta) + a_2 R^2(\phi, \theta), \quad (1)$$

where  $R$  is the distance from the shower axis in a EAS reference frame,  $\phi$  and  $\theta$  are shower angles and  $a_i$  are the free coefficients, their dependencies on shower parameters are yet to be studied. The precision of this method of EAS arrival direction estimation is around  $1\text{--}2^\circ$  (see Fig. 5). Since the model EAS sample used in this initial study was small and all showers had the same zenith angle, the systematic uncertainties of this method are not yet studied.

## 4.2. Direct Cherenkov Light Arrival Direction

The proposed construction of the SPHERE-3 detector, specifically its UAV carrier, allows for two separate ways of direct CL registration: first, with the main SiPM mosaic through the hole (or a set of pinholes arranged into a coded aperture) in the main mirror (see Fig. 6a); second, with a separate compact detector (see Fig. 6b). Both have their pros and cons in capabilities, operation, procedures etc. This project is aimed, among other things, at selecting of the direct CL registration method.



**Figure 6.** SPHERE-3 versions with different approaches for direct CL registration

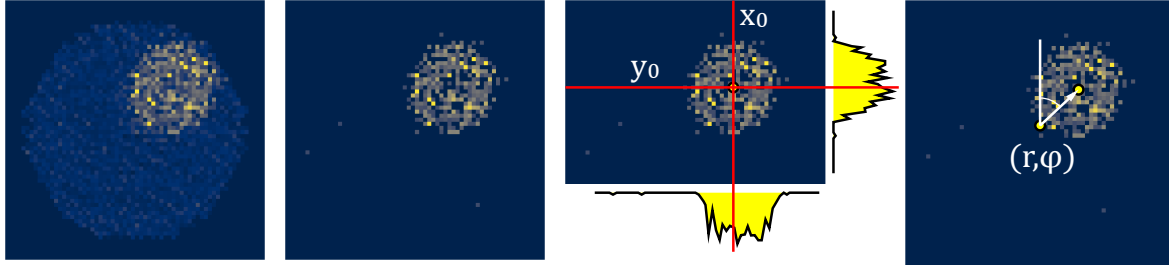
### 4.2.1. Main telescope mosaic approach

The difference in direct and reflected CL fluxes is so large that if a hole (or set of holes) configuration is used for direct light registration less than a 1% of the reflected aperture is needed, so the capabilities of the main mirror will not be affected.

As this is a work in progress, the simplest possible option was tested first – a large hole in the center of the mirror (the area of which is not involved in the reflected CL collection in any case). In the future, a more complex options may be evaluated for more data on other primary particle characteristics.

For this analysis a set of artificial events emulating EAS with a fixed brightness and varying angular distributions of zenith angles in a  $6\text{--}17^\circ$  range (azimuth angles were random) was used. These photons were transported through the detector and electronics response modelling. No

background was added, but random reflections and scattering of CL photons on various elements were accounted for. In case of an event (by the same method as was used for the reflected CL event location procedure described above) in each pixel the sum of a time series over a certain window was taken as the pixel value. Knowing the location of each pixel, this allows to construct an image (see Fig. 7) of the event.

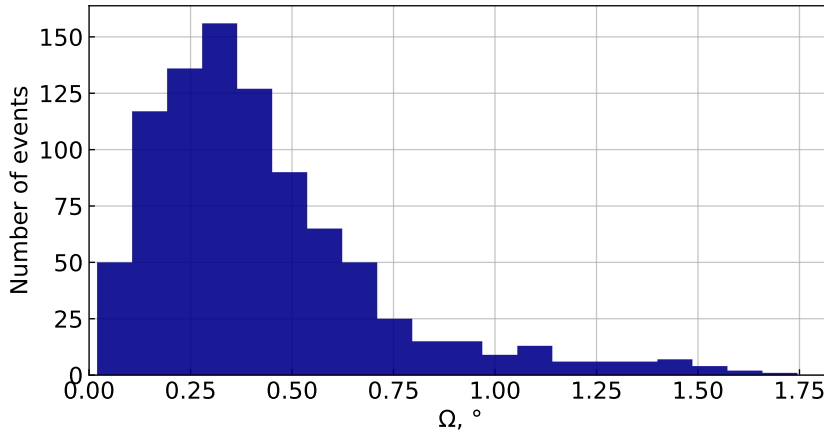


**Figure 7.** Direct CL images processing steps (from left to right): original image, cleanup, location of the center of the spot, feature determination

The images were then processed in the following steps:

- cleanup using the median value as a threshold, everything below is zeroed;
- the resulting spot's center was estimated as the center of mass;
- feature formation – distance between the center of the spot and the image center and arctangent of the spot's relative coordinates;
- a fully connected neural network with one hidden layer was used to reconstruct the true light arrival direction.

The resulting precision of the arrival direction determination was  $0.43^\circ \pm 0.29^\circ$  (see Fig. 8).



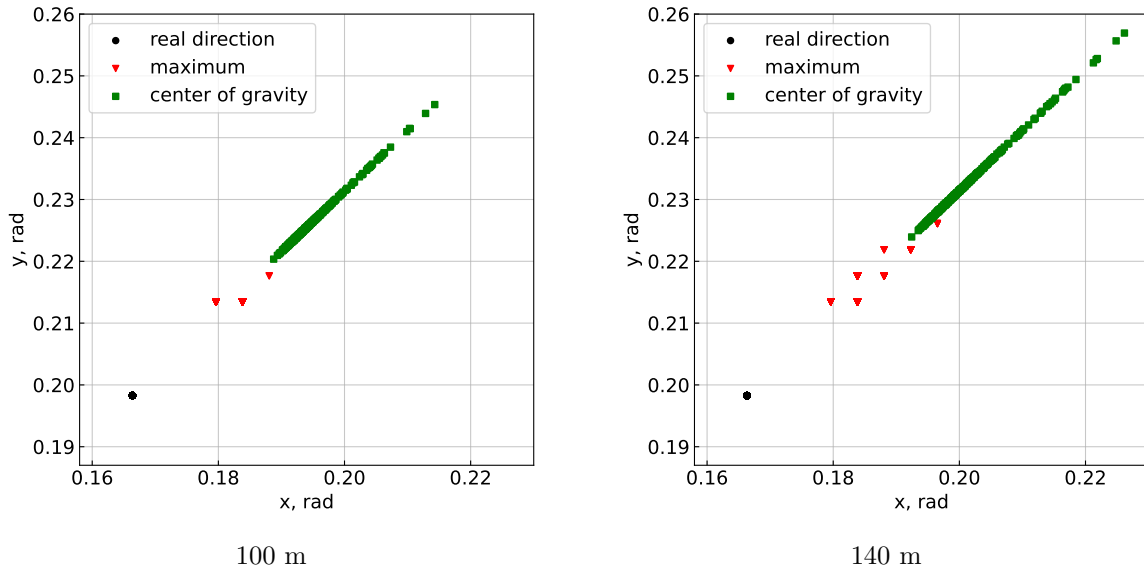
**Figure 8.** Arrival direction reconstruction precision using a neural network.  $\Omega$  is the angle between the reconstructed and true light arrival directions

#### 4.2.2. Using a separate detector for direct light

At the first stage for this approach, a pure EAS CL distribution was studied without any detector at a 100–200 m distance from the shower axis. In order to make the results comparable

and reasonable, photons that fell only on a small  $100 \text{ cm}^2$  area were used. This study allowed to evaluate the natural CL fluctuations and their effects on the expected results.

The photons that fell on this small area had an angular distribution – a subsample from the full EAS CL photon distribution at a given distance from the axis. This small distribution has a maximum and a weighted average (or center of mass), both shifted from the shower arrival direction. In Fig. 9, the results for two distances from a 10 PeV proton shower are shown. The samples were taken at 100 and 140 m from the shower axis at a fixed shower orientation. The shift of the CL spot's maximum (red dots, located in the vertices of the calculations grid), its center of mass (green dots) and the true shower arrival direction (black dot) are clearly seen.



**Figure 9.** Direct CL spot angular position (red and green, see text) relative to the true shower arrival direction at 100 m and 140 m distance from the shower axis

The spot position is shifted from the EAS arrival direction. However, the CL spot has an elliptical shape and the shift is along the spot long axis. The shift itself depends on the distance from the shower axis, EAS primary particle energy, type and arrival direction. But, this can be later accounted for. The precision of this method, even without corrections, is already high at around  $0.1\text{--}0.2^\circ$ , but, again, this result was obtained without interference from the background and real detector limitations.

A test with a simple one-lens detector (around 12 cm focal length,  $100 \text{ cm}^2$  aperture, high resolution detector) yielded virtually the same results in an analogous analysis without background.

## 5. EAS Energy and Mass Estimation

A more complex analysis is needed to estimate the EAS primary particle energy and mass, which are the main goals of this project. Some estimations were done, again, on a limited sample set to get a first look on what information there is in EAS CL distributions, so as to have a starting point for comparison during detector optimization.

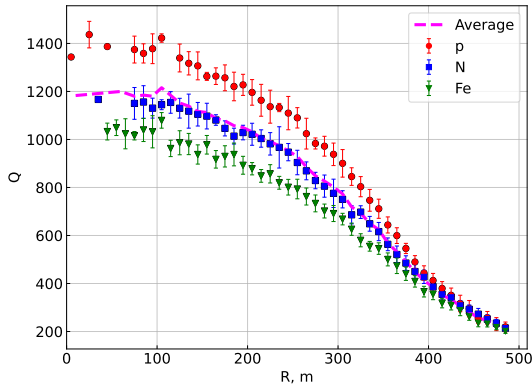


## 5.1. Energy Estimation

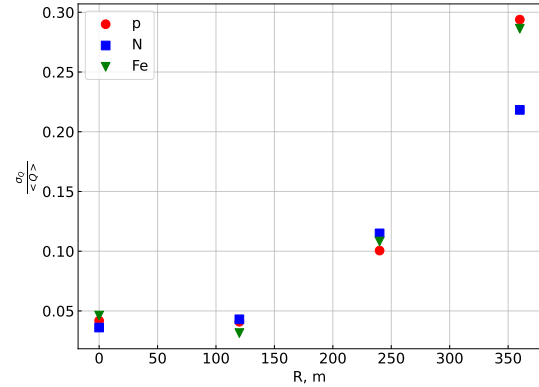
Energy estimation is possible using only reflected CL data. Direct CL registered with a small scale detector without information on axis position relatively to the detector does not allow for such a procedure.

At first, simple approach, the total number of photons  $Q$  reaching the detector SiPM mosaic was used as the main criteria for energy estimation. This number depends on the primary particle energy  $E_0$  and the distance from the center of the detector field of view to the shower axis  $R$ . Such dependencies (i.e.  $Q(E_0; R)$ ) can be obtained as a regression over the precalculated model values for different parameters (energies, angles, atmosphere conditions). Energy estimations are then based on  $Q^{exp}$  and  $R^{exp}$  as:  $E_0^{est} = E_0(Q^{exp}, R^{exp})$ .

An example of a modelled  $Q(E_0; R)$  dependence is shown in Fig. 10 for 1000 m detector flight altitude. Five model EAS with a 10 PeV primary energy for different nuclei with 100 random axis locations per shower (e.g. 500 data points per nuclei) were used. The relative fluctuations  $\sigma Q$  (see Fig. 11), as expected, grow with distance as the total amount of light collected on the detector gets smaller. The resulting uncertainty inevitably will play major role in the overall energy estimation error.



**Figure 10.** Total number of photons collected from a 10 PeV shower depending on the distance from the shower axis to the center of the detector field of view. Red dots represent data from primary protons, blue – nitrogen nuclei, green – iron nuclei

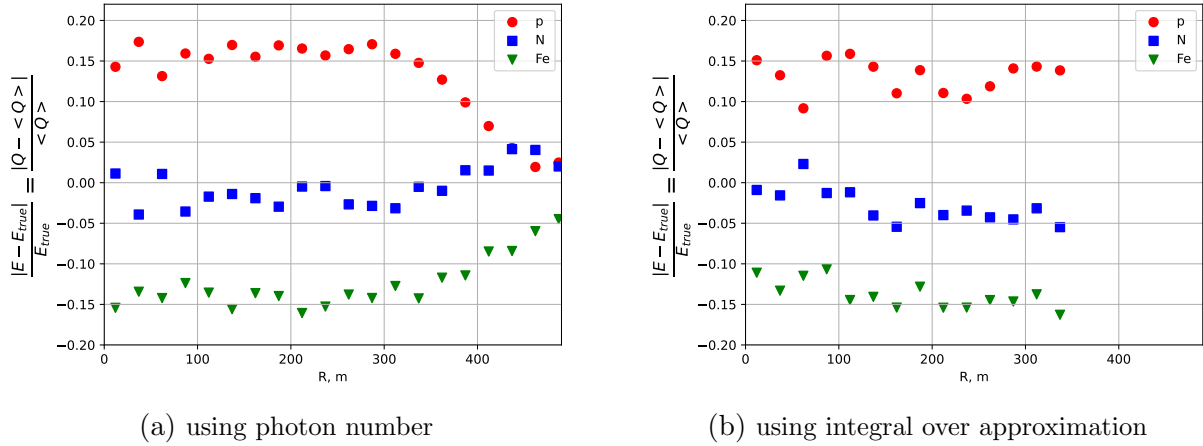


**Figure 11.** The relative fluctuations of the total number of reflected EAS CL photons at several distances for the showers from the left panel. Red dots represent data from primary protons, blue – nitrogen nuclei, green – iron nuclei

Since the primary particle mass is unknown in the experiment prior to a specific analysis, it is common practice to use average EAS characteristics with corrections applied at a later stage. If such an average (shown as a magenta line in Fig. 10) is applied to the same sample set of 10 PeV showers the resulting errors will not exceed 17% (see Fig. 12a).

This simple approach shows good ability to estimated primary mass, however, it can be updated for higher precision. The main issue with simple approach is that after certain distance  $R$  the significant portion of the shower spot is outside the detector field of view. To solve this issue an approximation of the photons LDF can be used to reconstruct their density. Thus, the photons distribution was fitted with a rational function (same as in [17]):

$$I(r) = \frac{p_0^2}{(1 + p_1 r + p_2 r^2 + p_3 r^{1.5})^2 (1 + p_4 r^s)}, \quad (2)$$



**Figure 12.** Errors in energy estimations for different nuclei based on the average  $E(Q; R)$  dependence for two approaches (see text). On the left,  $E(Q; R)$  was obtained using total number of photons in the detector. On the right, same procedure was done using LDF approximation. Red dots represent data from primary protons, blue – nitrogen nuclei, green – iron nuclei. Nitrogen has lower estimation errors since it has the closest to average profile

where  $r$  is the distance from shower axis,  $p_i$  and  $s$  are free parameters. An integral over this smooth function gave a better estimation of total number of photons in the shower and a bit better results. The errors in the primary energy reconstruction became smaller and did not exceed 15% (see Fig. 12b). It should be noted that approximation procedure has its cost and at the moment is not perfectly stable, up to 30% of the LDF fits failed. However, this happened mostly at lower energies (5 PeV showers were also considered) and at high  $R$ , where the number of available data points for approximation is low, what was somewhat expected.

The algorithm will undergo further improvements, namely, we have to optimize the set of limitations imposed on the registered events in order to make procedure more reliable (and reduce the number of lost events), while keeping the energy estimate error low.

## 5.2. Mass Estimation

Contrary to the energy, EAS primary particle mass can be estimated based on both direct and reflected CL.

### 5.2.1. Mass estimation using reflected Cherenkov light

Primary particle mass estimation was done using the same approach as previously used for the SPHERE-2 experiment [17]. In general, the CL lateral distribution function  $I(r)$  correlates with shower longitudinal profile, therefore, there may exist a parameter in the mentioned lateral distribution form description that should be sensitive to the primary particle mass. And such a parameter was found – integral steepness  $\eta$ , i.e. defined by the major part of the distribution and not just a single value at some point:

$$\eta = \frac{\int_0^{r_1} I dr}{\int_{r_1}^{r_2} I dr}, \quad (3)$$

where  $r_i$  are distances from shower axis. These distances can be selected according to the desired property of the shower separation criterion. In our case, the target property was minimal errors in shower separation by primary particle mass.

The steepness  $\eta$  is defined through the CL lateral distribution function  $I$  which traditionally is defined over the observation level. In our studies, we chose to use the CL distribution over the SiPM mosaic, a projection of the original distribution, known at the SiPM positions. Also, the real data will contain background and statistical fluctuations. Thus, the distribution was fitted with a rational function (same as in section above).

So, the mass estimation procedure now consists of shower data fitting and calculations of parameter  $\eta$ . This procedure was done for a set of model EAS from 10 PeV primary protons, nitrogen and iron nuclei with small zenith angles. For each EAS the parameter  $\eta$  was estimated for some set of  $r_i$  parameters. The shower separation procedure over  $\eta$  was done using some varying thresholds. Quality was then evaluated and the  $r_i$  set was updated. In the end the optimal set of  $r_i$  and thresholds was obtained with the following shower separation quality:

**Table 1.** Shower separation quality for p-N and N-Fe pairs

class	p-N	N-Fe
border value	0.699	0.614
class error	0.314	0.317

These results were obtained only for one energy and without background so far, but further investigations are planned for future work.

### 5.2.2. Direct Cherenkov light

The primary particle mass estimation from direct CL data was studied in two separate ways – from clean CL distributions themselves and from separate detector data. Mass estimation using direct CL data from the detector SiPM mosaic is not yet finished.

Direct CL distributions used the same data set as for the arrival direction (see section 4.2.2). Since the detector “sees” EAS from the side the direct CL spot is elongated and rotated in the direction of the shower axis. For our analysis, this angle can be found as:

$$\tan(2\varphi) = \frac{2\sigma_{xy}}{\sigma_{xx} - \sigma_{yy}}, \quad (4)$$

where  $\sigma_{xy}, \sigma_{xx}, \sigma_{yy}$  are second central momenta of the CL distribution ( $x$  and  $y$  are orthogonal coordinates fixed to the detector). The spot’s major and minor axes lengths then can be estimated as:

$$a_1 = \sigma_{xx} \cdot \cos^2(\varphi) + 2\sigma_{xy} \cdot \sin(\varphi) \cdot \cos(\varphi) + \sigma_{yy} \cdot \sin^2(\varphi), \quad (5)$$

$$a_2 = \sigma_{yy} \cdot \cos^2(\varphi) - 2\sigma_{xy} \cdot \sin(\varphi) \cdot \cos(\varphi) + \sigma_{xx} \cdot \sin^2(\varphi), \quad (6)$$

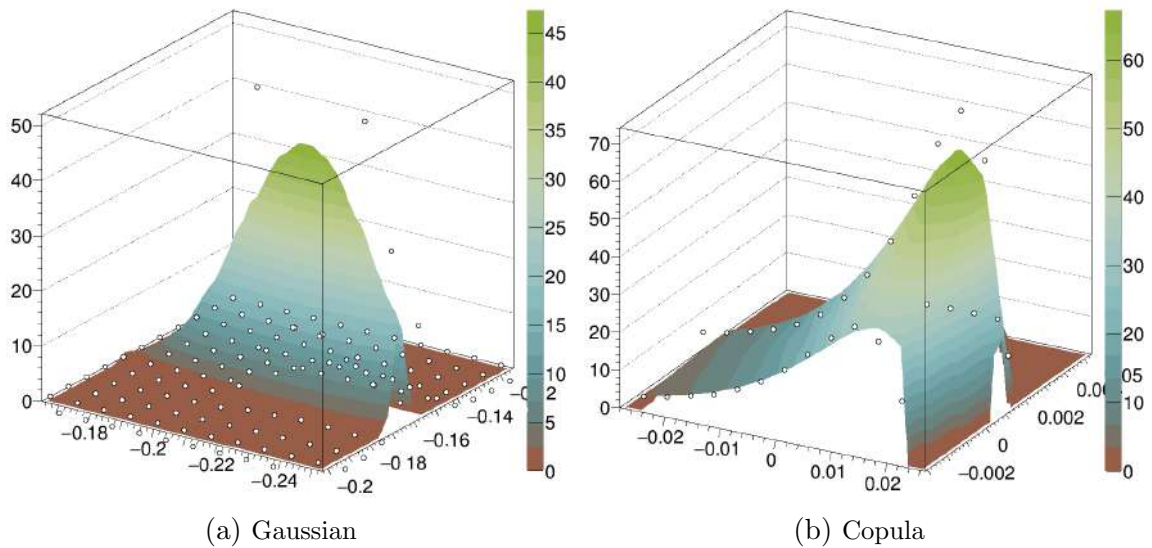
The major axis length is equal to  $a = \max\{a_1, a_2\}$  and can be used as the primary particle mass separation criterion. The precision of  $\varphi$  and  $a$  estimation by this approach is sensitive to the noise level so several thresholds were tested (e.g. rejection of everything below a certain value) both absolute and relative to the maximum of the distribution.

Another option was to fit the spot with some shape function (see Fig. 13a for an example):

$$F(x', y') = p_0 \cdot \exp \left[ -\frac{(x' - p_1)^2}{p_2^2} - \frac{(y' - p_3)^2}{p_4^2} - \frac{2p_5 \cdot (x' - p_1) \cdot (y' - p_3)}{p_2 \cdot p_4} \right], \quad (7)$$

where  $x'$  and  $y'$  are the rotated by  $\varphi$  coordinates,  $p_i$  are free parameters. The spot's long axis can be estimated as  $a = \max\{p_2, p_4\}$ .

Comparison of the quality of separation using estimations of the direct CL spot's long axis by these two approaches are show in Tab. 2. The results of approximation are not very good, since the distribution (7) is symmetrical along its major axes and the direct CL spot is definitely not. So, a new function is now in test – normal distribution over the short axis coupled with a Gamma-distribution along the long axis (see Fig. 13b for an example).



**Figure 13.** Sample fit functions for direct CL lateral distribution function: asymmetrical normal distribution on the left and normal copula function on the right

**Table 2.** Primary particle separation errors ( $\sigma$ ) using direct CL distribution parameters and lens detector data, simulations used bin size  $0.5^\circ \times 0.5^\circ$ , presented values have typical uncertainty around 0.02

approach	CL distribution				lens detector			
	p-N		N-Fe		p-N		N-Fe	
	$\sigma_p$	$\sigma_N$	$\sigma_N$	$\sigma_{Fe}$	$\sigma_p$	$\sigma_N$	$\sigma_N$	$\sigma_{Fe}$
abs. 3 photons	0.25	0.24	0.24	0.26	0.32	0.32	0.32	0.32
abs. 5 photons	0.27	0.26	0.24	0.26	0.27	0.27	0.27	0.27
rel. 5% max	0.35	0.29	0.29	0.29	0.49	0.49	0.47	0.47
rel. 7% max	0.35	0.19	0.29	0.29	0.36	0.35	0.38	0.38
approximation	0.62	0.12	0.23	0.27	–	–	–	–

This analysis was also done for a simple direct CL detector (one lens, large sensor) on the same data sample. While for the arrival direction estimations the image distortions are present,

but can be easily corrected, distortion of the spot shape does affect the mass separation quality. The analysis above applied to the simple detector data showed significantly larger errors (see Tab. 2).

## Conclusion

Here we described an ongoing development process of the SPHERE-3 telescope aimed at energy spectrum and mass composition study of primary cosmic rays in the 1–1000 PeV energy range. In comparison with the previous SPHERE detectors, which registered reflected CL from extensive air showers, the new one will also register direct CL from the same shower providing more valuable data on primary particle parameters. The detector design is still in progress, methods for the detection of direct CL are being tested along with the data analysis approaches.

The updated algorithms for processing reflected light data and the parallelized variant of the CORSIKA code which will allow to model the necessary characteristics of CL are considered.

The considered design of the reflected light telescope makes it possible to estimate the direction and mass of the primary particle no worse than the SPHERE-2 telescope. Its maximum error in estimating the primary energy is evaluated.

It is found that the shower arrival direction of the shower can be defined by the angular distribution of direct light with an error not exceeding  $0.5^\circ$ .

It is proved that the angular distribution of CL is sensitive to the mass of the primary particle and using the length of the spot's long axis as a criterion the primary particles can be divided by mass with classification errors equal or lower than when using CL reflected from the snow.

Also, it was realized that for a better study of the image one should approximate it by a function different from the two-dimensional Gaussian distribution and is asymmetric.

These first results were obtained using a simulations pipeline described here that will be used as the foundation of the general experiment optimization loop. Such optimization is needed due to the unique approach of the SPHERE experiments and, therefore, natural lack of readily available methods and tested solutions. The optimization, however, will be extremely computationally heavy with more and more parameters being set loose in the process. Right now, the only free optimization parameters available are the thresholds for the procedures, however, in the future, it is planned to set detector geometry and other parameters (such as flight altitudes and detector operation regimes) as free parameters in the optimization process, given it will be computationally possible.

## Acknowledgements

The research was carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [21].

This work is supported by the Russian Science Foundation under Grant No. 23-72-00006, <https://rscf.ru/en/project/23-72-00006/>.






*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. Agostinelli, S., Allison, J., Amako, K., *et al.*: Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506(3), 250–303 (2003). [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)
2. Allison, J., Amako, K., Apostolakis, J., *et al.*: Geant4 developments and applications. *IEEE Transactions on Nuclear Science* 53(1), 270–278 (2006). <https://doi.org/10.1109/TNS.2006.869826>
3. Allison, J., Amako, K., Apostolakis, J., *et al.*: Recent developments in Geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835, 186–225 (2016). <https://doi.org/10.1016/j.nima.2016.06.125>
4. Antonov, R.A., Aulova, T.V., Bonvech, E.A., *et al.*: Detection of reflected Cherenkov light from extensive air showers in the SPHERE experiment as a method of studying super-high energy cosmic rays. *Phys. Part. Nucl.* 46, 60–93 (2015). <https://doi.org/10.1134/S1063779615010025>
5. Antonov, R.A., Beschapov, S.P., Bonvech, E.A., *et al.*: Results on the primary CR spectrum and composition reconstructed with the SPHERE-2 detector. *Journal of Physics: Conference Series* 409 (feb 2013). <https://doi.org/10.1088/1742-6596/409/1/012088>
6. Bakhromzod, R., Galkin, V.: The search and analysis of optimal criteria for the selection of extensive air showers from  $\gamma$ -quanta by Cherenkov telescopes. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1018 (2021). <https://doi.org/10.1016/j.nima.2021.165842>
7. Bonvech, E.A., Azra, C., Chernov, D.V., *et al.*: Design of the Simulation Scheme for SPHERE-3 Telescope for the  $10^{15}$ – $10^{18}$  eV Primary Cosmic Ray Studies Using Direct and Reflected Cherenkov Light from the Extensive Air Showers. *Phys. Atom. Nucl.* 86(6), 1048–1055 (2023). <https://doi.org/10.1134/S1063778824010149>
8. Budnev, N., Chernov, D., Gress, O., *et al.*: Tunka-25 Air Shower Cherenkov array: The main results. *Astropart. Phys.* 50-52, 18–25 (2013). <https://doi.org/10.1016/j.astropartphys.2013.09.006>
9. Chernov, D.V., Azra, C., Bonvech, E.A., *et al.*: SPHERE-3 Project for Studying the Composition of Primary Cosmic Rays in the Energy Range Between 1 and 1000 PeV. *Phys. Atom. Nucl.* 85(6), 641–652 (2022). <https://doi.org/10.1134/S1063778822060059>
10. Chudakov, A.: A possible method to detect EAS by the Cherenkov radiation reflected from the snowy ground surface. (in russian). In: *Experimental methods of studying cosmic rays of superhigh energies: Proc. All-Union Symposium*. vol. 620, pp. 69–72 (1972)
11. Galkin, V.I., Borisov, A.S., Bakhromzod, R., *et al.*: A method for estimation of the parameters of the primary particle of an extensive air shower by a high-altitude detector. *Moscow University Physics Bulletin* 73(2), 179–186 (2018). <https://doi.org/10.3103/S0027134918020078>

12. Galkin, V.I., Gzhatdov, T.A.: Classifying groups of PCR nuclei with energies of  $10^{15}$ - $10^{16}$  eV according to the spatial-angular distribution of EAS Cherenkov light. Bulletin of the Russian Academy of Sciences: Physics 75(3), 309–312 (Mar 2011). <https://doi.org/10.3103/S1062873811030166>
13. Galkin, V., Borisov, A., Bakhromzod, R., *et al.*: EAS primary particle parameter estimation with the complex Pamir-XXI detector array. EPJ Web Conf. 145 (2017). <https://doi.org/10.1051/epjconf/201614515004>
14. Heck, D., Knapp, J., Capdevielle, J.N., *et al.*: CORSIKA: A Monte Carlo code to simulate extensive air showers. Report FZKA-6019 (2 1998). <https://doi.org/10.5445/IR/270043064>
15. Kalmykov, N., Ostapchenko, S., Pavlov, A.: Quark-gluon-string model and EAS simulation problems at ultra-high energies. Nuclear Physics B - Proceedings Supplements 52(3), 17–28 (1997). [https://doi.org/10.1016/S0920-5632\(96\)00846-8](https://doi.org/10.1016/S0920-5632(96)00846-8)
16. Knurenko, S., Petrov, I.: Mass composition of cosmic rays above 0.1 EeV by the Yakutsk array data. Advances in Space Research 64(12), 2570–2577 (2019). <https://doi.org/10.1016/j.asr.2019.07.019>
17. Latypova, V.S., Nemchenko, V.A., Azra, C.G., *et al.*: Method for Separating Extensive Air Showers by Primary Mass Using Machine Learning for a SPHERE-Type Cherenkov Telescope. Moscow University Physics Bulletin 78(1), S25–S31 (Dec 2023). <https://doi.org/10.3103/S0027134923070196>
18. Ostapchenko, S.: LHC data on inelastic diffraction and uncertainties in the predictions for longitudinal extensive air shower development. Phys. Rev. D 89 (Apr 2014). <https://doi.org/10.1103/PhysRevD.89.074009>
19. Podgrudkov, D.A., Bonvech, E.A., Vaiman, I.V., *et al.*: First results from operating a prototype wide-angle telescope for the TAIGA installation. Bulletin of the Russian Academy of Sciences: Physics 85(4), 408–411 (Apr 2021). <https://doi.org/10.3103/S1062873821040286>
20. Poole, C.M., Cornelius, I., Trapp, J.V., Langton, C.M.: A CAD interface for GEANT4. Australasian Physical & Engineering Sciences in Medicine 35(3), 329–334 (Sep 2012). <https://doi.org/10.1007/s13246-012-0159-8>
21. Voevodin, V.V., Antonov, A.S., Nikitenko, D.A., *et al.*: Supercomputer Lomonosov-2: Large scale, deep monitoring and fine analytics for the user community. Supercomputing Frontiers and Innovations 6(2), 4–11 (Jun 2019). <https://doi.org/10.14529/jsfi190201>

# Quantum-Chemical Study of Some Trispyrazolobenzenes and Trispyrazolo-1,3,5-triazines

Vadim M. Volokhov<sup>1</sup> , Vladimir V. Parakhin<sup>2</sup> , Elena S. Amosova<sup>1</sup> , David B. Lempert<sup>1</sup> , Vladimir V. Voevodin<sup>3</sup> 

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

Development of new high-energy density materials and study of their properties is an important task, since such materials are in high demand in various application areas. This paper continues the study of polynitrogen fused tetracyclic systems which include threeazole rings annelated with a benzene or azine ring. Such polycyclic structures attract special attention of scientists. This paper is dedicated to the study of properties of a number of promising high-energy tetracyclic compounds annelated with pyrazole nitro derivatives. For this study, we used quantum-chemical methods (the hybrid density functional B3LYP and the composite G4MP2 and G4 methods) within the Gaussian 09 and NWChem software packages at Lomonosov Moscow State University Supercomputer Complex. We used the atomization method and method of reactions to calculate the enthalpy of formation. We analyzed the dependence of the enthalpy of formation on the structural parameters of the compounds and calculated the optimized structures and IR absorption spectra. We also compare the Gaussian 09 and NWChem quantum chemical programs in terms of efficiency, parallelization and computational requirements. In the cases where the G4-level accuracy of the results is not required, the use of NWChem can significantly save computation time.

*Keywords:* quantum-chemical calculations, high-energy density materials, tris(azolo)benzenes, tris(azolo)azines, enthalpy of formation, high-performance computing.

## Introduction

High-energy polynitrogen heterocyclic compounds have become a focus of study of scientists around the world over the last decades. The key characteristic of such compounds is the enthalpy of their formation, because it is on this value that the energetic possibilities of the compounds mainly depend. Research chemists, before synthesizing new compounds, need to preliminarily assess their energetic possibilities, so as not to waste time and resources on objects that are not promising enough. In order to do so, it is necessary to know the enthalpy of formation, the best way to determine which is quantum-chemical calculations, the development of methods of which has been very rapidly progressing in recent decades. Even in the case of the already synthesized compounds, it is very important to estimate the enthalpy of formation by quantum-chemical calculations, since experimental measurements are not always reliable due to the insufficient purity of the experimental samples.

Our scientific group makes systematic studies of the energetic possibilities of polynitrogen structures consisting of three or four fused heterocycles [1–5]. In our previous works, we studied the properties of tris(pyrrolo)benzenes and -[1,3,5]triazines [4] and triimidazolobenzenes and -[1,3,5]triazines [5]. In this work, the objects of study are tris(pyrazolo)benzenes and [1,3,5]triazines, which consist of three pyrazole rings annelated with a central benzene or 1,3,5-triazine ring. We selected unsubstituted tetracycles **1a,b** and their hexanitro derivatives **2a,b,c** (Fig. 1) for quantum chemical calculations in order to assess the way the structural factors affect the

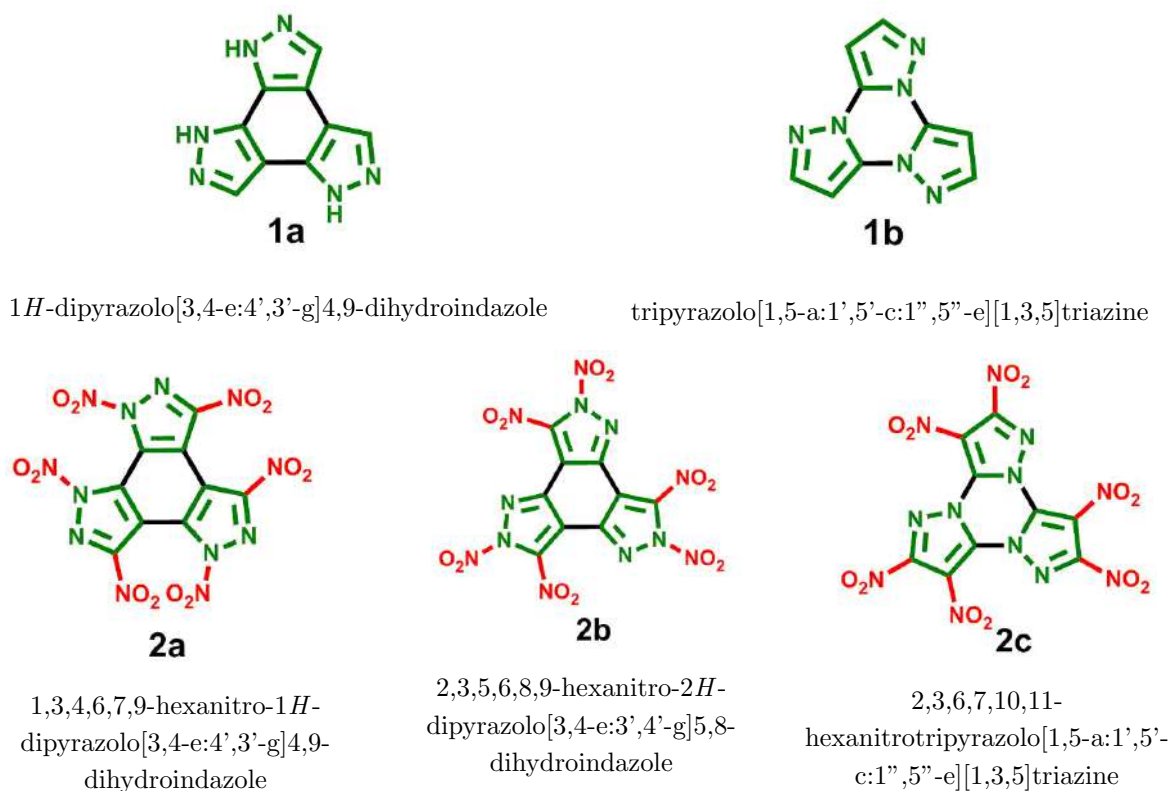
<sup>1</sup>Federal Research Center of Problems of Chemical Physics and Medicinal Chemistry of the Russian Academy of Sciences, Chernogolovka, Moscow Region, Russian Federation

<sup>2</sup>N.D. Zelinskiy Institute of Organic Chemistry of the Russian Academy of Sciences, Moscow, Russian Federation

<sup>3</sup>Research Computing Center of Lomonosov Moscow State University, Moscow, Russian Federation



enthalpy of formation in the gas phase at a temperature of 298 K and a pressure  $p = 1$  atm ( $\Delta H_{f(g)}^{298}$ ) of this series of compounds.



**Figure 1.** Objects of study – tetracyclic compounds **1a,b** and **2a,b,c**

The study of the physicochemical parameters of high-molecular compounds requires large-scale massive parallel calculations using specialized supercomputer resources and specialized software like Gaussian, NWChem, etc. Parallel implementations of such calculations based on MPI and Linda for molecules containing two to three dozen heavy atoms require up to several months of computing time on dozens of Gold class cores. In this work, we used the equipment of Lomonosov Moscow State University Supercomputer Complex for calculations.

The article is organized as follows. Section 1 is devoted to the methods used in this study. In Section 2 we discuss the results of our work. Section 3 contains computational details. Conclusion summarizes the study and points directions for further work.

## 1. Computation Method

Quantum-chemical program packages Gaussian 09 [6] and NWChem [7] were used for calculations. The geometry of the molecules under study was obtained by fully optimizing all geometric parameters using the hybrid density functional B3LYP [8, 9] with 6-311+G(2d,p) basis. Stability of the resulting configurations has been confirmed by subsequent calculation of vibrational frequencies using analytical first and second derivatives without taking into account the correction for anharmonicity (absence of imaginary frequencies). The enthalpy of formation in the gaseous phase of the studied substances was calculated using the composite G4MP2 method [10, 11] for all structures under study and using the composite G4 method [11] for structures **1a,b**. For NWChem calculations, a module was written that reproduces the sequence

of calculations of the composite G4MP2 method as given in [10, 12] and using the following formula to obtain the final energy of the molecule:

$$E_0[G4(MP2)] = CCSD(FC, T)/6 - 31G(d) + \Delta E_{MP2} + \Delta E_{HF} + \Delta E(SO) + E(HLC) + E(ZPE),$$

where  $CCSD(FC, T)/6 - 31G(d)$  is the energy calculation is at the triples-augmented coupled cluster level of theory, CCSD(T), with the 6-31G(d) basis set, using frozen core;  $\Delta E_{MP2}$  and  $\Delta E_{HF}$  are the energy corrections calculated by the MP2 and HF methods accordingly,  $\Delta E(SO)$  is the spin-orbit correction,  $E(HLC)$  is the higher-level correction, and  $E(ZPE)$  is the zero-point energy.

The IR absorption spectra were calculated using the hybrid density functional B3LYP with the 6-311+G(2d,p) basis and introducing a scaling factor of 0.967 [13].

To calculate the enthalpy of formation of the substances in the gas phase, we used two methods: 1) one based on the atomization reaction and 2) one based on reactions.

We used the method based on the atomization reaction in our previous works and for the  $C_wH_xN_yO_z$  molecule it consists of the following steps:

1. The atomization energy is calculated

$$\sum D_0 = wE_0(C) + xE_0(H) + yE_0(N) + zE_0(O) - E_0(C_wH_xN_yO_z),$$

where  $E_0(C), E_0(H), E_0(N), E_0(O), E_0(C_wH_xN_yO_z)$  are computed total energies of atoms and molecule.

2. The enthalpy of formation at 0K is calculated

$$\Delta H_f^\circ(C_wH_xN_yO_z, 0K) = w\Delta H_f^\circ(C, 0K) + x\Delta H_f^\circ(H, 0K) + y\Delta H_f^\circ(N, 0K) + z\Delta H_f^\circ(O, 0K) - \sum D_0,$$

where the first four summands are the enthalpies of formation of gaseous atomic components from the NIST-JANAF database of thermochemical parameters [14].

3. The enthalpy of formation at 298.15K is calculated

$$\begin{aligned} \Delta H_f^\circ(C_wH_xN_yO_z, 298K) = & \Delta H_f^\circ(C_wH_xN_yO_z, 0K) + \\ & + (H^0(C_wH_xN_yO_z, 298K) - H^0(C_wH_xN_yO_z, 0K)) - \\ & - w(H^0(C, 298K) - H^0(C, 0K)) - \\ & - x(H^0(H, 298K) - H^0(H, 0K)) - \\ & - y(H^0(N, 298K) - H^0(N, 0K)) - \\ & - z(H^0(O, 298K) - H^0(O, 0K)), \end{aligned}$$

where the second summand is obtained from the molecule computation, the third to sixth summands are known from experiment (or calculated from experimental molecular constants).

According to Hess's law, the enthalpy of a reaction does not depend on the specific path of its occurrence, therefore any thermodynamic cycle that links the reactants with products in stable standard states can be used for calculations. In this case, to calculate the enthalpy of formation of the compound under study, it is necessary to obtain the values of the electron energy of all

members of the cycle, which are calculated using the lower level theory, as well as the values of the enthalpy of formation of the reactants, which can be calculated using the atomization method. Compared to direct calculation of the enthalpy of formation of the molecule under study by the atomization method, this approach allows for a significant reduction in the overall computational complexity. The reaction schemes used to calculate the enthalpy of formation are presented in Fig. 2.

## 2. Results and Discussion

### 2.1. Enthalpy of Formation

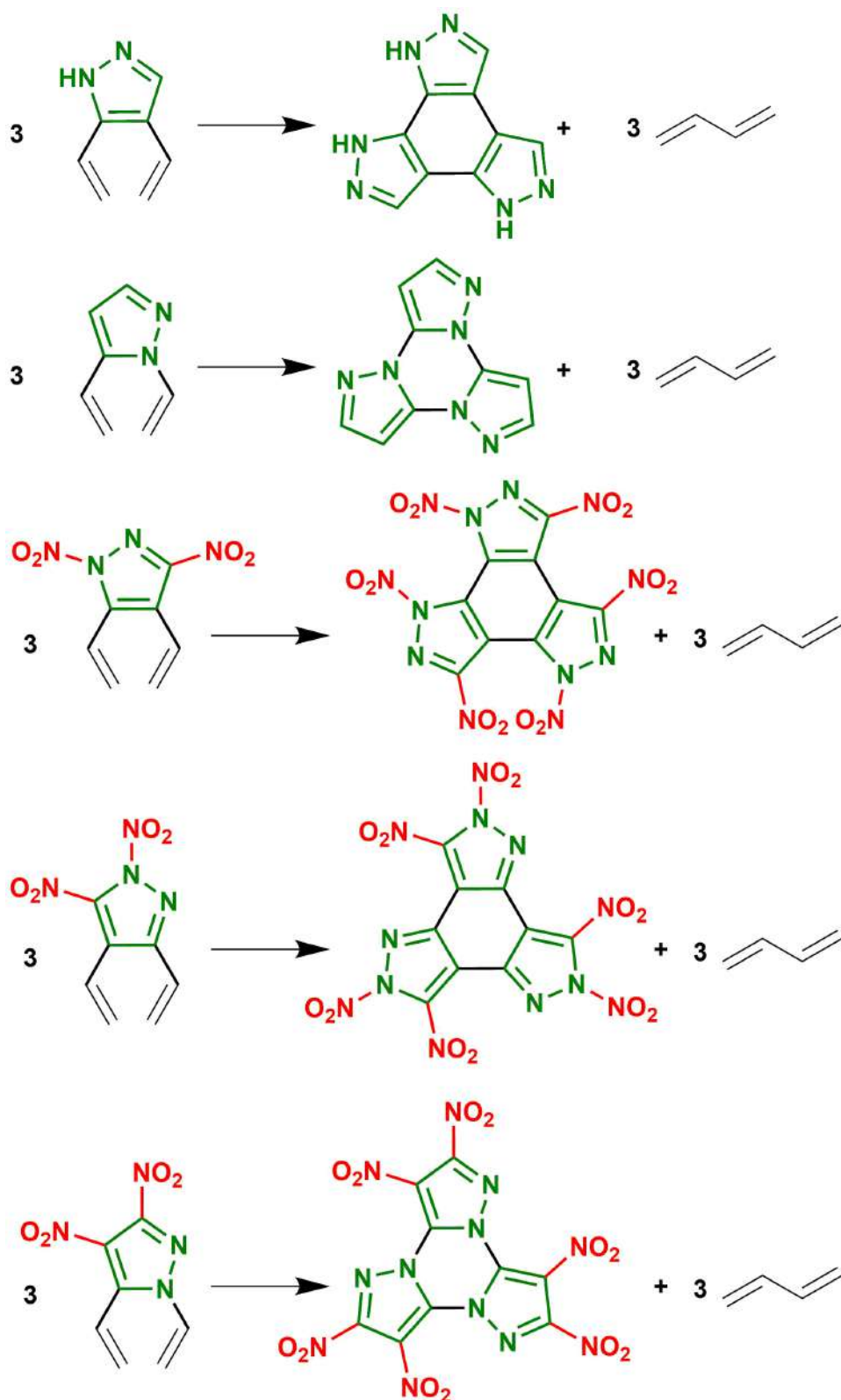
Geometric parameters of the optimized structure are pictured in Fig. 3. The results of calculation of the enthalpy of formation by various methods are gathered in Tab. 1.

**Table 1.** Results of the calculation of the enthalpy of formation of the studied tetracycles **C1–5**

№	Formula, {molecular mass [g/mol]}	$\Delta H_{f(g)}^{298}$ [kJ/mol (kJ/kg)]				
		B3LYP/6- 311+G(2d,p)	G4MP2	G4	NWChem G4MP2	Reactions G4MP2
<b>1a</b>	$C_9H_6N_6$ {198.065}	673.96 <b>(3402.70)</b>	567.38 <b>(2862.83)</b>	562.28 <b>(2837.08)</b>	568.94 <b>(2872.49)</b>	546.14 <b>(2757.37)</b>
<b>1b</b>		775.00 <b>(3912.85)</b>	664.44 <b>(3352.59)</b>	658.30 <b>(3321.59)</b>	665.47 <b>(3359.83)</b>	642.16 <b>(3242.17)</b>
<b>2a</b>	$C_9N_{12}O_{12}$ {467.976}	1138.19 <b>(2432.15)</b>	936.50 <b>(2000.33)</b>	–	941.90 <b>(2012.71)</b>	935.38 <b>(1998.77)</b>
<b>2b</b>		1186.21 <b>(2534.76)</b>	1002.27 <b>(2140.82)</b>	–	1007.47 <b>(2152.81)</b>	986.92 <b>(2108.92)</b>
<b>2c</b>		1087.87 <b>(2324.64)</b>	894.87 <b>(1911.42)</b>	–	898.81 <b>(1920.63)</b>	886.84 <b>(1895.06)</b>

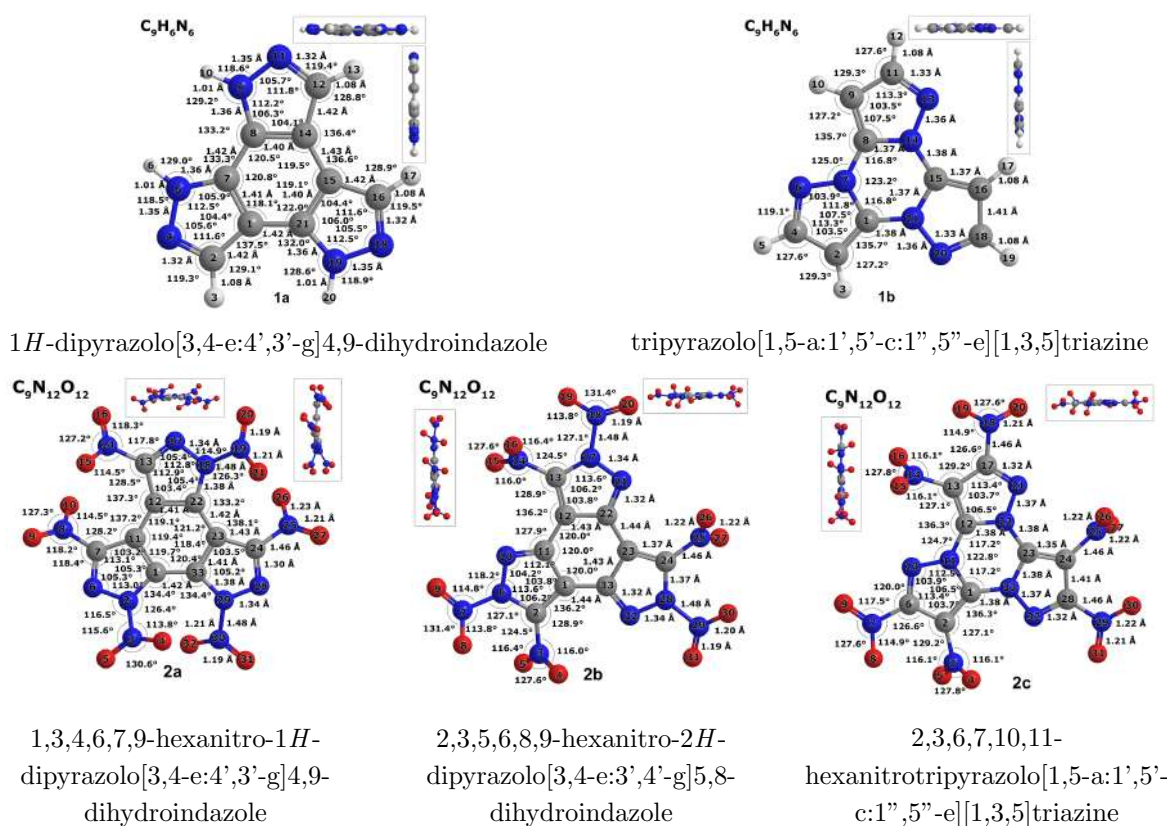
Values of  $\Delta H_{f(g)}^{298}$  obtained using density functional theory proved to be overestimated (they exceed those obtained by G4MP2 method by 107–110 kJ/mol for compounds **1a,b** and by 183–202 kJ/mol for compounds **2a,b,c**). Still, they follow the general trend of the enthalpy of formation behavior and can be used for an initial comparison of the characteristics of compounds. Values of  $\Delta H_{f(g)}^{298}$  obtained using the Gaussian G4MP2 and G4 methods (for compounds **1a,b**) and using NWChem are close (within 7 kJ/mol). The slight difference in the values obtained using the G4MP2 method in Gaussian and NWChem might be due to the fact that NWChem does not include the G4MP2 method and only follows the steps of the original G4MP2 method from Gaussian as it had been indicated in Introduction. Therefore, deviations may occur when calculating all the elements of the total energy. When using the G4MP2 method, the values of  $\Delta H_{f(g)}^{298}$  obtained through the reactions are lower than those obtained through the atomization reaction by 1–15 kJ/mol for compounds **2a,b,c** and by 21–22 kJ/mol for compounds **1a,b**.

Analysis of the results of quantum chemical calculations of  $\Delta H_{f(g)}^{298}$  of the structures under study shows that among unsubstituted tetracycles the  $\Delta H_{f(g)}^{298}$  value of compound **1b** based on the 1,3,5-triazine ring is  $\sim 450$  kJ/kg higher than that of its isomer **1a** with a benzene ring in the center of the molecule, which is quite natural, since **1b** contains a larger number of C – N bonds



**Figure 2.** The reaction schemes used to calculate the enthalpy of formation

in its structure than its isomeric benzotriazole **1a**. At the same time, in the case of hexanitro derivatives of tetracycles **2a,b,c**, it is, on the contrary, isomer **2c** based on the 1,3,5-triazine ring that has the lowest  $\Delta H_{f(g)}^{298}$  as compared to compounds **2a** and **2b** with a central benzene ring (the difference is  $\sim 40$ – $110$  kJ/mol or  $\sim 90$ – $230$  kJ/kg). This can be explained by the fact



**Figure 3.** Structures and geometric parameters (in Å and °) of the studied tetracycles **1a,b** and **2a,b,c** (calculation level: B3LYP/6-311+G(2d,p))

that in addition to  $C - NO_2$ , there are also more endothermic  $N - NO_2$  bonds in the structure of isomers **2a,b**, which are absent in **2c**. Meanwhile  $\Delta H_{f(g)}^{298}$  of tetracycle **2a** is lower than that of **2b**, probably due to the position of nitro groups in each of the pyrazole rings. In the case of **2b**, the nitro groups are located in positions 1 and 2 in relation to each other, whereas in the case of isomer **2a** they are in position 1 and 3, while it is known that closer arrangement of nitro groups in isomers increases  $\Delta H_{f(g)}^{298}$  [15].

## 2.2. IR Spectra and Frequency Analysis

The IR absorption spectra are shown in Fig. 4. Compounds **1a,b** containing hydrogen are characterized by the intense absorption bands associated with vibrations of hydrogen bonds. Thus the most intense peak in the region of  $\sim 3553 \text{ cm}^{-1}$  of the spectrum of compound **1a** corresponds to stretching vibrations of  $N - H$  bonds, and the peak in the region of  $\sim 398\text{--}358 \text{ cm}^{-1}$  corresponds to out-of-plane bending vibrations of the same bonds. The intense absorption bands in the region of  $\sim 911 \text{ cm}^{-1}$  are associated with bending vibrations in the pyrazole rings, and in region of  $\sim 1604 \text{ cm}^{-1}$  – with stretching vibrations of  $C = C$  bonds in the benzene ring. The intense peak in the region of  $\sim 1591 \text{ cm}^{-1}$  of the **1b** spectrum can be attributed to the stretching vibrations of the  $C = C$  bonds in the pyrazole rings, and the peaks in the region of  $\sim 1444\text{--}1297 \text{ cm}^{-1}$  to the bending vibrations of  $C - H$  bonds. Compounds **2a,b,c** are characterized by intense absorption bands associated with vibrations in nitro groups. The intense absorption bands in the region of  $\sim 1667\text{--}1555 \text{ cm}^{-1}$  and  $\sim 1263\text{--}1255 \text{ cm}^{-1}$  of the **2a** and **2b**

spectra can be attributed, respectively, to asymmetric and symmetric stretching vibrations of  $N - O$  bonds of nitro groups, and peaks in the region of  $\sim 827 - 776 \text{ cm}^{-1}$  to bending vibrations of the same bonds. Peaks in the region of  $\sim 1340 - 1328 \text{ cm}^{-1}$  of the **2a,b,c** spectra are associated with the stretching vibrations of  $C - N$  bonds between the pyrazole rings and nitro groups. Peaks in the region of  $\sim 1568 - 1550 \text{ cm}^{-1}$  of the **2c** spectrum can be attributed to the stretching vibrations of  $N - O$  bonds of nitro groups, and peaks in the region of  $\sim 840 - 797 \text{ cm}^{-1}$  with the bending vibrations in nitro groups. Peak in the region of  $\sim 1605 \text{ cm}^{-1}$  corresponds to stretching vibrations of  $C = C$  bonds in the pyrazole rings.

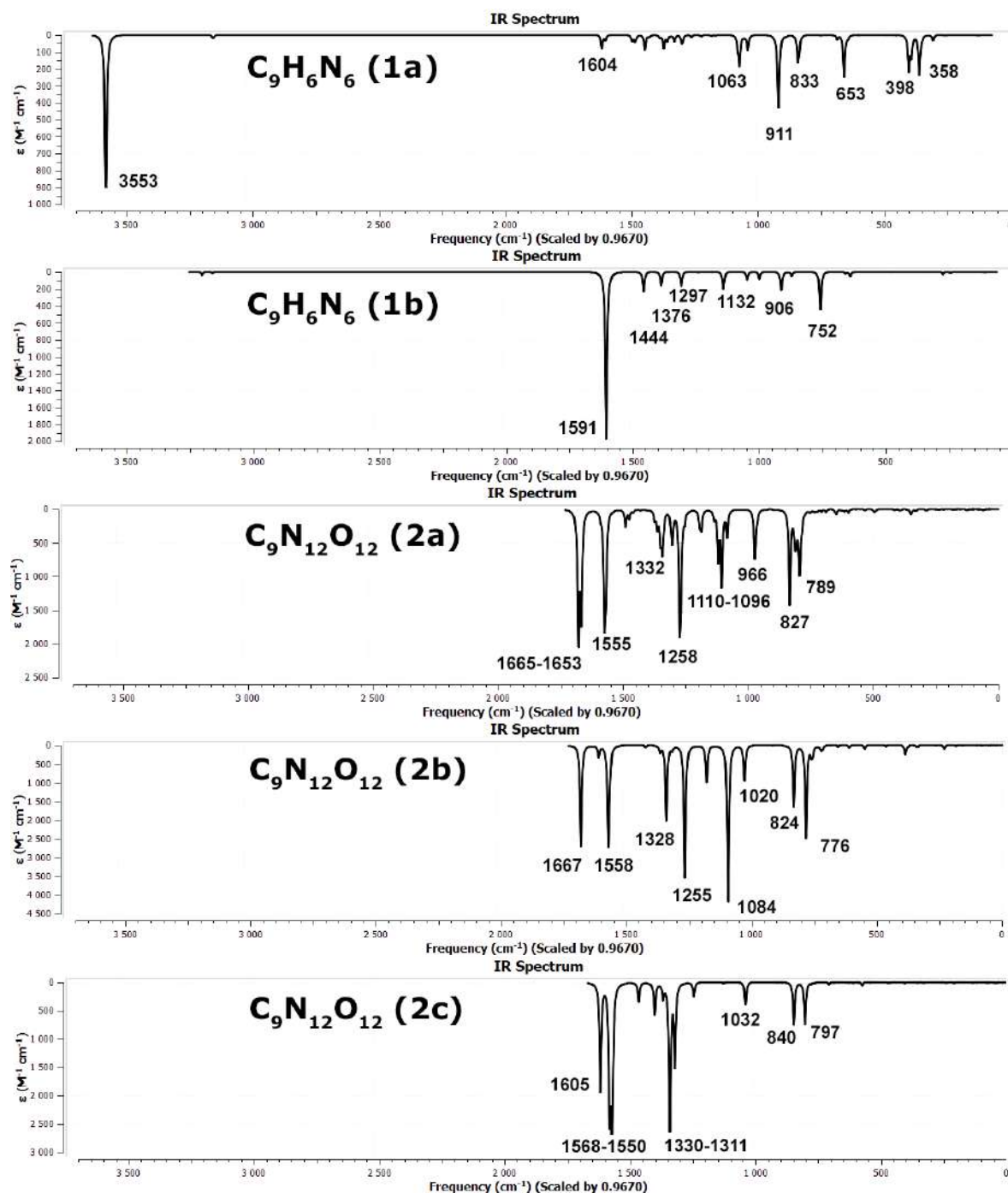


Figure 4. IR absorption spectra of the studied tetracycles **1a,b** and **2a,b,c** (calculation level: B3LYP/6-311+G(2d,p))

### 3. Computational Details

Quantum-chemical computations were carried out at Lomonosov Moscow State University Supercomputer Complex [16–18] (project 2312) and at the Federal Research Center of Problems of Chemical Physics and Medicinal Chemistry of the Russian Academy of Sciences. Computations by the G4 and G4MP2 method using the Gaussian software package and by the G4MP2 method using the NWChem software package were carried out on the *volta2* section of the Lomonosov-2 supercomputer, equipped with Intel Xeon Gold 6240 processors (18 cores, 2.60 GHz, 1497.6 GFlop/s) and Nvidia Tesla V100 graphics accelerators (900-2G500-0010-000, 1246 MHz, 7 TFlop/s).

Gaussian uses its own Linda software for parallelization, but the distribution of tasks between processors of one node is not very efficient. G4MP2 calculations for compounds **1a,b** took about 7 hours, and G4 calculations took about 17 hours.

NWChem allows tasks to be distributed between several nodes using MPI, leading to a significant reduction in the time required to complete calculations. NWChem computations for compounds **1a,b** used two nodes and took about an hour, which is significantly faster than in the case of Gaussian. NWChem computations for molecules **2a,b,c** with large number of heavy atoms required four nodes for each task and took about 30–35 hours depending on the molecule.

Due to time limitations in the *volta2* section and particular qualities of the G4MP2 method within Gaussian program package computations using this method for compounds **2a,b,c** were carried out on a separate computing resource equipped with an Intel(R) Xeon(R) Gold 6140 CPU (2.30 GHz), 259 Gb RAM, and 20 TB of disk space, and took about 40 days. In such circumstances, it was impossible to perform the computation for **2a,b,c** substances using the most accurate G4 method (the computation time was estimated at several months approximately).

## Conclusions

We computed the optimized geometry, IR absorption spectra, and enthalpy of formation in the gaseous phase of a series of tetracycles with a central benzene or [1,3,5]triazine ring annelated with three pyrazole rings. We analyzed the dependence between the enthalpy of formation and the structural parameters of the molecule (number and types of bonds, presence of functional groups). Gaussian and NWChem program packages were compared in terms of accuracy and time costs. In the cases, where the G4-level accuracy of the results is not required (the deviation from experimental data reaches 4 kJ/mol), the use of NWChem can significantly save computation time owing to the efficient distribution of tasks between nodes while maintaining an acceptable accuracy of the results. Within the Gaussian software package, time savings can be achieved using reactions in calculations, but the accuracy of the results, in this case, is reduced compared to NWChem. The obtained data on the enthalpy of formation becomes the basis for predicting the energetic possibilities of the considered compounds in high-energy compositions.

## Acknowledgements

V.M. Volokhov and E.S. Amosova performed the research in accordance with the state task, state registration No. 124013100856-9. V.V. Parakhin was engaged in the formulation of a scientific problem, literature review, analysis of the results, writing and editing of the article. D.B. Lempert performed the work in accordance with the state task, state registration

No. 124020100045-5. V.V. Voevodin participated in the quantum-chemical research and the analysis of the results. Calculations on the resources of the supercomputer complex of Lomonosov Moscow State University were supported by the Russian Science Foundation (project No. 23-71-00005).

*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. Volokhov, V.M., Amosova, E.S., Volokhov, A.V., *et al.*: Quantum-chemical calculations of physicochemical properties of high enthalpy 1,2,3,4- and 1,2,4,5-tetrazines annelated with polynitroderivatives of pyrrole and pyrazole. Comparison of different calculation methods. *Comput. Theor. Chem.* 1209, 113608 (2022). <https://doi.org/10.1016/j.comptc.2022.113608>
2. Volokhov, V.M., Parakhin, V.V., Amosova, E.S., *et al.*: Quantum-Chemical Study of Gas-Phase 5/6/5 Tricyclic Tetrazine Derivatives. *Supercomputing Frontiers and Innovations* 10(3), 61–72 (2023). <https://doi.org/10.14529/jsfi230306>
3. Volokhov, V.M., Parakhin, V.V., Amosova, E.S., *et al.*: Quantum-chemical calculations of the enthalpy of formation of 5/6/5 tricyclic tetrazine derivatives annelated with nitrotriazoles. *Russ. J. Phys. Chem. B* 18(1), 28–36 (2024). <https://doi.org/10.1134/S1990793124010196>
4. Volokhov, V.M., Amosova, E.S., Parakhin, V.V., *et al.*: Quantum-Chemical Study of Some Tris(pyrrolo)benzenes and Tris(pyrrolo)-1,3,5-triazines. In: Voevodin, V., Sobolev, S., Yakobovskiy, M., Shagaliev, R. (eds) *Supercomputing. RuSCDays 2023. Lecture Notes in Computer Science*, vol. 14388, pp. 177–189. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-49432-1\\_14](https://doi.org/10.1007/978-3-031-49432-1_14)
5. Volokhov, V.M., Amosova, E.S., Parakhin, V.V., *et al.*: Quantum-Chemical Simulation of Some Triimidazolobenzenes and Triimidazolo-1,3,5-Triazines Parallel Computational Technologies. *PCT 2024. Communications in Computer and Information Science* (in print)
6. Frisch, M.J., Trucks, G.W., Schlegel, H.B., *et al.*: *Gaussian 09, Revision B.01*. Gaussian, Inc., Wallingford CT (2010).
7. Valiev, M., Bylaska, E. J., Govind, N., *et al.*: NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Commun.* 181, 1477 (2010). <https://doi.org/10.1016/j.cpc.2010.04.018>
8. Becke, A.D.: Densityfunctional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* 98(4), 5648–5652 (1993). <https://doi.org/10.1063/1.464913>
9. Johnson, B.J., Gill, P.M.W., Pople, J.A.: The performance of a family of density functional methods. *J. Chem. Phys.* 98(4), 5612–5626 (1993). <https://doi.org/10.1063/1.464906>



10. Curtiss, L.A., Redfern, P.C., Raghavachari, K.: Gaussian-4 theory using reduced order perturbation theory. *J. Chem. Phys.* 127, 124105 (2007). <https://doi.org/10.1063/1.2770701>
11. Curtiss, L.A., Redfern, P.C., Raghavachari, K.: Gn theory. *Comput. Mol. Sci.* 1, 810–825 (2011). <https://doi.org/10.1002/wcms.59>
12. Curtiss, L.A., Redfern, P.C., Raghavachari, K.: Gaussian-4 theory. *J. Chem. Phys.* 126, 084108 (2007). <https://doi.org/10.1063/1.2436888>
13. CCCBDB Vibrational Frequency Scaling Factors. <https://cccbdb.nist.gov/vsfx.asp>, accessed: 2024-03-10
14. NIST-JANAF Thermochemical Tables <https://janaf.nist.gov/>, accessed: 2024-03-10
15. Kizin, A.N., Dvorkin, P.A., Ryzhova, G.L, Lebedev, Yu.A.: Parameters for calculation of standard enthalpies of formation of organic compounds in the liquid state. *Russ Chem Bull* 35, 343–346 (1986). <https://doi.org/10.1007/BF00952920>
16. Voevodin, V.V., Antonov, A.S., Nikitenko, D.A., *et al.*: Supercomputer Lomonosov-2: large scale, deep monitoring and fine analytics for the user community. *Supercomput. Front. Innov.* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
17. Voevodin, V., Zhumatiy, S., Sobolev, S., *et al.*: Practice of Lomonosov supercomputer. *Otkrytye sistemy [Open Syst.]* 7, 36–39 (2012) (in Russian).
18. Nikitenko, D., Voevodin, V., Zhumatiy, S.: Deep analysis of job state statistics on Lomonosov-2 supercomputer. *Supercomput. Front. Innov.* 5(2), 4–10 (2019). <https://doi.org/10.14529/jsfi180201>

# Wing Noise Simulation of Supersonic Business Jet in Landing Configuration

Alexey P. Duben<sup>1</sup> , Tatiana K. Kozubskaya<sup>1</sup> , Pavel V. Rodionov<sup>1</sup> 

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

The paper presents the results of wing noise simulations for the prototype of supersonic business jet in landing mode. The near-field airflow is modeled according to Delayed Detached Eddy Simulation approach. The finite-volume vertex-centered scheme with the low weight of upwind component is used for convective flux approximation. The noise at the far-field points is calculated by the Ffowcs Williams–Hawkings method. The noise spectra at the near-field points are presented, and the impact of local mesh resolution and numerical instability on the near-field acoustics is discussed. For the Ffowcs Williams–Hawkings method due to the features of the wing geometry and the resulting flow configuration, we used non-standard integration surfaces to reduce computational costs of the scale-resolving simulations. Additionally, we employed optimized mesh resolution on the integration surfaces to significantly reduce the disk space required for storing the data for far-field noise calculations. The tests performed for the near-field and far-field points demonstrated applicability of the proposed optimizations.

*Keywords:* computational fluid dynamics, aeroacoustics, airframe noise, turbulent flow, detached eddy simulation, mixed-element mesh, FWH method.

## Introduction

The first generation of supersonic civil aircraft is represented by two airliners: Tupolev Tu-144 developed in Soviet Union (produced in 1967–1983) and Concorde jointly developed by France and United Kingdom (produced in 1965–1979). Despite considerable scientific and industry expectations, the level of technologies and materials available at the time did not allow such planes to become widely used and economically feasible. The main reason was the intractable problem of sonic boom. When an aircraft travels at speeds greater than the local speed of sound, it generates a number of shockwaves that transform to a short intense acoustic disturbance at long distances perceived as an explosion or a thunderclap near the ground surface. For civil aircrafts, this effect led to a temporary prohibition of supersonic flights over populated areas which made manufacturing and maintaining costs of supersonic airliners unreasonable.

Technological advances of the last fifty years, characterized by development of new materials, evolution of aircraft engines, progress in computer-aided engineering based on numerical modeling, improvement of automated systems for diagnostics and control, provoke attempts to design supersonic transport of a new generation. The primary focus of the corresponding projects that exist in Russia and in some other countries is to develop a supersonic business jet (SSBJ) for a small number of passengers that can provide low intensity of sonic boom at supersonic cruise flight [20]. Important technical tasks also include optimization of airframe aerodynamics for all flight modes, achievement of high fuel efficiency and reduction of total noisiness. According to preliminary technological and economical assessments [20], the supersonic business transport of a new generation will be able to provide high level of passenger safety and sufficient comfort of the flight while remaining commercially reasonable.

As for other civil aircrafts, SSBJ is required to comply with the current certification standards of International Civil Aviation Organization (ICAO) for noise during takeoff, flyover and approach to receive the permission to land at most airports. For modern commercial airliners,

---

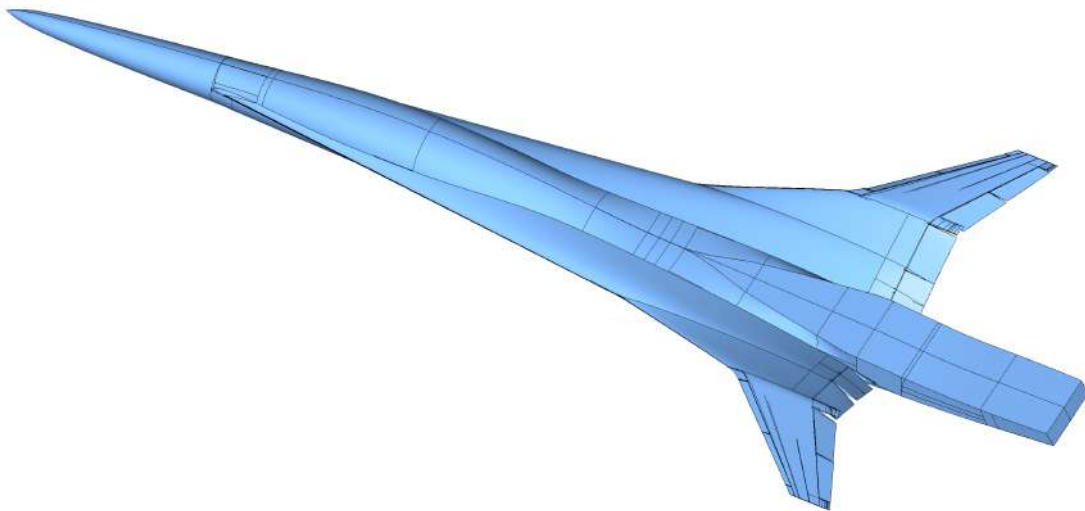
<sup>1</sup>Keldysh Institute of Applied Mathematics, RAS, Moscow, Russian Federation

the dominant component to the total noise during takeoff is generated by engine [23]. During landing, the noise sources associated with engine (primarily with fan and turbine) also make a significant contribution to the total noise, however, the high bypass ratio of modern civil aircraft engines led to a comparable importance of the noise generated by airframe elements such as landing gear, slats and flaps [5, 6, 23]. For SSBJ, engine and jet will probably be the dominant noise sources at both takeoff and landing due to the reduced engine bypass ratio and the wing design features. To provide data supporting this hypothesis, we aimed to investigate the SSBJ wing noisiness in landing mode by numerical simulations. The final confirmation of this hypothesis is possible only after corresponding numerical and/or experimental studies of SSBJ landing gear and engine which are out of scope of this paper.

We present the results of scale-resolving wing noise simulations of SSBJ prototype in landing mode based on Delayed Detached Eddy Simulation (DDES) approach [12, 17]. Recently, wing noise simulation of other SSBJ prototype in landing mode was performed by NASA in collaboration with Dassault Systmes using the PowerFLOW code [7, 11, 14] based on lattice Boltzmann method (LBM).

## 1. Problem Formulation

The full-scale SSBJ airframe with  $10^\circ - 10^\circ$  deflection of droop noises and  $10^\circ - 20^\circ - 20^\circ - 10^\circ$  deflection of elevons on each side of the wing (Fig. 1) is placed inside the uniform airflow with the velocity  $U_\infty = 68$  m/s, the pressure  $P_\infty = 101325$  Pa and the temperature  $T_\infty = 288.15$  K at an angle of attack  $10^\circ$ . The length of considered geometry is 45 m, the wingspan is 20 m. The corresponding Mach number is 0.2, the Reynolds number based on the characteristic length  $L = 1$  m is  $4.6 \times 10^6$ .



**Figure 1.** SSBJ airframe with high-lift devices in landing configuration

## 2. Computational Setup

To model the properties of air, we use the calorically perfect gas with the ratio of specific heats  $\gamma = 1.4$  and the specific gas constant  $R_{sp} = 287.05$  J/(kg K). For preliminary analysis of the flow, we perform simulations based on solving unsteady Reynolds-averaged Navier–Stokes

(RANS) equations with the Menter SST turbulence model adjusted by rotation and curvature correction of Stabnikov and Garbaruk [19]. We perform scale-resolving simulations according to the DDES approach [12, 17] with the subgrid scale  $\Delta = \tilde{\Delta}_\omega$  [12] and the subgrid model  $\sigma$  [13] in the large eddy simulation (LES) region and the Spalart–Allmaras (SA) turbulence model [18] in the RANS region.

Due to the symmetry of the considered geometry and the problem parameters, we simulate the flow only for half of the airframe. For visualization purposes, we duplicate and reflect the resulting flow fields relative to the plane of symmetry  $y = 0$ . All the acoustic data presented in the paper is calculated only for half of the airframe as well. Because the acoustic sources located on different sides of the airframe are spatially separated, we can consider them as uncorrelated. Hence, to obtain sound intensity for the full airframe, one can increase the corresponding intensity for half of the geometry by 3 dB.

The slip boundary conditions are set at the plane of symmetry  $y = 0$ , the free-stream conditions are used at the outer boundaries. Zero velocity and zero heat flux are specified on the streamlined geometry. To prevent the reflection of acoustic waves from the plane of symmetry in DDES simulations, the sponge layer [9] based on the averaged RANS solution is set in the region  $0 \text{ m} \leq z \leq 1.5 \text{ m}$ .

The computational domain is defined by the parallelepiped  $2000 \text{ m} \times 2000 \text{ m} \times 1000 \text{ m}$  with the exclusion of SSBJ airframe interior. The center of the boundary at the plane of symmetry is coincided with the reference point for pitching moment calculation located 31.5 m away from the SSBJ fore point along the  $x$ -axis.

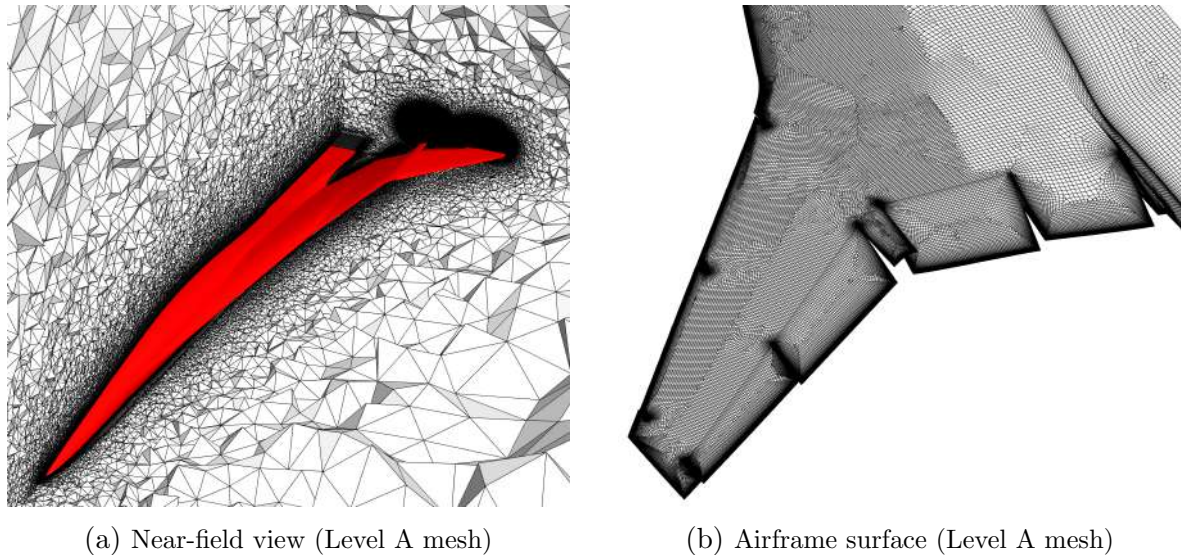
We use the finite-volume vertex-centered EBR5 PL scheme [4] to approximate the convective flux and the method of local element splittings [3] to approximate the diffusive flux. For time integration, we apply the first-order implicit scheme based on the backward differentiation formula (BDF1) in RANS simulations and the second-order implicit BDF2 scheme in DDES simulations. To solve the system of nonlinear equations at each time step, we use one Newton iteration in RANS simulations and two Newton iterations in DDES simulations. At each iteration, we solve of the system of linear equations by the bi-conjugate gradient stabilized (BiCGStab) method [22] with the symmetric Gauss–Seidel (SGS) preconditioner.

Simulations are performed on two meshes denoted as Level A and Level B. Their general structure is shown in Fig. 2, their parameters are summarized in Tab. 1, where  $h_{\text{fuselage}}$  is the length of mesh edges in tangential directions near the fuselage and the lower surface of the wing,  $h_{\text{vortices}}$  is the length of mesh edges in the region of stable vortex flow over the wing. Outside the prismatic layers built near the streamlined geometry, the zone of increased mesh resolution over the wing is filled with an isotropic unstructured tetrahedral mesh.

**Table 1.** Mesh parameters

Mesh	$N_{\text{nodes}}$	$N_{\text{elements}}$	$N_{\text{surf.nodes}}$	$N_{\text{surf.elements}}$	$h_{\text{fuselage}}$	$h_{\text{vortices}}$
Level A	21 166 948	46 552 132	337 330	342 475	70 mm	35 mm
Level B	61 601 940	219 587 977	678 233	687 362	70 mm	17.5 mm

In RANS simulations, we use the maximum time step providing stability of the computational process. After reaching the steady state, the numerical solution is averaged over time interval 50–150  $L/U_\infty$  to obtain the resulting flow fields.



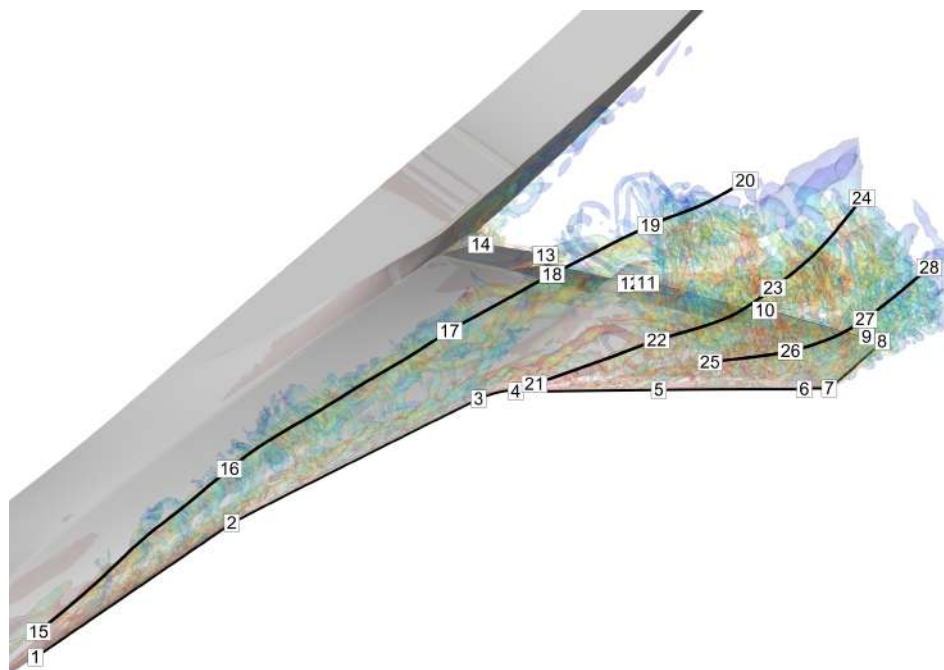
**Figure 2.** General mesh structure

In DDES simulations, we decrease the weight of upwind component of the EBR5 PL scheme according to the approach proposed in [10]. The minimum weight of upwind component in the zone of increased mesh resolution over the wing is set to 0.15. We choose the time step providing the relatively small size of regions containing numerical instability. In terms of  $CFL_{\text{vortices}} = \Delta t \times (c_\infty + U_\infty) / h_{\text{vortices}}$ , where  $\Delta t$  is the time step and  $c_\infty$  is the speed of sound at infinity, we use  $CFL_{\text{vortices}} = 0.083$  on the Level A mesh and  $CFL_{\text{vortices}} = 0.125$  on the Level B mesh. The initial flow fields for the DDES simulation on the Level A mesh are defined by the averaged RANS solution. The initial flow fields for the DDES simulation on the Level B mesh are defined by instantaneous DDES solution obtained on the Level A mesh after reaching the steady state according to aerodynamic coefficients. When the flow is reached the steady state and the instantaneous solution is proved to have only small regions of numerical instability, we start to record the near-field acoustic data and accumulate the average flow fields. The data recording is performed for time interval  $60 L/U_\infty$  or 0.88 s. This interval size allows us to obtain smoothed spectra (averaged for 30 time segments with 0.5 overlapping) at the near-field and far-field points with the minimum resolved frequency 20 Hz.

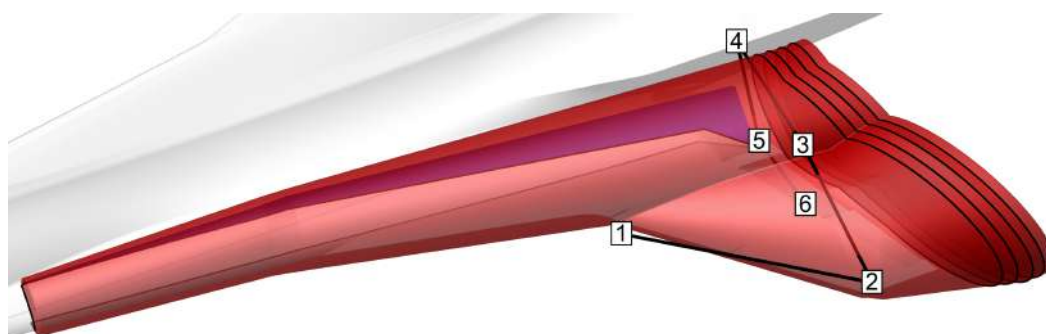
To evaluate the acoustic characteristics of the flow in the near field, pressure pulsations are recorded at the points of the discretized curves shown in Fig. 3. The approximate distance between these points is 10 cm, some of these points are marked with numerical labels. The lower curve is located approximately 4 cm below the wing edge, the upper curves are located approximately above the centers of the main vortices formed over the wing.

The second-order Ffowcs Williams–Hawkings (FWH) method [2, 8, 15, 16] was used to calculate acoustic pressure pulsations in the far field. The corresponding FWH surface with five end caps used for accumulation of the required acoustic data is located near the boundaries of the zone of increased mesh resolution (Fig. 4). Note that this surface has a slit on the fuselage side to prevent intersection with the wing surface. Formally, a non-closed surface is not allowed to be used for the FWH method, however, the construction of a closed surface for the considering problem is undesired for the following reasons.

If we locate the FWH surface at a significant distance from the SSBJ airframe, as it was done, for example, in [11, 14], the requirement to resolve acoustic pulsations on this surface will



**Figure 3.** Location of pressure sensors in the near field



**Figure 4.** Location of FWH surface and the near-field points used to test FWH method

lead to the use of increased mesh resolution in the vicinity of the entire SSBJ airframe, resulting in a significant increase of computational cost of scale-resolving modeling. This approach would be rational if substantial acoustic sources are located along entire streamlined body. However, in the considered problem, the fuselage generates almost no large-scale turbulent pulsations that have any substantial effect on the total airframe noise.

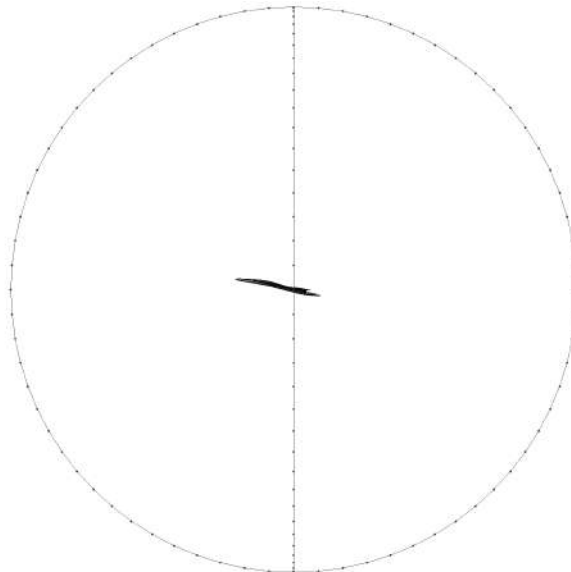
In theory, one could propose to use a closed FWH surface formed by the union of the surface shown in Fig. 4 with some surface located in a slight distance from the streamlined geometry and containing the fuselage and wing root within. However, since the mesh resolution near the fuselage and wing root is coarser than the resolution inside the zone of intense vortical flow, acoustic perturbations propagating from the main vortices toward the plane of symmetry will rapidly dissipate, and hence their contribution to the far-field noise obtained by the FWH method will be close to zero. The use of increased mesh resolution near the fuselage and wing root will lead to unjustified increase of computational cost similar to the previous variant.

The main problem originating from the location of the FWH surface according to Fig. 4 is the inability to model the reflection of acoustic waves from the wing surface by the FWH method.

In scale-resolving simulation, acoustic waves propagating downward near the upper surface of the wing will be reflected. However, in calculation according to the FWH method, these waves will propagate to the region below the wing without any reflection. In order to evaluate the influence of the described effect on the far-field noise, the constructed surface was divided into two parts: the main surface denoted in Fig. 4 by red color, and the extension denoted in Fig. 4 by blue color. Further, we will use the label FWH to denote noise calculations based on the data from the main surface only, and the label FWH Ext to denote noise calculations based on the data from both the main surface and the extension. Note that all end caps (e.c.) belong to the main surface.

To find optimal mesh resolution for the FWH surface, we accumulate the required acoustic data on the three types of meshes formed mainly by quadrilaterals. We will use the label Coarse for isotropic meshes with edge length  $2h_{\text{vortices}}$ , the label Fine for isotropic meshes with edge length  $h_{\text{vortices}}$ , and the label Mixed for meshes with edge length  $h_{\text{vortices}}$  at end caps and edge length  $2h_{\text{vortices}}$  at the rest part of the FWH surface. We record the data with the sampling frequency  $(c_{\infty} + U_{\infty})/(2h_{\text{vortices}})$  on Coarse and Mixed meshes, and with the sampling frequency  $(c_{\infty} + U_{\infty})/h_{\text{vortices}}$  on Fine meshes.

The described methodology of noise calculation based on the FWH method is tested by comparing the acoustic spectra obtained by DDES simulations and FWH calculations at the near-field points. These points are located in the outer region relative to the FWH surface and denoted by numerical labels in Fig. 4. The far-field points used for SSBJ wing noise assessment (Fig. 5) belong to the sphere of radius 150 m.



**Figure 5.** Location of the far-field points

All the simulations presented in this paper are performed using the NOISEtte code [1] written in C++ and suitable for computations in CPU, GPU (OpenCL) and heterogeneous CPU+GPU modes with combined MPI+OpenMP parallelization. DDES simulations are carried out using NVIDIA Tesla V100 GPUs on the Lomonosov-2 supercomputer [21] installed at Lomonosov Moscow State University. For the DDES simulation on the Level A mesh, 8 GPUs (4 compute nodes each equipped with 2 GPUs) are utilized for 21 hours to accumulate the required data on the time interval  $60 L/U_{\infty}$ . For the DDES simulation on the Level B mesh,

24 GPUs (12 compute nodes) are utilized for 24 hours to achieve the steady flow state and for 72 hours to accumulate the required data on the time interval  $60 L/U_\infty$ .

### 3. DDES Performance

For reliable flow modeling using the hybrid RANS-LES approach DDES, boundary layer should be simulated in RANS mode because mesh for DDES simulation is not fine enough in tangential direction near walls to resolve boundary layer in LES mode properly. The switching between RANS and LES modes in DDES is controlled by the  $f_d$  blending function.

DDES performance in the simulation of the flow around SSBJ airframe on the Level B mesh can be evaluated from Fig. 6 and Fig. 7. Isolines  $f_d = 0.99$  indicating approximate interface between RANS and LES zones are depicted in Fig. 6d for the instantaneous DDES solution (location of the corresponding cross sections is shown in Fig. 6a and Fig. 6b). Figure 6c presents isolines  $F_1 = 0.99$  of the SST  $F_1$  blending function for the averaged RANS solution. These isolines approximately correspond to the edge of boundary layer. Distributions of the distance to the wall  $d_w$  for the considered isolines as functions of the spanwise coordinate are shown in Fig. 6e for Section 1 and in Fig. 6f for Section 2. In most of the domain, the approximate interface between RANS and LES zones ( $f_d = 0.99$ ) lies farther from the wall than isosurface  $F_1 = 0.99$ , hence the boundary layer is simulated predominantly in RANS mode by DDES. Distributions of the friction coefficient  $C_f$  for the considered cross sections (Fig. 7) demonstrate that, except the regions with resolved turbulence, the averaged DDES solution is close to the averaged RANS solution, even though in the corresponding simulations the different turbulence models are used (SA in DDES and SST in RANS). Figure 6b shows the instantaneous distribution of the ratio between the subgrid scale  $\Delta$  and the distance to the wall  $d_w$  on isosurface  $f_d = 0.99$ . In the regions with no resolved turbulence, this ratio is mostly close to 1, while in the areas with stable vortical flow above the wing, where mesh resolution is intentionally better, it generally belongs to the interval  $0.2 \lesssim \Delta/d_w \lesssim 0.6$ .

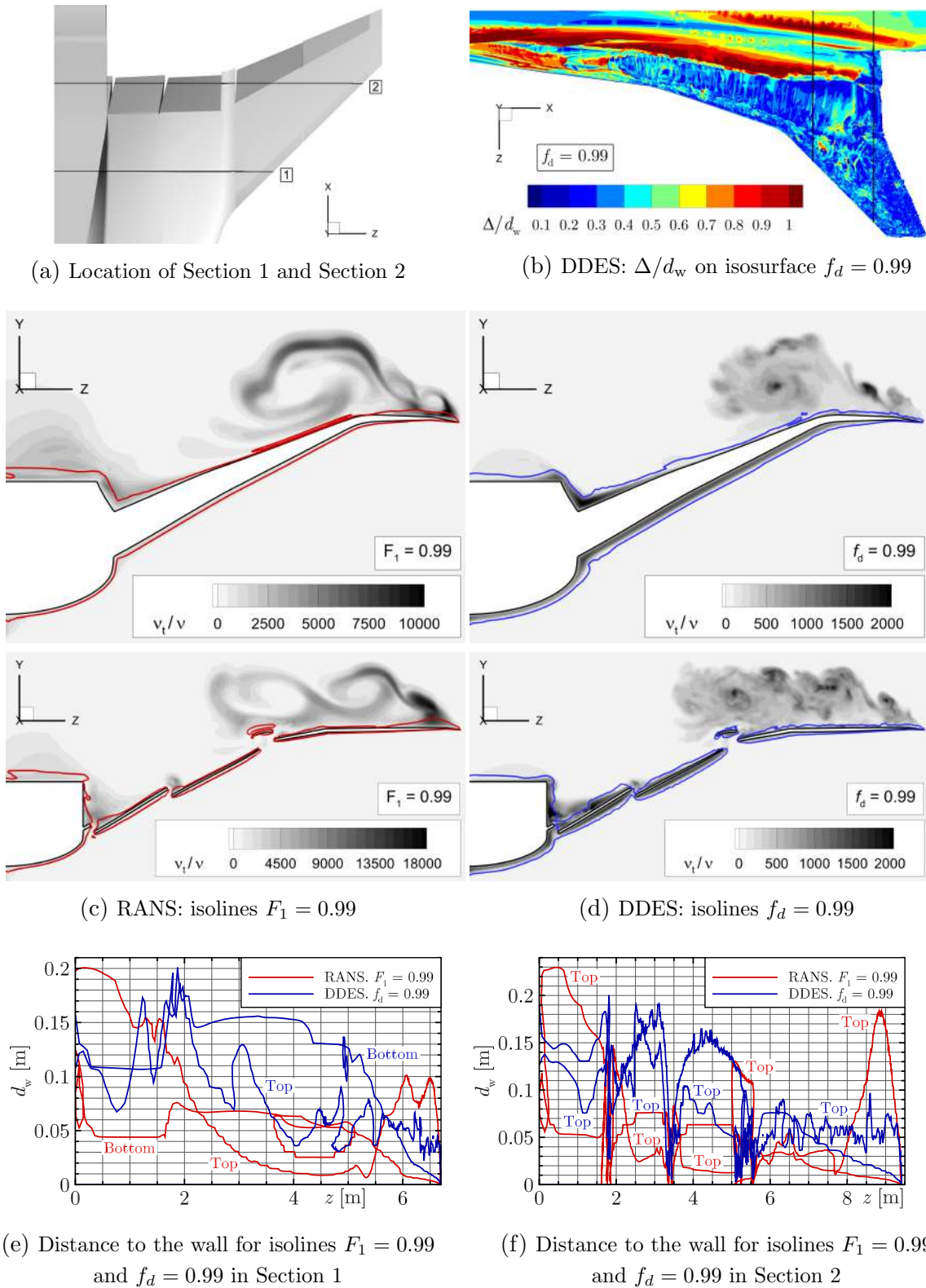
Thus, we can conclude that the boundary layer shielding, in the sense of protecting boundary layer from switching to the unresolved LES regime, is effective and sufficient in the considered DDES simulation.

### 4. Aerodynamics

The mean flow fields obtained by DDES simulations are shown in Fig. 8. We see that the stable macro-scale vortices are formed over the wing at the considered angle of attack. These vortices provide a substantial region of rarefaction on the upper surface of the wing, which increases the airframe lift force. Despite some insignificant differences, the mean flows obtained on the Level A and Level B meshes are very close.

The values of aerodynamic coefficients obtained by RANS and DDES simulations on different meshes are given in Tab. 2. We see that the difference between the results of RANS and DDES simulations is approximately 3% in the lift coefficient, approximately 5% in the drag coefficient, and 10–15% in the pitching moment coefficient. We also note that mesh refinement slightly increases the difference between the RANS and DDES solutions in lift and pitching moment coefficients while the difference in drag coefficient remains almost unchanged.





**Figure 6.** Boundary layer thickness: averaged RANS and instantaneous DDES solutions on the Level B mesh ( $\Delta$  is the subgrid scale,  $d_w$  is the distance to the wall,  $F_1$  is the SST blending function,  $f_d$  is the DDES blending function;  $\nu$  is the kinematic viscosity,  $\nu_t$  is the kinematic turbulent viscosity)

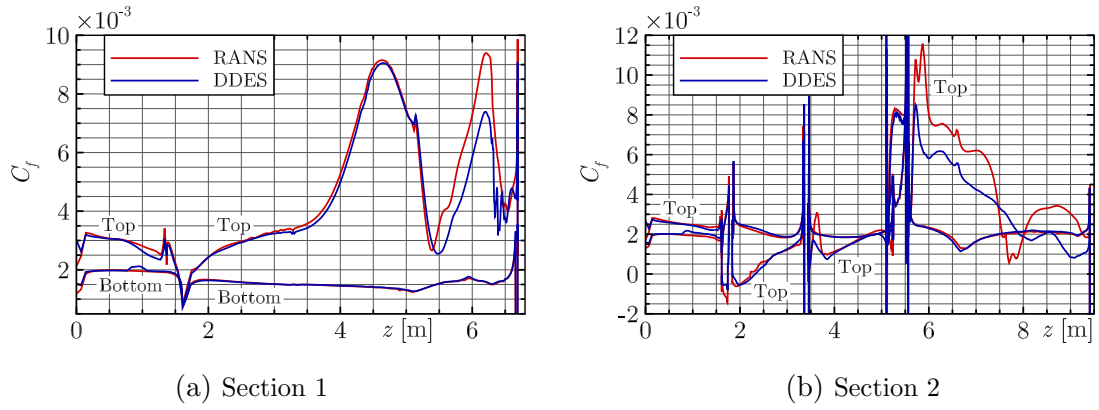


Figure 7. Skin friction coefficient: averaged RANS and averaged DDES solutions

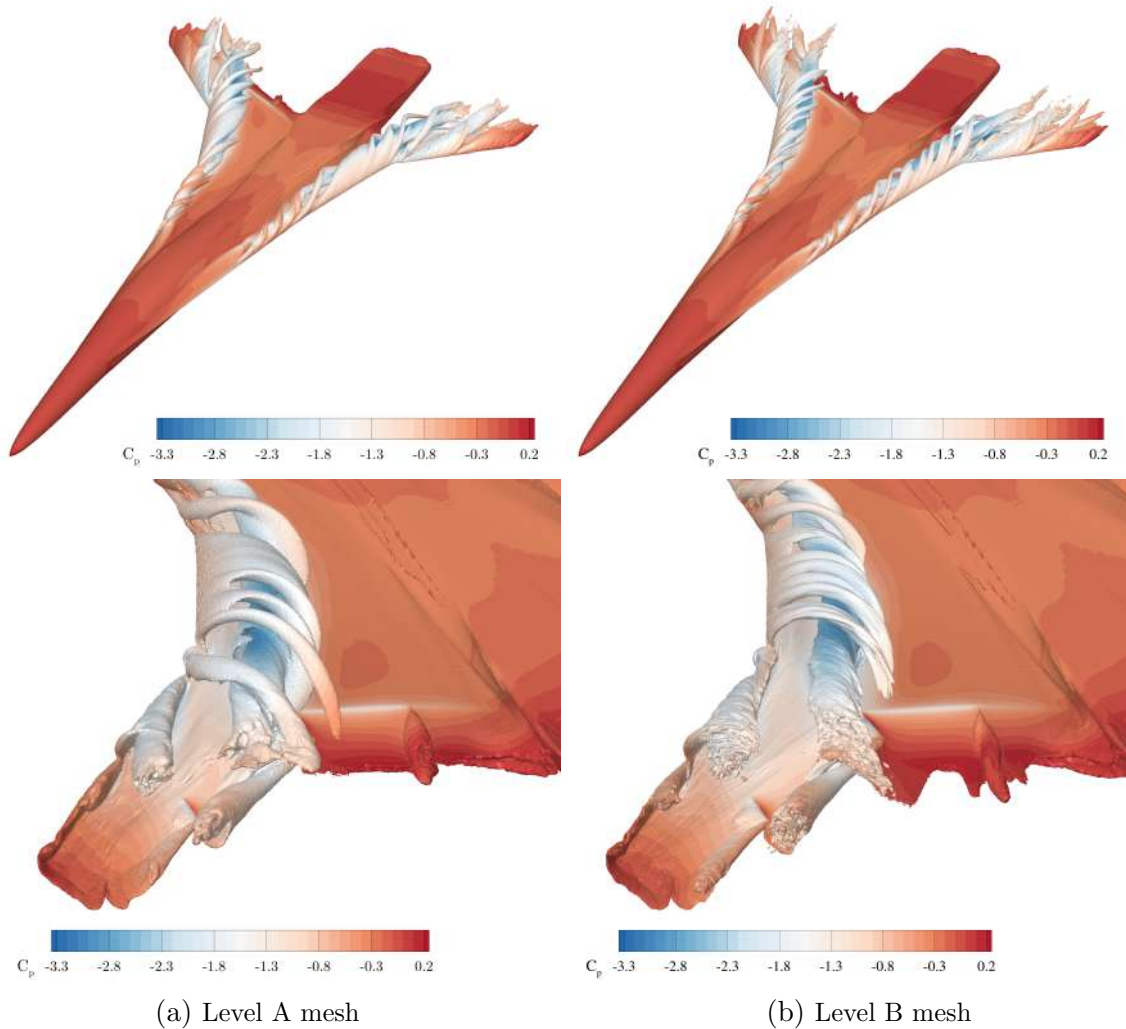


Figure 8. Mean flow fields obtained by DDES simulations (isosurfaces of vorticity magnitude corresponding to the value 200 1/s colored by pressure coefficient)

**Table 2.** Lift (CL), drag (CD) and pitching moment (CM) coefficients

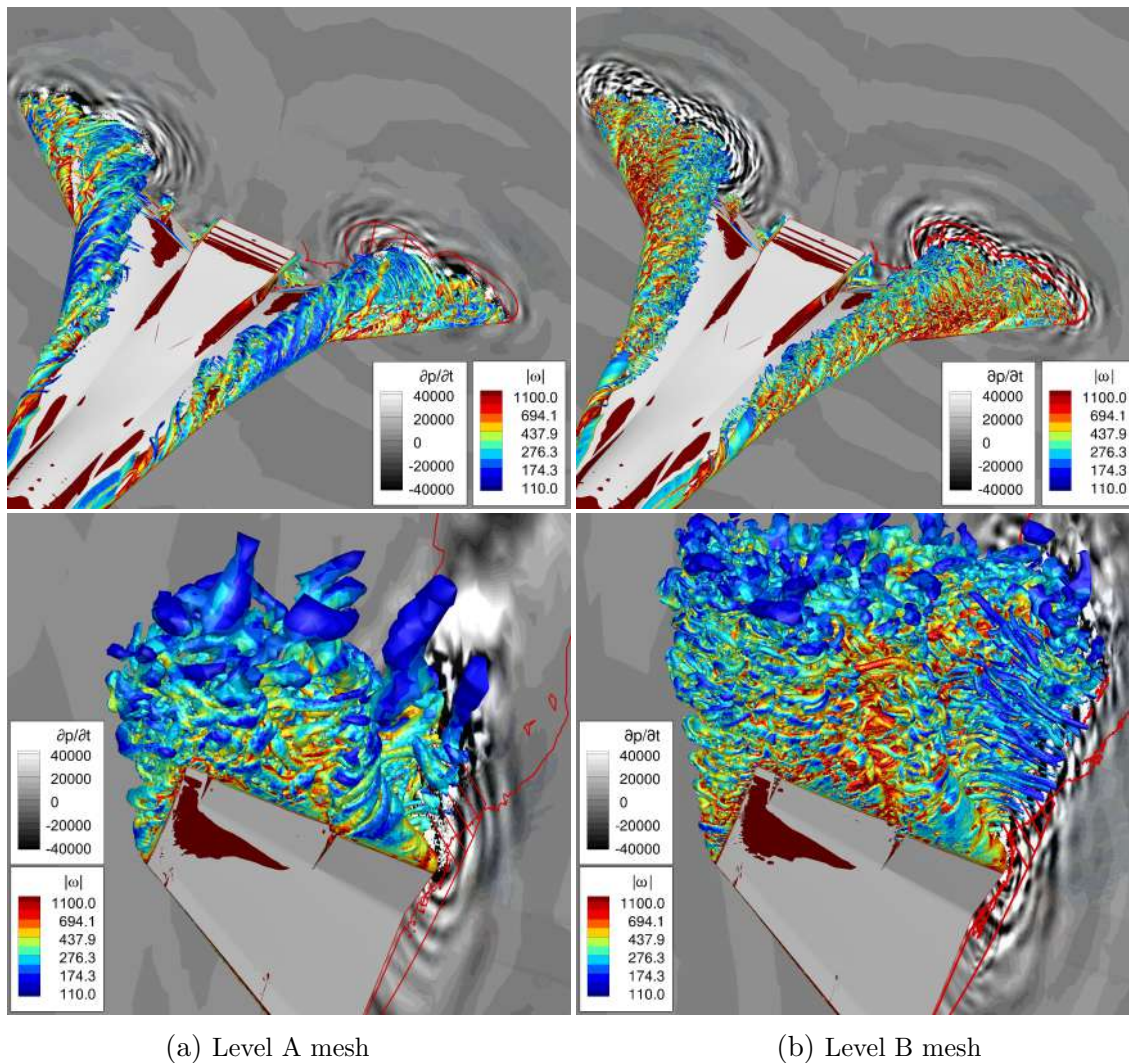
	CL	diff	%	CD	diff	%	CM	diff	%
RANS (Level A)	0.907	0.025	2.9%	0.2244	0.0102	4.8%	-0.0638	-0.0060	10.3%
RANS (Level B)	0.914	0.032	3.7%	0.2255	0.0113	5.3%	-0.0677	-0.0099	17.1%
DDES (Level A)	0.878	-0.004	0.4%	0.2132	-0.0010	0.5%	-0.0566	0.0013	2.2%
DDES (Level B)	0.881	0	0.0%	0.2142	0	0.0%	-0.0578	0	0.0%

## 5. Near-Field Acoustics

The instantaneous flow fields obtained by DDES simulations after reaching the steady state are shown in Fig. 9. We see that the turbulent vortical flow over the wing surface is a source of acoustic pulsations. In the vicinity of the fuselage, the flow is almost stationary and does not contain significant acoustic sources. As it was expected, the mesh refinement allow DDES method to reproduce smaller turbulent structures above the wing, which leads to the appearance of higher-frequency harmonics in the simulated noise. Outside the zone of increased mesh resolution over the wing, acoustic pulsations rapidly dissipate due to increasing size of mesh edges.

The noise spectra at points of the near-field curves (Fig. 3) are shown in Fig. 10. We see that the presented spectra are more broadband near the geometry corners of the wing and its high-lift devices comparing to the rest of the computational domain. This effect is caused by a higher mesh resolution used for proper discretization of the corresponding geometry features. The resulting high-frequency components of the noise can be of a pure numerical nature or even related to numerical instability since a much coarser mesh is used at the rest of the domain. As we move along the wing edge toward the wing tip, the spectra shift almost linearly toward the low frequencies. This feature can be explained by the gradual enlargement of the stable vortices formed over the wing, which are the main sources of the wing noise. At the points close to the wing edge, the spectra contain narrowband peaks at high frequencies that appear to be very sensitive to the mesh resolution. These peaks are the footprints of the small local regions of numerical instability dependent on mesh and parameters of numerical method. On the curve located above the center of the main vortex, the numerical instability does not arise due to the relatively large size of the corresponding mesh elements. Hence, the spectra on this curve does not contain narrowband high-frequency peaks and sudden expansions in the resolved frequency range.

The acoustic spectra calculated for the near-field points marked with numerical labels (Fig. 3) are shown in Fig. 11. In accordance with the results presented in Fig. 10, these spectra are broadband, and, at most points, the intensity of noise decreases with increasing frequency. Similar features of noise spectra were obtained in [11, 14]. As location and size of regions containing numerical instability depend on mesh, the comparison of the spectra obtained by scale-resolving simulations on the Level A and Level B meshes allow us to evaluate the impact of numerical instability on the resulting flow acoustics. For example, when switching from the Level A mesh to the Level B mesh, the narrowband high-frequency peaks disappear from the spectra for points 11, 12, while such peaks arise in the spectra for points 5, 6, 10. Note that these narrowband peaks do not affect considerably the general features of the noise spectra in the near field.

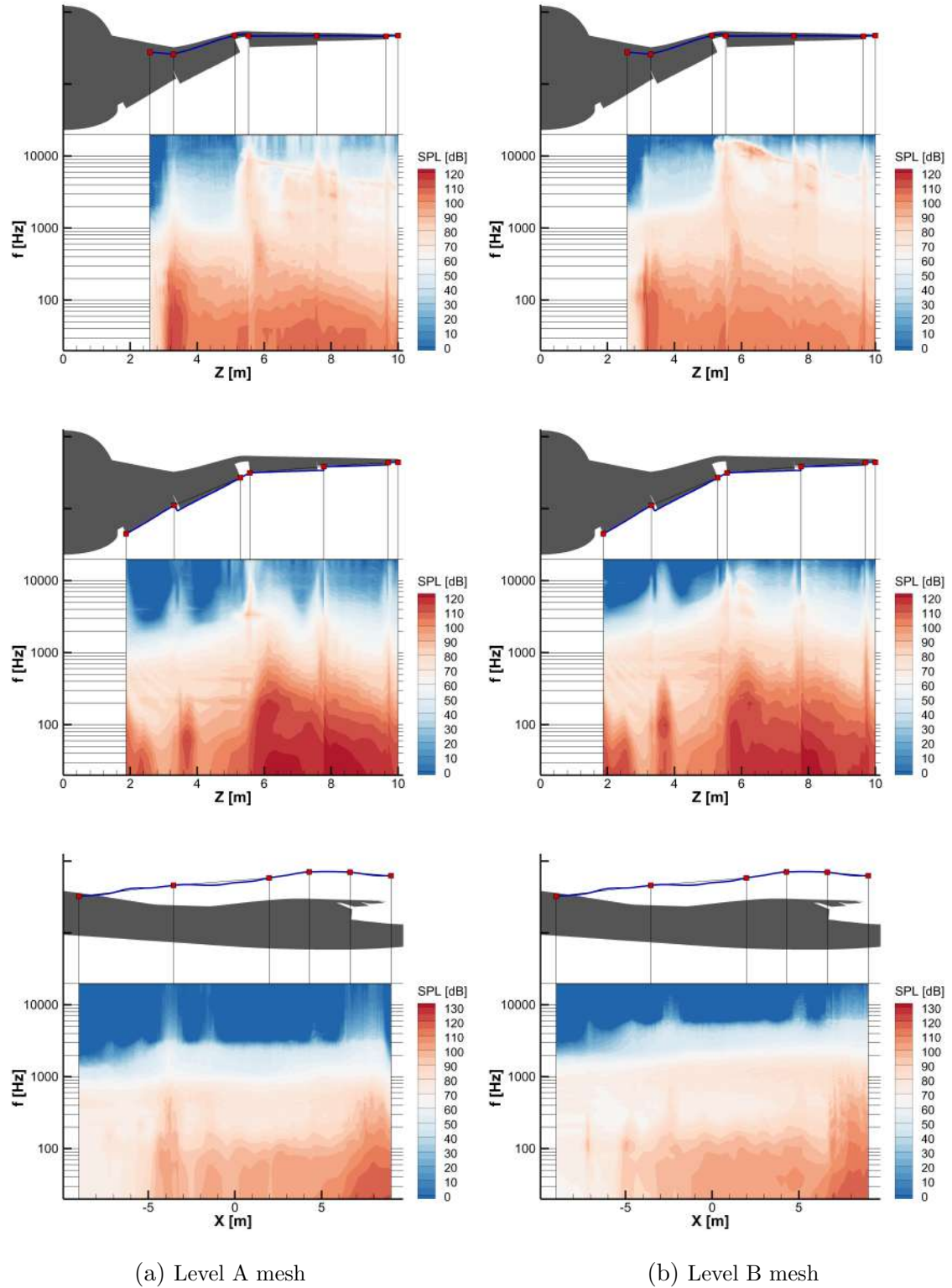


**Figure 9.** Instantaneous flow fields obtained by DDES simulations (time derivative of pressure and Q-criterion isosurfaces corresponding to the value  $5000 \text{ 1/s}^2$  colored by vorticity magnitude). The smooth curves denote position of the FWH surface, the non-smooth curves denote location of the isosurface of mean vorticity magnitude corresponding to the value  $2 \text{ 1/s}$

## 6. Far-Field Acoustics

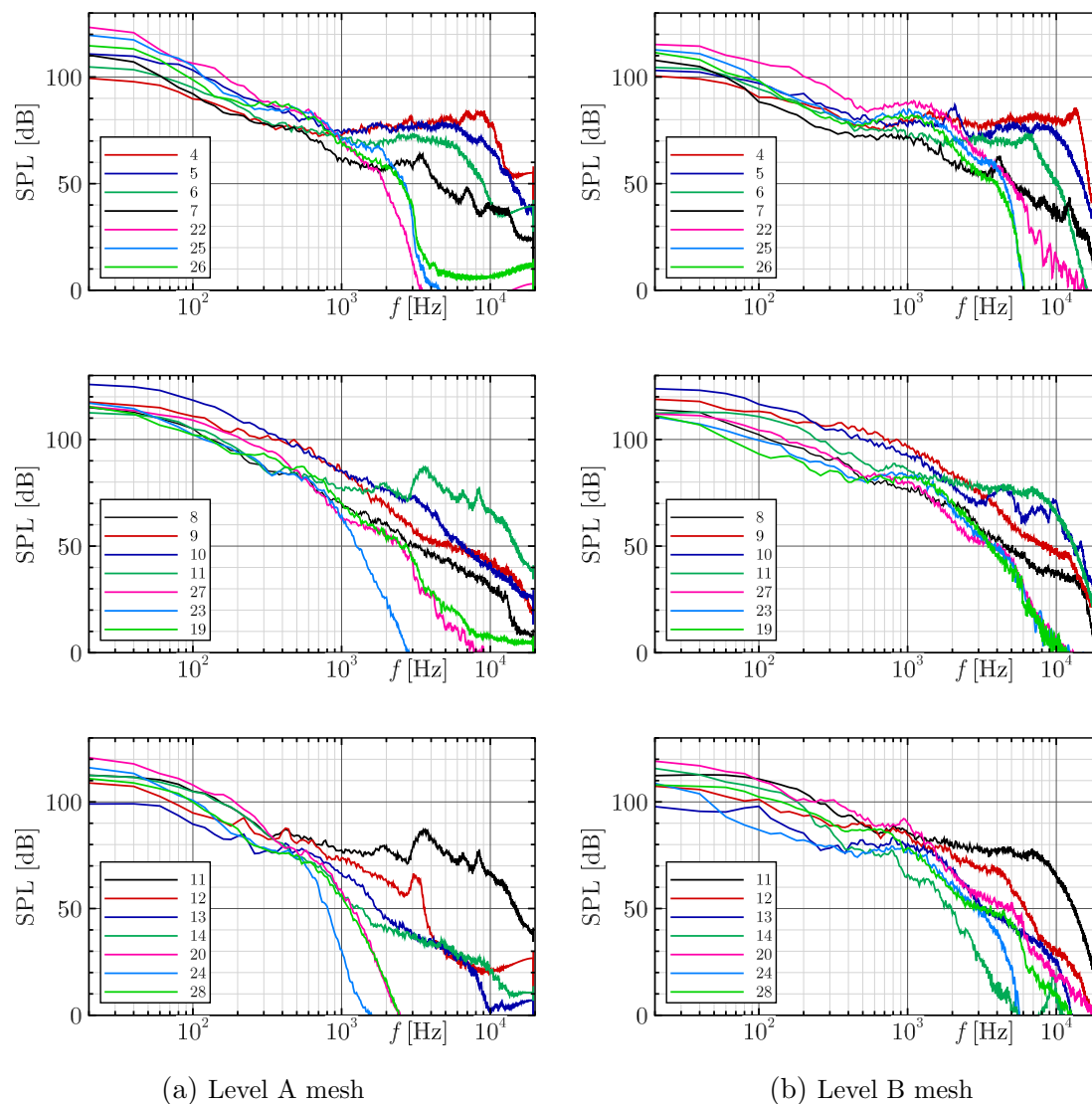
To test the approach to far-field noise calculation described in Section 2, we compare the spectra based on pressure history extracted directly from the DDES solution and the spectra calculated using the FWH method at the near-field points marked with numerical labels in Fig. 4.

Comparison of noise spectra calculated using the Coarse and Fine FWH meshes is shown in Fig. 12. We see that for the Coarse FWH mesh the use of 5 end caps is preferable because the results for 3 end caps demonstrate an increase of error for some frequency bands by approximately 2 dB. For the Fine FWH mesh, there is almost no difference between the results obtained with 3 and 5 end caps. Note that the use of the Coarse FWH mesh is justified as the high-frequency range [5 kHz, 10 kHz] resolved by the Fine FWH mesh is not properly resolved in the DDES simulation. At the considered near-field points, the Coarse and Fine FWH meshes without end caps provide almost identical spectra, very close to the spectra based on pressure history. The spectra obtained using the Fine FWH mesh with 1 end cap are appeared to be



**Figure 10.** Noise spectra in the near field

more accurate than the spectra obtained using the Coarse FWH mesh with 1 end cap. Hence, to reduce the volume of accumulated data, one may replace the Fine FWH mesh by the Mixed FWH mesh (and use the sampling frequency corresponding to the Coarse FWH mesh) and receive almost the same results in the far field.

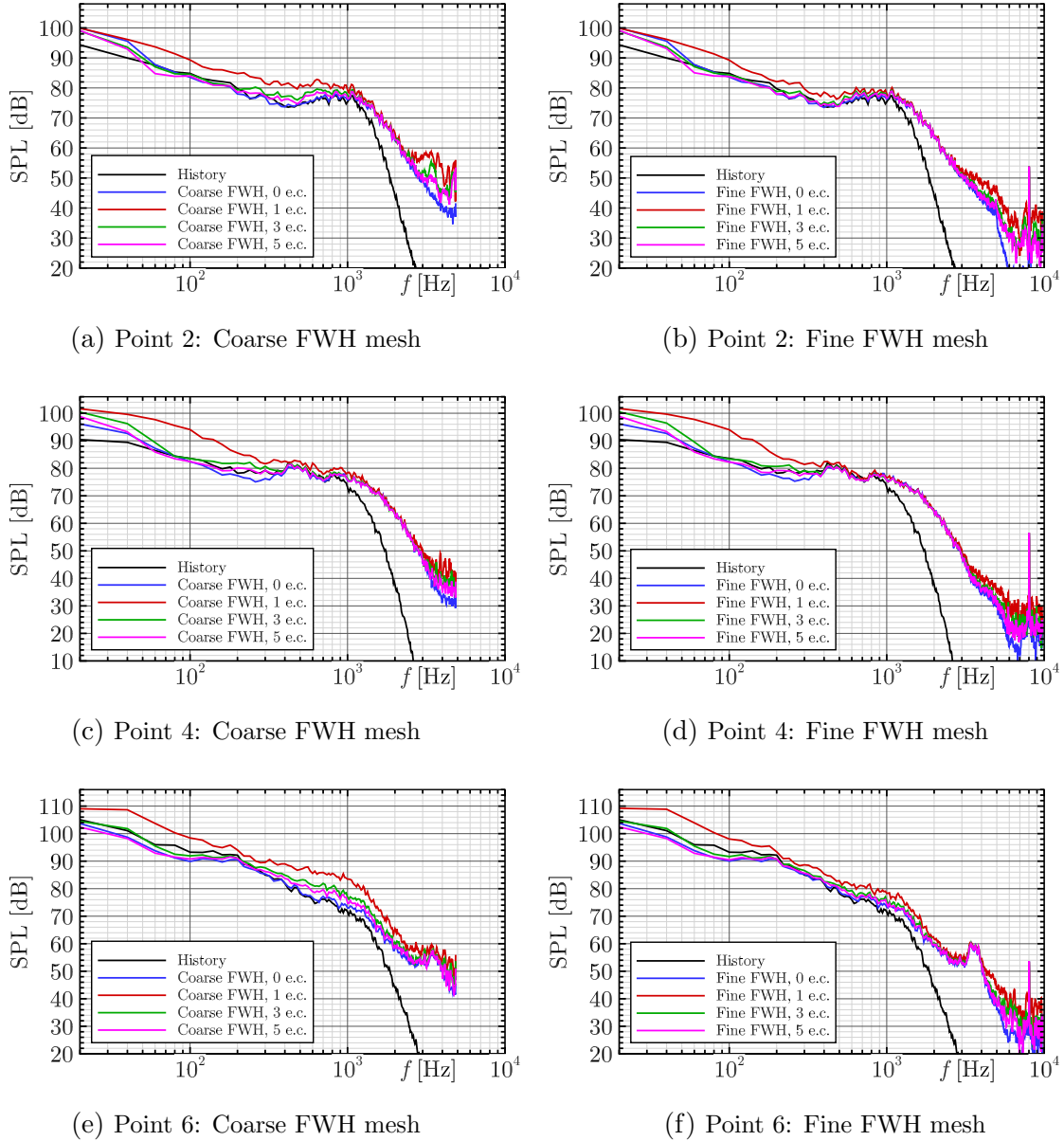


**Figure 11.** Noise spectra in the specific near-field points

For comparison of the FWH and FWH Ext surfaces, we choose the point closest to the FWH extension (Point 5 in Fig. 4). Figure 13 shows that the main difference between the basic and extended FWH surfaces appears in the range [40 Hz, 400 Hz]. We see that this difference is limited by 6 dB while the difference between the spectra calculated by the FWH method and the spectra based on pressure history in the range [40 Hz, 400 Hz] reaches 4–6 dB at some frequencies. At other test points, the difference between the basic and extended FWH surfaces is barely recognizable.

The spectra obtained by the FWH method at far-field points (Fig. 5) depending on the DDES and FWH meshes are presented in Fig. 14. We see that the spectra based on the data extracted from DDES simulations on the Level A and Level B meshes quantitatively differ by about 2.5 dB over a wide frequency range. The difference between the spectra calculated using the Coarse and Fine FWH meshes is less than 1 dB at most far-field points and frequency ranges.

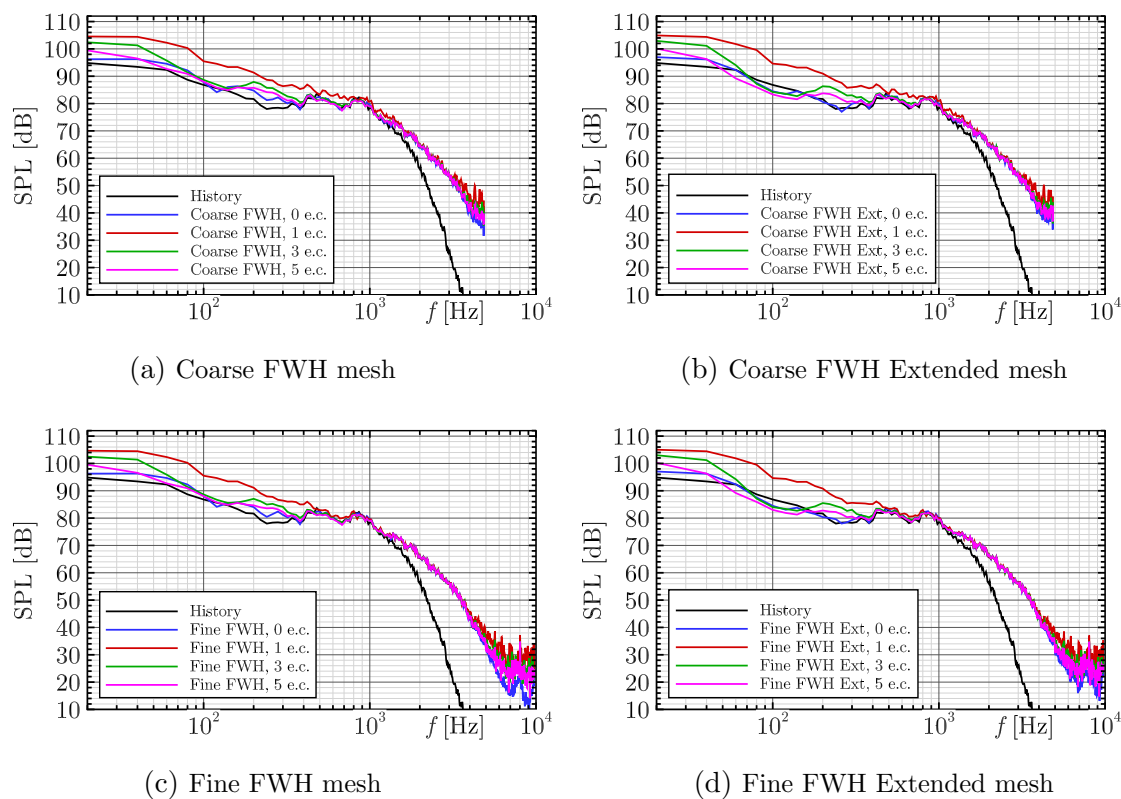
Dependence of the far-field noise spectra on the use of the extended FWH surface is shown in Fig. 15. The main impact of the FWH extension is concentrated in the low-frequency range



**Figure 12.** Comparison of noise spectra extracted directly from the DDES solution (History) and calculated using the FWH method (Coarse FWH, Fine FWH). Raw numerical data is accumulated during the DDES simulation on the Level B mesh

[20 Hz, 100 Hz], where the difference between the FWH and FWH Ext surfaces can reach 2–5 dB. Figure 15 also presents the far-field noise spectra calculated using the Mixed FWH mesh. We see that the spectra obtained using the Mixed and Fine FWH meshes differ only at the high-frequency range, which is not properly resolved in the DDES simulation.

We note that the use of the Coarse FWH mesh instead of the Fine FWH mesh leads to an 8-fold reduction in disk space required for storing FWH data and a 2-fold reduction in the FWH sampling frequency in DDES simulation. For example, in DDES simulation on the Level B mesh, 523 GB and 72 GB of FWH data is accumulated for the Fine FWH mesh and for the Coarse FWH mesh, respectively. Because the corresponding impact on the spectra is within 1 dB, this approach is justified from a practical point of view. If higher accuracy is needed, one can use



**Figure 13.** Comparison of noise spectra at Point 5 extracted directly from the DDES solution (History) and calculated using the FWH method. Raw numerical data is accumulated during the DDES simulation on the Level B mesh

the Mixed FWH mesh instead of the Fine FWH mesh. In DDES simulation on the Level B mesh, the Mixed FWH mesh provides a 3.3-fold reduction in disk space required for storing FWH data (158 GB of accumulated FWH data instead of 523 GB) along with a 2-fold reduction in the FWH sampling frequency. If the area of end caps is small relative to the area of entire FWH surface, the use of the Mixed FWH mesh instead of the Fine FWH mesh would lead to approximately an 8-fold reduction in the required disk space.

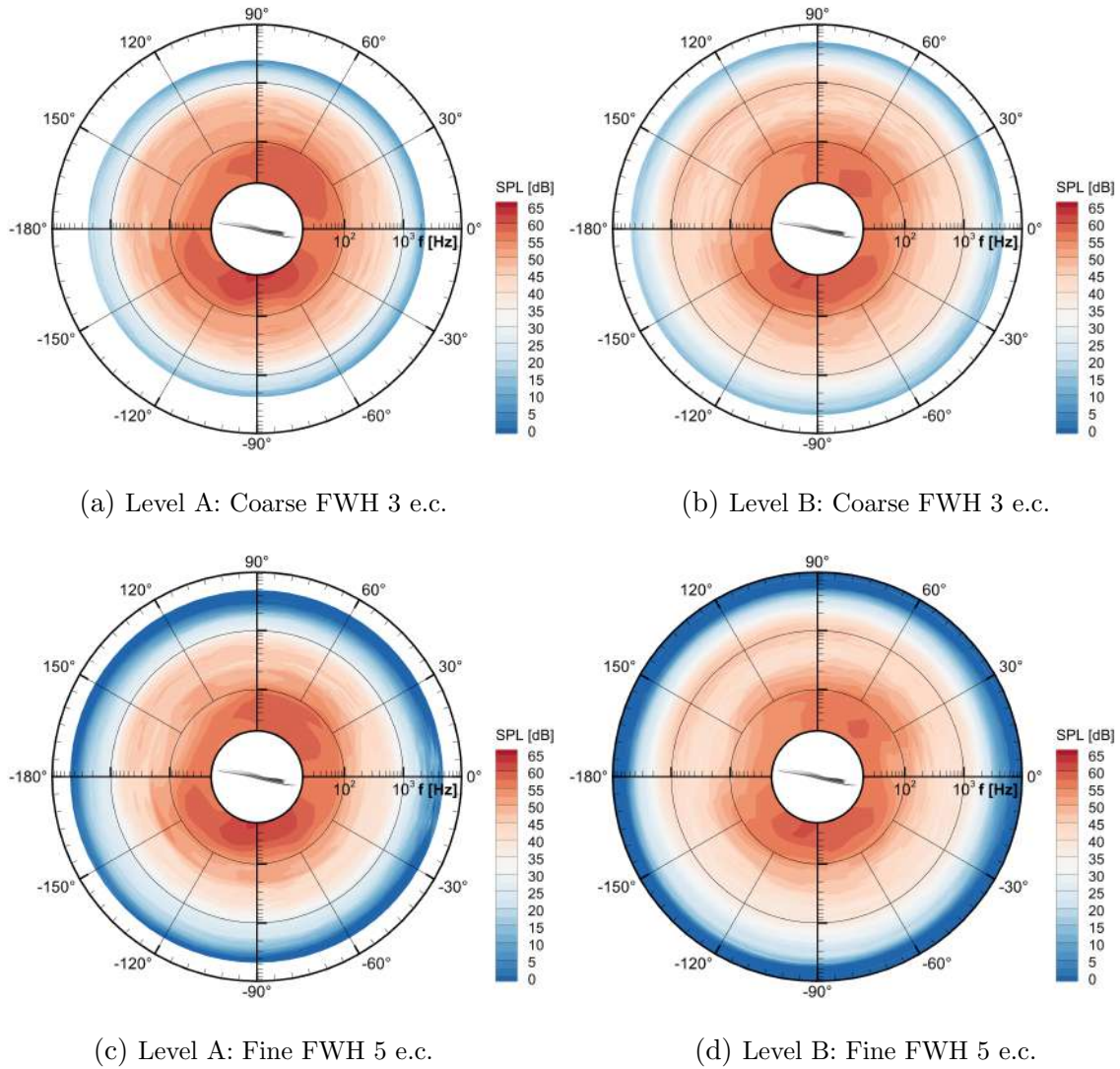
## Conclusions

The performed scale-resolving simulations of the wing noise for the supersonic business jet in landing configuration demonstrated the ability of hybrid RANS-LES methods to successfully solve the challenging aviation problems. The computations based on DDES approach allowed us to investigate aerodynamics of the target flow, accumulate and analyze the acoustic data in the near field, and calculate the far-field noise using the FWH method. The obtained results can be used for estimation of the total noise of supersonic business jets in landing mode for the international aircraft noise certification.

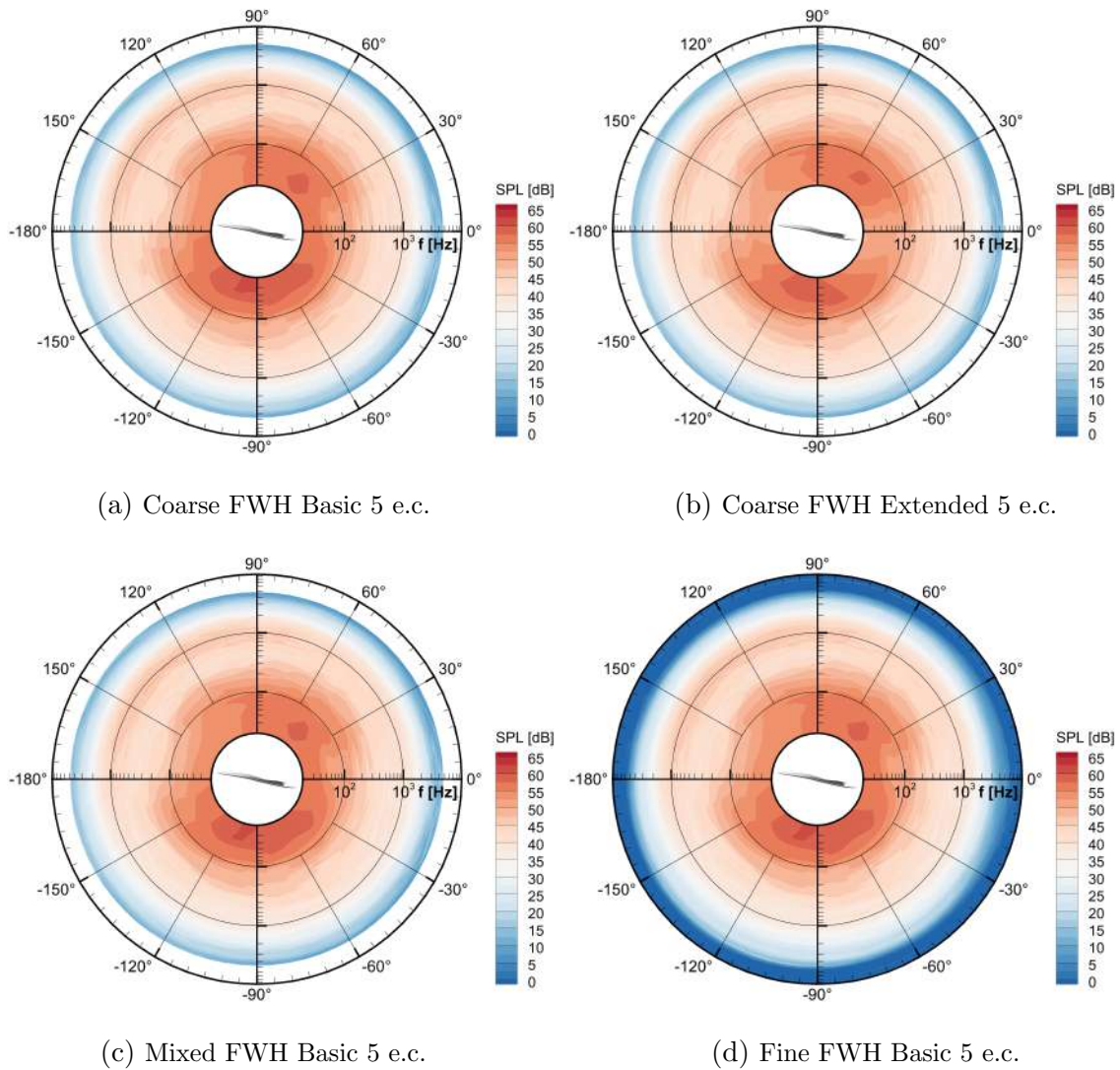
Particular attention in the presented study is paid for parameters of the FWH method. The considered wing geometry, the features of the target flow and the desire to reduce computational cost of DDES simulations provoked construction of the non-standard FWH surfaces. These surfaces were placed around the region of the intense vortical flow, but did not contain the



whole streamlined geometry within. The tests at the near-field and far-field points showed the applicability of the proposed surfaces for the FWH method. We also investigated the impact of the FWH surface discretization on properties of the calculated signals, and, finally, proposed the parameters of the mixed FWH mesh resolution that allows to save up to 8 times disk space required for storing FWH data.



**Figure 14.** Noise spectra of the SSBJ wing in landing configuration at the far-field points. Raw numerical data is accumulated on the Coarse and Fine FWH surface meshes during DDES simulations on the Level A and Level B meshes



**Figure 15.** Noise spectra of the SSBJ wing in landing configuration at the far-field points. Raw numerical data is accumulated on the Coarse, Fine and Mixed FWH surface meshes during the DDES simulation on the Level B mesh

## Acknowledgements

The paper is prepared in the implementation of the program for the creation and development of the World-Class Research Center “Supersonic” for 2020–2025 funded by the Ministry of Science and Higher Education of the Russian Federation (Grant agreement of 25.04.2022, № 075-15-2022-330). The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University with the additional use of the hybrid supercomputer K60 installed in the Supercomputer Centre of Collective Usage of KIAM RAS.






*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. Abalakin, I.V., Bakhvalov, P.A., Bobkov, V.G., *et al.*: NOISEtte CFD&CAA Supercomputer Code for Research and Applications. *Supercomput. Front. Innov.* 11(2), 78–101 (aug 2024). <https://doi.org/10.14529/jsfi240206>
2. Bakhvalov, P.A., Kozubskaya, T.K., Kornilina, E.D., *et al.*: Technology of predicting acoustic turbulence in the far-field flow. *Math. Model. Comput. Simulations* 4(3), 363–373 (may 2012). <https://doi.org/10.1134/S2070048212030039>
3. Bakhvalov, P.A., Surnachev, M.D.: Method of averaged element splittings for diffusion terms discretization in vertex-centered framework. *J. Comput. Phys.* 450, 110819 (feb 2022). <https://doi.org/10.1016/J.JCP.2021.110819>
4. Bakhvalov, P., Kozubskaya, T., Rodionov, P.: EBR schemes with curvilinear reconstructions for hybrid meshes. *Comput. Fluids* 239, 105352 (may 2022). <https://doi.org/10.1016/J.COMPFLUID.2022.105352>
5. Delfs, J.: Simulation of aircraft installation noise – a key to low noise aircraft design (2016), [https://ceaa.imamod.ru/2016/files/ceaa2016.pdfs/D3S01\\_Delfs.pdf](https://ceaa.imamod.ru/2016/files/ceaa2016.pdfs/D3S01_Delfs.pdf)
6. Dobrzynski, W., Ewert, R., Pott-Pollenske, M., *et al.*: Research at DLR towards airframe noise prediction and reduction. *Aerosp. Sci. Technol.* 12(1), 80–90 (jan 2008). <https://doi.org/10.1016/j.ast.2007.10.014>
7. Ferris, R., Sacks, M., Cerizza, D., *et al.*: Aeroacoustic Computations of a Generic Low Boom Concept in Landing Configuration: Part 1 – Aerodynamic Simulations. *AIAA Aviat. Aeronaut. Forum Expo. AIAA Aviat. Forum 2021* (2021). <https://doi.org/10.2514/6.2021-2195>
8. Ffowcs Williams, J.E., Hawkings, D.L.: Sound generation by turbulence and surfaces in arbitrary motion. *Philos. Trans. R. Soc. London. Ser. A, Math. Phys. Sci.* 264(1151), 321–342 (may 1969). <https://doi.org/10.1098/rsta.1969.0031>
9. Gorobets, A.V., Duben, A.P., Kozubskaya, T.K., Rodionov, P.V.: Approaches to the Numerical Simulation of the Acoustic Field Generated by a Multi-Element Aircraft Wing in High-Lift Configuration. *Math. Model. Comput. Simulations* 15(1), 92–108 (feb 2023). <https://doi.org/10.1134/S2070048223010088>
10. Guseva, E.K., Garbaruk, A.V., Strelets, M.K.: An automatic hybrid numerical scheme for global RANS-LES approaches. *J. Phys. Conf. Ser.* 929(1), 012099 (nov 2017). <https://doi.org/10.1088/1742-6596/929/1/012099>
11. Khorrami, M.R., Shea, P.R., Winski, C.S., *et al.*: Aeroacoustic Computations of a Generic Low Boom Concept in Landing Configuration: Part 3 – Aerodynamic Validation and Noise Source Identification. *AIAA Aviat. Aeronaut. Forum Expo. AIAA Aviat. Forum 2021* (2021). <https://doi.org/10.2514/6.2021-2197>
12. Mockett, C., Fuchs, M., Garbaruk, A., *et al.*: Two Non-zonal Approaches to Accelerate RANS to LES Transition of Free Shear Layers in DES. In: *Prog. Hybrid RANS-LES Model.*, vol. 130, pp. 187–201. Springer Verlag (2015). [https://doi.org/10.1007/978-3-319-15141-0\\_15](https://doi.org/10.1007/978-3-319-15141-0_15)

13. Nicoud, F., Toda, H.B., Cabrit, O., *et al.*: Using singular values to build a subgrid-scale model for large eddy simulations. *Phys. Fluids* 23(8), 085106 (aug 2011). <https://doi.org/10.1063/1.3623274>
14. Ribeiro, A.F., Ferris, R., Khorrami, M.R.: Aeroacoustic Computations of a Generic Low Boom Concept in Landing Configuration: Part 2 – Airframe Noise Simulations. *AIAA Aviat. Aeronaut. Forum Expo. AIAA Aviat. Forum 2021* (2021). <https://doi.org/10.2514/6.2021-2196>
15. Shur, M.L., Spalart, P.R., Strelets, M.K.: Noise Prediction for Increasingly Complex Jets. Part I: Methods and Tests. *Int. J. Aeroacoustics* 4(3), 213–245 (jul 2005). <https://doi.org/10.1260/1475472054771376>
16. Shur, M.L., Spalart, P.R., Strelets, M.K.: Noise Prediction for Increasingly Complex Jets. Part II: Applications. *Int. J. Aeroacoustics* 4(3), 247–266 (jul 2005). <https://doi.org/10.1260/1475472054771385>
17. Shur, M.L., Spalart, P.R., Strelets, M.K., Travin, A.K.: An Enhanced Version of DES with Rapid Transition from RANS to LES in Separated Flows. *Flow, Turbul. Combust.* 95(4), 709–737 (jun 2015). <https://doi.org/10.1007/S10494-015-9618-0>
18. Spalart, P.R., Allmaras, S.R.: A one-equation turbulence model for aerodynamic flows. In: *AIAA Pap.* 92-0439 (1992). <https://doi.org/10.2514/6.1992-439>
19. Stabnikov, A.S., Garbaruk, A.V.: Testing of modified curvature-rotation correction for  $k-\omega$  SST model. *J. Phys. Conf. Ser.* 769(1), 012087 (nov 2016). <https://doi.org/10.1088/1742-6596/769/1/012087>
20. Sun, Y., Smith, H.: Review and prospect of supersonic business jet design. *Prog. Aerosp. Sci.* 90, 12–38 (2017). <https://doi.org/10.1016/j.paerosci.2016.12.003>
21. Voevodin, V.V., Antonov, A.S., Nikitenko, D.A., *et al.*: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. *Supercomput. Front. Innov.* 6(2), 4–11 (jun 2019). <https://doi.org/10.14529/jsfi190201>
22. van der Vorst, H.A.: Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* 13(2), 631–644 (1992). <https://doi.org/10.1137/0913035>
23. Zaporozhets, O., Tokarev, V.I., Attenborough, K.: *Aircraft Noise: Assessment, prediction and control.* Spon Press (2011), [https://books.google.com/books/about/Aircraft\\_Noise.html?hl=ru&id=wSMXnVVgOSQC](https://books.google.com/books/about/Aircraft_Noise.html?hl=ru&id=wSMXnVVgOSQC)

# Tool and Algorithm for the Determination of Aptamers in Nanopore Sequencing Data: AptaLong

*Maria A. Grigoryeva*<sup>1</sup> , *Maria G. Khrenova*<sup>1,2</sup> , *Maksim F. Subach*<sup>1</sup> ,  
*Vladimir V. Voevodin*<sup>1</sup> , *Maria I. Zvereva*<sup>1</sup> 

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

Nanopore sequencing is a third generation sequencing technology that allows direct, real-time sequencing of individual DNA or RNA molecules. It utilizes a nanopore – an extremely small pore – in a membrane to pass a single strand DNA or RNA. As the sequence passes through the nanopore, changes in electrical current are detected and used to determine the nucleotide sequence. Nanopore sequencing has several advantages. It offers long read lengths, allowing for the sequencing of difficult regions of the genome, such as repetitive regions. It also enables real-time sequencing, providing immediate data generation without the need for extensive library preparation. Many bioinformatics pipelines and tools have been developed specifically for nanopore sequencing data analysis, addressing the unique characteristics and challenges of this technology, while dealing with non-standard long reads, derived from the ligation process of shorter oligonucleotides, might be challenging. In this research we present a new algorithm that extracts an aptamer sequence from the results of nanopore sequencing of several SELEX experiments with single-stranded DNA. The algorithm is based on statistical methods, based on known primer sequences and length of searching aptamer. We used step-by-step displacement of the reference sequence with positional alignment and calculated the positional frequencies of each nucleotide. As a result, the nucleotide frequencies obtained at each step are averaged, and thus, we find the sequence that is more likely to represent the aptamer.

*Keywords: nanopore sequencing, aptamer, SELEX, primer, sequence alignment.*

## Introduction

Aptamers are a specific type of targeting ligands based on single-stranded nucleic acids that can bind to a target molecule with high affinity and specificity. Aptamers can bind to targets ranging from small molecules to complex structures such as protein complexes or cell surface. Due to these unique characteristics, as well as low immunogenicity, toxicity, ease of synthesis with minor variations, good stability, they are used for a variety of diagnostic and therapeutic applications. Aptamers are also used as molecular probes instead of antibodies [4]. After a quarter of a century of research aptamers undergo pharmacological revision as selective drug for a specific clinical need [5]. Aptamers are usually selected from synthetic nucleic acids libraries. There are different strategies to obtain aptamers, as well as approaches to design initial compound libraries based on pre-structured sequences and modified nucleotides for optimisation of properties after selection of the best sequence [7]. The process of aptamer identification known as systematic evolution of ligands by exponential enrichment (SELEX) involves numerous singular processes, each of which contributes to the success or failure of aptamer generation [3]. Usually, the library for SELEX is presented as a set of nucleic acids with different sequences, each consisting of possible aptamers sequence connected with flanking regions for amplification by PCR and primers hybridisation after every round of selection in SELEX. Every sequence in the library is a combination of A, C, G, T nucleotides. The choice of modification position to improve properties is carried out for an already selected aptamer sequence based on the structural data of the aptamer-target. Recently, the use of libraries with modified bases (an additional one is

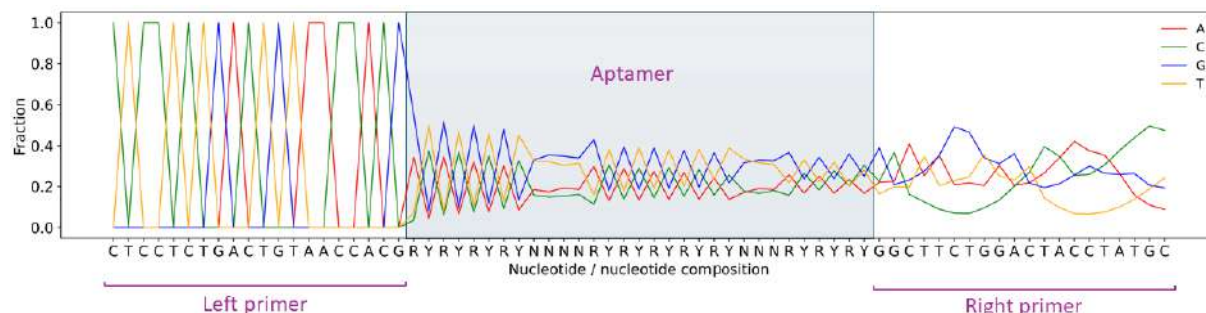
<sup>1</sup>Lomonosov Moscow State University, Moscow, Russian Federation

<sup>2</sup>Federal Research Centre “Fundamentals of Biotechnology” of the Russian Academy of Sciences

added to the four) has been proposed as methods for determining the sequence with modified bases directly have become available, in particular single-molecule nanopore sequencing [1]. The possibility of using nanopore sequencing for aptamer identification has recently been shown experimentally [2]. Since nanopore sequencing is focused on long reads, the selected sequences were combined into long reads for analysis. This required additional data analysis tools, which we offer.

By cutting out all the short sequences that include the left and right primers along with the aptamer between them, it theoretically becomes feasible to calculate the frequency of each nucleotide's occurrence at every position. However, due to the sample preparation peculiarities, attempting a global alignment of sequences based on known primers, such as starting from the left primer shown in Fig. 1, reveals that the probabilities of the remaining nucleotides are too low to reliably identify the aptamer. Global alignment only provides insight into the prevailing positional probabilities of specific types of nucleotides. For instance, in Fig. 1, the notations R, Y, and N denote positions within the aptamer, indicating:

- R = 45:05:45:05 A/C/G/T – enriched with purine bases;
- Y = 05:45:05:45 A/C/G/T – enriched with pyrimidine bases;
- N = 25:25:25:25 A/C/G/T – equally probable.

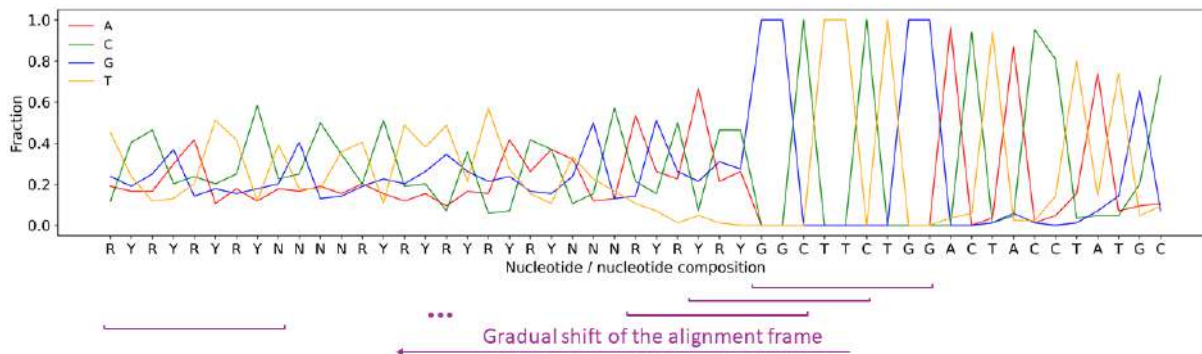


**Figure 1.** Globally aligned sequences: “Left primer – Aptamer – Right Primer”. Alignment from the left primer with the positional probabilities of nucleotide occurrences

Performing a global alignment of sequences starting from one of the primers does not enable the complete reconstruction of an aptamer in a single step. However, by selecting a shorter fragment of a primer for the alignment, a greater number of sequences can be included, providing more statistical data and increasing the likelihood of determining positional probabilities of nucleotides more effectively. Thus, we can take various fragments of a known primer, or fragments of an aptamer that were found in the immediate vicinity of the primer, in order to perform alignments step by step, clarifying previous statistical findings at each stage. By applying positional alignment to various known (or most likely known) fragments, averaging the probabilities obtained, it is possible to restore the aptamer.

The idea of the proposed algorithm is to incrementally shift some known reference fragment with sequence alignment and calculation of positional probabilities of nucleotide occurrence (see Fig. 2). At each stage, nucleotides with a high probability of occurrence will be added to the current reference fragment, thus the number of recognized nucleotides will increase. And as a result, we will only have to average the probabilities obtained in order to build an entire aptamer.

The paper is structured into five sections. Section 1 outlines current methods for aptamer detection. In Section 2, data representation is discussed. Section 3, which is further divided into five subsections, delves into the proposed algorithm, addressing data preparation, sequence extraction and alignment, description of the gradual movement of the reference sequence, pro-



**Figure 2.** Visualization of the general idea of the algorithm

cessing of collected statistics for all shifts, aggregation of results, and bifurcations. Section 4 is dedicated to statistical metrics for evaluating results. Lastly, Section 5 examines the validation of the AptaLong method.

## 1. Existing Approaches for Aptamer Search

### 1.1. Experimental Data Preparation

The preparation of SELEX samples for long-read nanopore and PacBio (sequencing of nucleic acids which involves real-time DNA replication with fluorescent nucleotide triphosphates producing long, accurate sequencing) requires the retrieval of long sequences, which is crucial for accurate characterization of aptamer candidates [6]. These methods can be widely used for detecting modified nucleic bases in aptamers. Currently there are two proposed methods for sample preparation after SELEX for the ability to analyze data using nanopore sequencing. The first method involves self-ligation, where SELEX library sequences after N rounds of selection are ligated with themselves, resulting in the formation of longer DNA molecules. This method was successfully used for obtaining aptamers for SARS-CoV-2 RBD protein with high affinity to the target [2]. Another method involves ligating the SELEX library after N rounds with a linearized plasmid vector (a circular DNA molecule that has been cut to form a linear piece). For this sequencing preparation TA cloning (a technique used to insert a piece of DNA into a plasmid vector) is used. The method takes advantage of a special feature where the DNA to be cloned has a single “A” (adenine) at each end, and the plasmid vector has a single “T” (thymine) at each end. These “A” and “T” ends naturally stick together, allowing the DNA to be easily inserted into the plasmid for replication and further use [9]. It is assumed that after N rounds of selection, the oligonucleotide pool becomes enriched with high-affinity aptamers. We can use this approach with further transformation of *E. coli* bacteria to yield a high amount of a plasmid containing most abundant aptamers for further sequencing. Moreover, ligation with a plasmid increases the length of aptamer sequences, making them suitable for nanopore sequencing [3].

### 1.2. Existing Algorithmical Methods for Aptamer Detection

There are many algorithms and tools for the analysis of nanopore sequencing, but in most cases these tools are aimed at searching for motifs. Motif commonly refers to a recurring pattern or sequence within a DNA/RNA molecule. An aptamer, on the other hand, is a specific type of sequence or molecule that can bind to a target molecule with high affinity and specificity. The

**Table 1.** Example representation of GAM for an aptamer consisting of 31 nucleotides

	0	1	2	3	4	5	6	7	8	9	10	...	26	27	28	29	30
<b>A</b>	0.45	0.05	0.45	0.05	0.45	0.05	0.45	0.05	0.25	0.25	0.25	...	0.05	0.45	0.05	0.45	0.05
<b>C</b>	0.05	0.45	0.05	0.45	0.05	0.45	0.05	0.45	0.25	0.25	0.25	...	0.45	0.05	0.45	0.05	0.45
<b>G</b>	0.45	0.05	0.45	0.05	0.45	0.05	0.45	0.05	0.25	0.25	0.25	...	0.05	0.45	0.05	0.45	0.05
<b>T</b>	0.05	0.45	0.05	0.45	0.05	0.45	0.05	0.45	0.25	0.25	0.25	...	0.45	0.05	0.45	0.05	0.45

specificity of aptamers lies in the method of their production – SELEX, which involves multiple binding and amplification operations, during which various clusters of compounds can be formed that are best bound to the target. Then, these compounds are ligated using primers, resulting in long chains of nucleotides. At the stages of amplification and ligation, various distortions of the sequences may occur, and therefore, the results can be significantly noisy, and the lengths of the sequences passing through the nanopores can also vary greatly. All this makes it much more difficult to use algorithms designed to find motifs.

One of the utilities that might be suitable is FASTAptamer<sup>3</sup>. It counts, normalizes and ranks read counts in a FASTQ file, compares populations for sequence distribution, generates clusters of sequence families, calculates fold-enrichment of sequences throughout the course of a selection and searches for degenerate sequence motifs. However, the sequences obtained from SELEX might not be in a form that is directly compatible with FASTAptamer. SELEX-derived aptamer sequences may contain modified bases, adapters, or linkers used in the SELEX process. These additional elements can make the sequences more complex and may not be recognized or handled properly by FASTAptamer, designed specifically for analyzing standard aptamer sequences. Therefore, while FASTAptamer may be useful for general analysis of aptamer sequences, it does not provide specialized functionalities for nanopore sequencing data or ligated sequences. For this reason, we decided to implement our own algorithm that takes into account the specifics of the data.

## 2. Data Representation

The initial data is represented in FASTQ files and several a priori known features.

- Left  $P_L$  and right  $P_R$  primers are known in advance, and are unique for a FASTQ file being analyzed.
- All sequences in a FASTQ file have variable lengths, but are supposed to contain left ( $P_L$ ) and right ( $P_R$ ) primers, and an aptamer  $A$  between them. In the sequencing process, primers and aptamers might be detected with errors reaching 10%. Moreover, during the ligation process, the sequences “Left Primer – Aptamer – Right Primer” might be disrupted.
- Global Alignment Matrix (GAM) with the prevailing positional probabilities of specific types of nucleotides.

## 3. Aptamer Search Method

The proposed algorithm, AptaLong, involves incrementally shifting the alignment frame while calculating the positional probabilities of each nucleotide. As the frame gradually moves,

<sup>3</sup><https://fastaptamer2.missouri.edu/>



nucleotide statistics are gathered at each shift. Upon reaching the shifting limit, these statistics are combined and summarized to reconstruct an aptamer.

The algorithm comprises multiple stages:

- data preparation;
- extraction and alignment of sequences;
- incremental movement of the reference sequence;
- consolidation of nucleotide probabilities over all reference shifts;
- traversal of bifurcations.

In the subsequent sections, we delve into a detailed explanation of each of these procedures.

### 3.1. Data Preparation

Complementary and direct forms of primers and aptamers often coexist within a FASTQ file. The analysis reveals a common occurrence of linked or “glued” complementary and direct primers. These connections, when abundant, can introduce statistical biases during sequence alignment due to shifts. In certain cases, as many as 50% of sequences in a FASTQ file may exhibit such fused primers. Hence, during the initial data preparation phase, it becomes critical to detect and segregate sequences with “glued” primers at their junctions. This step can notably increase the overall number of sequences initially.

#### Example:

Right primer ( $P_R$ ): GGCTTCTGGACTACCTATGC

Complementary right primer ( $P_{CR}$ ): GCATAGGTAGTCCAGAAGCC

Sequence with “glued”  $P_{CR}$  and  $P_R$ :

GACTGTAACACAGGATGTGTTCCCCTGTACGTTGTGCGTGTGCATAGGTAGTCCAGAAGCCGGCTTCTGGACTACCTATGCACACGAACACACTCTAACGACGCCACCGTGGTTACAGTCAGAGAGAATATACAGGCTAGAGAAGCAGTC

The resulting two sequences obtained by splitting the original sequence at the primer junction:

- GACTGTAACACAGGATGTGTTCCCCTGTACGTTGTGCGTGTGCATAGGTAGTCCAGAAGCC;
- GGCTTCTGGACTACCTATGCACACGAACACACTCTAACGACGCCACCGTGGTTACAGTCAGAGAGAATA  
TACAGGCTAGAGAAGCAGTC.

### 3.2. Sequences Extraction and Alignment

#### 3.2.1. Detection of the initial reference sequence

First of all, the initial reference sequence must be chosen. Since only primers are known in advance, the entire primer sequence could be utilized for alignment and aptamer discovery. However, given that primers might be distorted during ligation and nanopore sequencing, relying on the complete primer for alignment may not be advisable. Instead, certain primer fragments could be significantly represented in the data. Therefore, opting for a fragment from one of the primers as the starting point for the search is the most logical approach. This chosen fragment serves as the initial reference sequence and it should be linked to the sought aptamer. Experimental findings suggest that the optimal length of the reference sequence should not exceed half of the entire primer length.

In the context of optimal primer-aptamer ligation, in the sequence structure denoted as Left Primer - Aptamer - Right Primer ( $P_L - A - P_R$ ), the positioning of the reference

sequence at both ends of the aptamer is predetermined. In the provided illustration, with identified primers and the aptamer length specified, the initial reference sequence started at position  $idx$ , with the left portion of the right primer highlighted in red:

CTCCTCTGACTGTAACACG\*\*\*\*\*GGCTTCTGGACTACCTATGC  
 0----- $idx$ ----- $L_A + 2 * L_P$

Another option, is to select the initial reference as the right part of the left primer:

CTCCTCTGACTGTAACACG\*\*\*\*\*GGCTTCTGGACTACCTATGC  
 0----- $idx$ ----- $L_A + 2 * L_P$

### 3.2.2. Input data

- FastQ sequences – a list of DNA sequences of variable lengths;
- GAM, represented as a set of vectors with the prevailing probabilities of all nucleotides for a DNA library:

$$GAM = \begin{pmatrix} p(A_i)_G \\ p(C_i)_G \\ p(G_i)_G \\ p(T_i)_G \end{pmatrix};$$

- Left  $P_L$  and right  $P_R$  primers;
- $Ref_{idx} = [X]\{L_{Ref}\}$  – reference sequence – a fragment from the  $P_L$  or  $P_R$ , where  $L_{Ref}$  is the length of the reference sequence,  $X = [ACGT]$  – one of nucleotides A, C, G and T, and  $idx$  – the index of the occurrence of the reference;
- $L_P + L_A$  – the length of the fragments, where  $L_A$  – the length of an aptamer,  $L_P$  – the length of a primer;
- $2 * L_P + L_A$  – the total length of an ideally ligated sequence.

### 3.2.3. Fragments extraction and alignment by reference

At this stage, a set of fragments having equal length of  $L_P + L_A$ , aligned to the reference fragment  $Ref$  at its start position  $idx$ , are extracted from the initial sequences. If  $Ref$  belongs to the  $P_R$ , then the extracted sequence  $S_i$  can be represented as a potential aptamer, followed by the reference and the remaining part of the right primer:

$$S_i = [X]\{L_A\}[X]\{L_{Ref}\}[X]\{L_P - L_{Ref}\}.$$

If  $Ref$  belongs to the  $P_L$ , then the sequence consists of the initial part of the left primer, the reference and an aptamer:

$$S_i = [X]\{L_P - L_{Ref}\}[X]\{L_{Ref}\}[X]\{L_A\}.$$

For instance, shown below is a sequence comprised of two segments with the reference  $Ref = GGCTTCTGG$  beginning from the 31st position and  $P_R = GGCTTCTGGACTACCTATGC$ . The potential 31-base aptamers are marked in red, the reference segment in blue, and the remaining part of the potential right primer in gray:

ACCCTCCTCGGCTGCTTGGTCGCGGGCGGGTGTGGA**TACACTGGAGGTGCGCATAGGTGGTCAAGCCGGCTTCTGG**ACT  
 ACCTATGCAGACATCAAACGTACAGGTACCCATGCATCGTGGTTACAGTCAGAGAAGCTTCTGGACTTTACTATGCATA  
 TACAAATGTAAAGAGAGAAATTGCTT**TACACAACGTGGATTTACCAGTCAGAGGAGGCTTCTGG**ACTGCCTATTAGCAC

Two fragments can be found and aligned by the position of the reference GGCTTCTGG:

- TACTACTGGAGGTGCGCATAGGTGGTCAAGCCGGCTTCTGGACTACCTATGC;
- TTTACACAACGTGGATTTACCAGTCAGAGGAGGCTTCTGGACTGCCTATTA.

Consequently, a series of aligned fragments are obtained from the FASTQ file. These fragments can be presented in a tabular form (Tab. 2), where the columns represent position numbers and the rows signify the sequential numbers of the fragments.

The extracted sequences indicate considerable diversity among the aptamers, even when they originate from the same extended sequence, and the residual portions of the right primer also display variations.

A set of the extracted and aligned sequences can be represented as:  $Sequences = \langle S_1, S_2, S_3, \dots, S_N \rangle$ , where  $N$  is the number of extracted sequences.

**Table 2.** Table with the results of the initial alignment: columns are the numbers of positions, rows – numbers of extracted fragments. Position of the right primer is highlighted as gray. At the beginning of each extracted sequence there is an aptamer of  $L_A$  length, then there is a reference sequence and the remaining part of primer  $P_R$ . And the total length of the extracted sequences is  $L_A + L_P$

	0	1	2	3	4	5	...	LA	idx	idx	idx	idx	idx	...	LA
										+	+	+	+		+
										1	2	3	4		LP
0	T	G	G	T	T	A	...	...	G	G	C	T	T	...	G
1	A	G	A	C	A	T	...	...	G	G	C	T	T	...	G
2	A	G	T	G	C	T	...	...	G	G	C	T	T	...	G
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
N	C	T	G	G	G	T	...	...	G	G	C	T	T	...	G

**Filtration of the extracted sequences.** Another vital stage in the alignment process is the exclusion of sequences with incorrect nucleotides at primer positions. This becomes crucial, especially as the alignment extends beyond the primer region. If the reference sequence strays too far from the primer, there might be fragments with entirely different sequences at the primer positions. Such sequences can arise from amplification process nuances during SELEX and distortions in sequences during ligation. Misalignment of primer positions with the reference sequence can result in missing or distorted primers, complicating the confirmation process. It is recommended to remove these sequences from the analysis to maintain the accuracy of statistical evaluations.

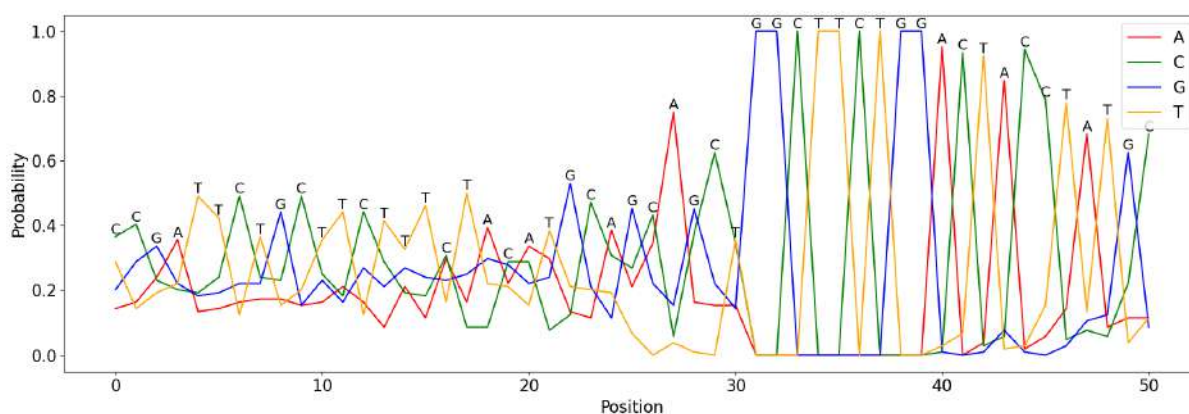
Various methods can be employed to measure the similarity between a primer and its corresponding sequence fragment. In our study, we utilized the Levenshtein distance algorithm for this purpose. The similarity is computed for each sequence within a shift and subsequently averaged.

### 3.2.4. Position-specific probability of nucleotide occurrences

Positional occurrence probabilities are calculated for each nucleotide from the collected fragments, as shown in Tab. 3. Subsequently, Fig. 3 illustrates the graphical representation of the distribution of occurrence probabilities for all nucleotides at each position.

**Table 3.** Positional probabilities of nucleotide occurrences. At the initial alignment the highest probability is at the region of the primer, and it decreases as moving further from the reference

	0	1	2	3	4	5	...	LA	idx	idx	idx	idx	idx	...	LA
										+	+	+	+		+
										1	2	3	4		LP
<b>A</b>	0.14	0.16	0.24	0.35	0.13	0.14	...	...	0	0	0	0	0	...	0
<b>C</b>	0.36	0.40	0.23	0.20	0.19	0.24	...	...	0	0	1	0	0	...	0
<b>G</b>	0.20	0.28	0.33	0.22	0.18	0.19	...	...	1	1	0	0	0	...	1
<b>T</b>	0.22	0.14	0.19	0.22	0.49	0.42	...	...	0	0	0	1	1	...	0
<b>C</b>	<b>C</b>	<b>G</b>	<b>A</b>	<b>T</b>	<b>T</b>	...	...	<b>G</b>	<b>G</b>	<b>C</b>	<b>T</b>	<b>T</b>	...	<b>G</b>	



**Figure 3.** Graphical representation of the distribution of positional probabilities of nucleotide occurrences for the alignment by the reference sequence GGCTTCTGG

At each position, the probabilities for A, C, G, and T are determined by the ratio of the specific nucleotide count at that position among all sequences to the total number of the aligned sequence fragments, calculated as:

$$p(X_i) = \frac{N_{X_i}}{N_i} .$$

Here,  $N_{X_i}$  represents the number of nucleotides X (A, C, G, or T) at position  $i$  across all  $N$  sequences, and  $N_i$  – number of all nucleotides at position  $i$ .

Consequently, for each position  $i$ , a vector consisting of four probabilities is generated:

$$p_i = \begin{pmatrix} p(A_i) \\ p(C_i) \\ p(G_i) \\ p(T_i) \end{pmatrix} .$$

The whole representation of probabilities is a set of such vectors for each position:  $Probabilities = \langle p_i \rangle$ , where  $i = 0$  to  $L_A + L_P$ .

The result sequence for the current reference alignment would be a set of nucleotides with the maximum positional probabilities.

### 3.3. Gradual Movement of Reference Sequence

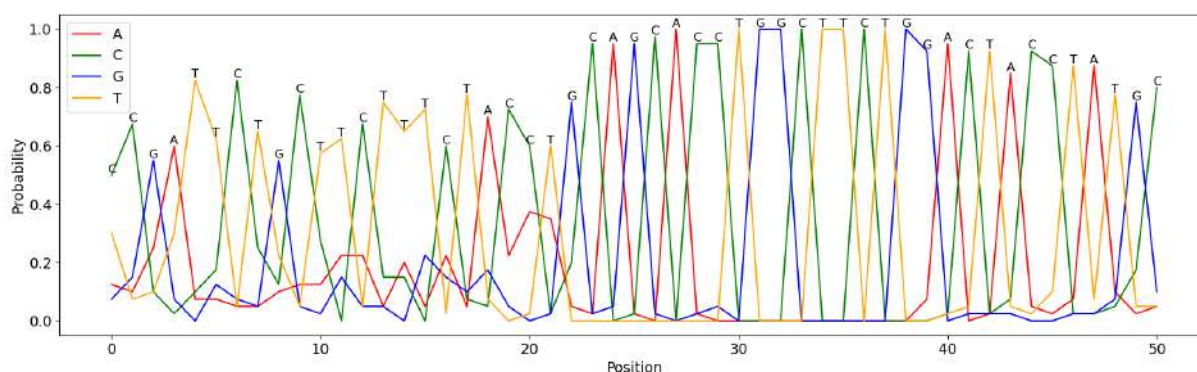
The proposed method implies a gradual movement of a reference sequence in both directions with the evaluation of the positional probabilities of occurrence for each nucleotide. The length of the reference sequence remains the same at each shift, but the combination of nucleotides might be different and is determined along the way depending on the nucleotide statistics. And the optimal choice of the reference for the next shift is the main challenge in this algorithm.

The shift can be uniquely identified by the combination of nucleotides,  $Ref$ , and its position within an ideally ligated primer and aptamer:  $idx: RefID = \{Ref, idx\}$ . At each shift  $N$  sequence fragments are extracted in accordance with the position of the reference and a length equal to primer plus aptamer (a set of sequences ( $Sequences_{RefID}$ )) and a matrix with nucleotides probabilities ( $Probabilities_{RefID}$ ) are calculated and saved for further processing.

#### 3.3.1. Reference sequence offset

After the distributions of positional probabilities for the initial reference alignment have been obtained at step 3.2.4, the reference sequence must be shifted to the left or to the right.

To determine how many positions  $k$  to move, it is necessary to estimate the maximum probabilities of nucleotides in the immediate vicinity of the current reference sequence. If there are nucleotides with a high probability (for example, more than 85%) of occurrence near the current reference, then this reference sequence shifts to the last highly probable nucleotide. It is illustrated in Fig. 4. If there is not a single nucleotide with a high positional probability in



**Figure 4.** Graphical representation of the distribution of positional probabilities of nucleotide occurrences for the alignment by the reference TGGCTTCTG with 7 highly probable nucleotides to the left

the immediate vicinity, then the displacement occurs by one step (as it is shown in Fig. 3 in Section 3.2.4).

With each subsequent shift, the next nucleotide in the reference sequence is not known in advance. In order to determine the most optimal nucleotide for the next displacement, it is necessary to perform some additional steps to evaluate the following characteristics for each variant of the reference sequence.

Thus, for each shift there might be four possible options, in accordance with the number of nucleotides (A, C, G and T). If the initial reference sequence, represented as a regular expression, is  $Ref_{idx} = (X)\{L_{Ref}\}$ , then the options shifted by one nucleotide to the left will be  $Ref_{idx-x} =$

$(X)(X)\{L_{Ref} - 1\}$ , where  $X = A|C|G|T$ . And if the movement in another direction, to the right, then the options will be:  $Ref_{idx+x} = (X)\{L_{Ref} - 1\}(X)$ .

For each of these options, the actions outlined in sections 3.2.3 and 3.2.4 are executed. Subsequently, based on the outcomes, the following metrics are calculated for each variation of the subsequent reference:

- $N_{RefID}$  – the number of sequences, extracted and aligned with a reference  $RefID$ ;
- $Sim_P$  – average similarity of  $P_L$  or  $P_R$  with the corresponding fragments of the extracted sequences:  $S[L_A : ]$  for comparison with the right primer, and  $S[: L_P]$  – for the left. The similarity can be measured as a number between 0 and 1;
- $N_{hits} = N_{RefID} * Sim_P$  – the number of hits as to the product of the number of sequences and similarity with primer.

Finally, the option with the highest value of  $N_{hits}$  is chosen as a candidate for the next shift.

The reference sequence displacement to the left or to the right is repeated until the index of the reference sequence reaches its limit. At each shift the following data is obtained:

- $Sequences_{RefID}$  – a set of sequences aligned by the current reference  $Ref$  at  $idx$  index of the start position in a sequence;
- $Probabilities_{RefID}$  – position-based probability of occurrences of each nucleotide in  $RefID$ ;
- $Bifurcations$  – an array of equally (or almost equally) probable reference sequences detected as a candidates for the next shifts.
  - If several sequences with close values of  $N_{hits}$  are found, and candidates for the next shift are equally likely, one of them is selected (with the maximum value of  $N_{hits}$ ), and the others are written to the array of bifurcations. In the next pass, an aptamer search will be started from the bifurcation.

### 3.4. Processing of the Collected Statistics for All Shifts

The data obtained at each shift can be represented as a list of sequences and positional probabilities of nucleotides:  $Shift_{RefID} = \langle Sequences_{RefID}, Probabilities_{RefID} \rangle$ .

And the final stage of the algorithm is the aggregation of data obtained at all shifts. Therefore, for each nucleotide its total positional representation is produced based on the  $Probabilities$  from all shifts.

The probability representation for nucleotide X can be expressed as a vector of probabilities of this nucleotide at each position for all shifts:

$$p(X) = \left\langle \begin{pmatrix} p(X_{1_1}) \\ p(X_{1_2}) \\ \dots \\ p(X_{1_{N_{shifts}}}) \end{pmatrix}, \begin{pmatrix} p(X_{2_1}) \\ p(X_{2_2}) \\ \dots \\ p(X_{2_{N_{shifts}}}) \end{pmatrix}, \dots, \begin{pmatrix} p(X_{(L_A+LP)_1}) \\ p(X_{(L_A+LP)_2}) \\ \dots \\ p(X_{(L_A+LP)_{N_{shifts}}}) \end{pmatrix} \right\rangle = \langle p(X_{i_j}) \rangle,$$

where  $i$  is the index of the nucleotide in a sequence,  $j$  is the index of the offset or shift.

The overall representation of probabilities for all nucleotides is:

$$p = \begin{pmatrix} p(A) \\ p(C) \\ p(G) \\ p(T) \end{pmatrix}.$$

After calculating the positional probabilities for all nucleotides, these values can be aggregated to derive the sequence encrypted by these probabilities. Thus, for a nucleotide X the average probability at each position is calculated as:  $\overline{p(X_i)}$ , where  $i$  is an index of the nucleotide in a sequence.

For each nucleotide, a sequence of positional probabilities can be calculated by averaging the values of all positional probabilities across all shifts, resulting in a probability representation of an aptamer:

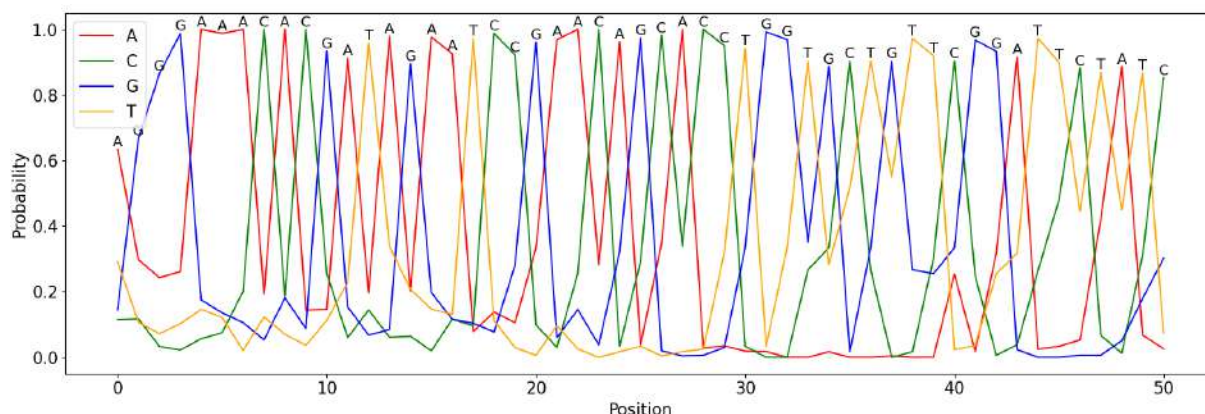
$$\overline{p(X)} = \left\langle \overline{\sum_{j=1}^{N_{shifts}} p(X_1)}, \overline{\sum_{j=1}^{N_{shifts}} p(X_2)}, \dots, \overline{\sum_{j=1}^{N_{shifts}} p(X_{L_A+L_P})} \right\rangle .$$

Finally, to obtain an aptamer sequence, the nucleotides corresponding to the highest probability are chosen at each position:

$$Aptamer = \left\langle \max_i \begin{pmatrix} \overline{p(A)} \\ \overline{p(C)} \\ \overline{p(G)} \\ \overline{p(T)} \end{pmatrix} \right\rangle ,$$

where  $i = 0$  to  $L_A + L_P$ .

Figure 5 demonstrates an example of the final distribution of positional probabilities of all nucleotides.



**Figure 5.** Aggregated table with positional probabilities of occurrence of each nucleotide: the first 31 nucleotides represent an aptamer, and remaining – right primer

### 3.5. Bifurcations

In the process of stepwise displacement of the reference sequence, bifurcations may appear at various stages, namely, nucleotides with similar values of positional probabilities of occurrence. In this case, the algorithm selects the most likely nucleotide, and writes the one closest to it to the list of bifurcations.

The table below (Tab. 4) shows a fragment of sequences aligned with the reference belonging to the right primer starting at position 31. It is necessary to select the nucleotide for the next shift one step to the left, that is, to position 30. C and T nucleotides in this position are almost equally likely (34 and 35%, respectively). Consequently, the algorithm will opt for T as

the nucleotide for the next shift, while recording C in the bifurcation list. Therefore, starting from the original reference GGCTTCTGG, applying an offset of one results in TGGCTTCTG. Given the new starting position with C, CGGCTTCTG, this change is noted at position 30 as a bifurcation.

**Table 4.** Nucleotides probabilities with a bifurcation: nucleotides C and T are equally probable at position 30

		25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
<b>A</b>	...	0.21	0.34	0.75	0.16	0.15	0.15	0	0	0	0	0	0	0	0	0
<b>C</b>	...	0.26	0.43	0.05	0.37	0.62	0.34	0	0	1	0	0	1	0	0	0
<b>G</b>	...	0.45	0.22	0.15	0.45	0.22	0.14	1	1	0	0	0	0	0	1	1
<b>T</b>	...	0.06	0	0.03	0.00	0	0.35	0	0	0	1	1	0	1	0	0
	...	<b>G</b>	<b>C</b>	<b>A</b>	<b>G</b>	<b>C</b>	<b>C</b> or <b>T</b>	<b>G</b>	<b>G</b>	<b>C</b>	<b>T</b>	<b>T</b>	<b>C</b>	<b>T</b>	<b>G</b>	<b>G</b>

Thus, the array of bifurcations consists of tuples, representing the index of the occurrence of the reference and the reference itself:

$$Bifurcations = \langle (Idx, Ref) \rangle .$$

The concept of preserving these bifurcations involves repeating all stages of aptamer search multiple times, starting from each identified bifurcation as an initial reference sequence. To prevent looping, the maximum number of bifurcations can be limited by the user.

#### 4. Statistical Metrics for Evaluation of the Results

To ensure that the obtained aptamer is statistically significant and to evaluate how well it aligns with the GAM and its corresponding primer position, several metrics are calculated for the output of each bifurcation.

1. Average number of sequences at all shifts:  $\overline{N_s} = \overline{\sum_{j=1}^{N_{shifts}} N_{RefID}}$  .
2. Overall probability of the determined aptamer:

$$\overline{p(Z)} = \overline{\sum_{i=1}^{L_P+L_A} \max(p(X)_i)} .$$

3. Similarity with the GAM:

First of all, the maximum probability of the GAM is calculated:

$$\max(GM) = \overline{\sum_{i=1}^{L_P+L_A} \max(p(X_i)_G)} .$$

Then, the average position probability of each statistically found nucleotide in accordance with the GAM is determined:

$$\overline{Z_{GM}} = \overline{\sum_{i=1}^{L_P+L_A} p(X_i|Z_i)_G} ,$$



where  $Z$  represents the sequence referred to the detected aptamer.

Finally, the similarity of this aptamer with the global alignment is calculated as:  $Z_{GM}/\max(GM)$ .

4. Similarity with the primer – calculated as the percentage of the similarity between the sequence that was obtained in the searching process and the primer.

## 5. Validation of the Algorithm

Validation of AptaLong was carried out based on the research aimed to identify and characterize a novel DNA aptamer, named MEZ, that binds to the receptor-binding domain (RBD) of the SARS-CoV-2 spike protein. Key steps in the research included the generation of aptamers through the SELEX method, specifically targeting the RBD of the SARS-CoV-2 spike protein from the Wuhan-Hu-1 strain. Aptamer sequences were identified with the novel methodology based on nanopore sequencing, described in this paper [2].

MEZ, the best candidate aptamer detected by the developed algorithms, was chemically synthesized and tested for its binding affinity to the SARS-CoV-2 Spike RBD domain from different strains. The research found that MEZ had a comparable binding affinity to known aptamers, along with a shorter length of only 31 nucleotides. Experimental data and computational simulations showed that the 3'-end of the aptamer plays a crucial role in binding to the SARS-CoV-2 spike protein and strain identification.

## Conclusion

The developed algorithm, AptaLong, facilitates the exploration of aptamers within custom sequences derived from the ligated outcomes of SELEX experiments. The algorithm functions by initially selecting a known segment of the sequence (referred to as the reference fragment) and conducting multiple alignments based on the predetermined position of this reference. Alignments proceed through the stepwise shifting of the reference in both left and right directions. Subsequently, the positional probabilities of all nucleotides are computed across all shifts, ultimately leading to aptamer identification.

This tool enables the analysis of a variety of FASTQ files containing diverse types of aptamers, provided they share identical primers. Through the utilization of bifurcation in the search process, diverse aptamers can be effectively identified. The results are presented visually through graphical representations showcasing nucleotide probabilities, along with detailed information in Excel files for each shift and bifurcation stage, ensuring straightforward interpretation of the results.

The AptaLong tool can also be utilized for aptamer sequence determination from the data obtained on highthroughput variant of the nanopore sequencer, PrometION. In this case, a set of 96 samples, or even more, can be processed simultaneously and rapidly. Such extensive data analysis demands access to supercomputing facilities. Future enhancements to the AptaLong will incorporate a parallel implementation strategy to further optimize and scale the tool for efficient processing of large volumes of data.

## Acknowledgements





This work was supported by the project of the Interdisciplinary Scientific and Educational School of Moscow State University “Molecular technologies of living systems and synthetic biology” (№ 23-Sh04-45). The study was performed using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [8].

*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. Berkovich, A., Pyshkina, O., Zorina, A., *et al.*: Direct determination of the structure of single biopolymer molecules using nanopore sequencing. *Biochemistry (Moscow)* 89, S234–S248 (2024). <https://doi.org/10.1134/S000629792414013X>
2. Khrenova, M., Nikiforova, L.A., Grabovenko, F., *et al.*: Highly specific aptamer for SARS-CoV-2 Spike protein from the authentic strain. *Org. Biomol. Chem.* 22, 5936–5947 (2024). <https://doi.org/10.1039/D40B00645C>
3. Kohlberger, M., Gadermaier, G.: SELEX: Critical factors and optimization strategies for successful aptamer selection. *Biotechnol. Appl. Biochem.* 69(5), 1771–1792 (2022). <https://doi.org/10.1002/bab.2244>
4. Kumar Kulabhusan, P., Hussain, B., Yüce, M.: Current perspectives on aptamers as diagnostic tools and therapeutic agents. *Pharmaceutics* 12(7), 646 (2020). <https://doi.org/10.3390/pharmaceutics12070646>
5. Nimjee, S.M., White, R.R., Becker, R.C., *et al.*: Aptamers as therapeutics. *Annu. Rev. Pharmacol. Toxicol.* 57, 61–79 (2017). <https://doi.org/https://doi.org/10.1146/annurev-pharmtox-010716-104558>
6. Rhoads, A., Au, K.F.: PacBio Sequencing and its Applications. *Genomics Proteomics Bioinformatics* 13(5), 278–289 (2015). <https://doi.org/10.1016/j.gpb.2015.08.002>
7. Subach, M.F., Khrenova, M.G., Zvereva, M.I.: Modern methods of aptamer chemical modification and principles of aptamer library selection. *Moscow Univ. Chem. Bull.* 79, 79–85 (2024). <https://doi.org/10.3103/S002713142470010X>
8. Voevodin, V.V., Antonov, A.S., Nikitenko, D.A., Shvets, P.A., Sobolev, S.I., Sidorov, I.Y., Stefanov, K.S., Voevodin, V.V., Zhumatiy, S.A.: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. *Supercomput. Front. Innov.* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
9. Zhou, M.Y., Gomez-Sanchez, C.E.: Universal ta cloning. *Curr. Issues Mol. Biol.* 2(1), 1–7 (2000). <https://doi.org/10.21775/cimb.002.001>

# Modeling Microtubule Dynamics on Lomonosov-2 Supercomputer of Moscow State University: from Atomistic to Cellular Scale Simulations

*Nikita B. Gudimchuk*<sup>1,2,3</sup> , *Veronika V. Alexandrova*<sup>1</sup>,  
*Evgeniy V. Ulyanov*<sup>1</sup>, *Vladimir A. Fedorov*<sup>4</sup> , *Ekaterina G. Kholina*<sup>4</sup> ,  
*Ilya B. Kovalenko*<sup>4</sup> 

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

Cytoskeletal polymers of tubulin, the microtubules, are critically important for cellular physiology. Their remarkable non-equilibrium dynamics and unusual mechanical properties have nurtured interest in exploring microtubules with diverse experimental methods and modeling their properties at different scales. In this work, we overview the studies of microtubules from the atomistic level of detail to the cellular dimension, focusing on the computational modeling work that has been carried out by our group on Lomonosov-2 supercomputer of Moscow State University since 2015. Our computational efforts have been aimed at understanding of microtubules through a set of models at multiple spatial and temporal scales, starting from examining the properties of tubulin dimers, as the building blocks, and further elucidating how those properties enable more complex assembly/disassembly and force-generation behaviors of microtubules, emerging at larger scales. Our methodology includes different approaches, from atomistic molecular dynamics to more coarse-grained techniques, such as Brownian dynamics and Monte Carlo simulations. We describe the motivation and the context for each model, overview the major conclusions from the simulations, which we believe were instrumental in building an integrative understanding of these polymers. We also discuss some technical aspects of the modeling, such as the computational performance of different types of simulations, current limitations and potential future directions for description of the microtubule dynamics, using the multi-scale approach.

*Keywords:* microtubule, Lomonosov-2, computational performance, multi-scale simulations, molecular dynamics, Brownian dynamics, kinetic Monte Carlo simulations.

## Introduction

Microtubules are essential biopolymers that form a core component of a system of cytoskeletal fibers in eukaryotic cells, alongside actin and intermediate filaments [9]. Tubulin monomers, the building blocks of microtubules, are globular proteins approximately 4 nm in diameter. Alpha and beta tubulins, associate into heterodimers, which then polymerize into hollow cylindrical microtubules with a helical lattice, typically composed of thirteen protofilaments. Due to their high flexural rigidity, microtubules can span entire cell, sometimes reaching lengths of tens of micrometers. A remarkable feature of microtubules is their dynamic assembly/disassembly behavior. These polymers undergo extended phases of elongation and shortening [26], with stochastic transitions between these phases termed catastrophes and rescues. This energy-consuming process allows microtubules to continuously explore the interior of the cells and rebuild their network. Intriguingly, they can even turn the energy of their growth and shortening into useful work, producing significant forces within cells (reviewed in [19]). These multi-tasking polymers are involved in numerous critical processes, including intracellular transport, maintenance of cell

---

<sup>1</sup>Department of Physics, Lomonosov Moscow State University, Moscow, Russia

<sup>2</sup>Center for Theoretical Problems of Physicochemical Pharmacology, Moscow, Russia

<sup>3</sup>Pskov State University, Pskov, Russia

<sup>4</sup>Department of Biology, Lomonosov Moscow State University, Moscow, Russia

shape, building the cell division apparatus for correct segregation of chromosomes, transporting chromosomes and remodeling membranes, etc. [20].

To explain the mechanism underlying the transitions of microtubules between elongation and shortening, the idea of the guanosine triphosphate (GTP) cap has been put forward about four decades ago [2], and further refined in the late 1980s and 1990s, into a widely accepted model. The key postulates of this model can be formulated as follows:

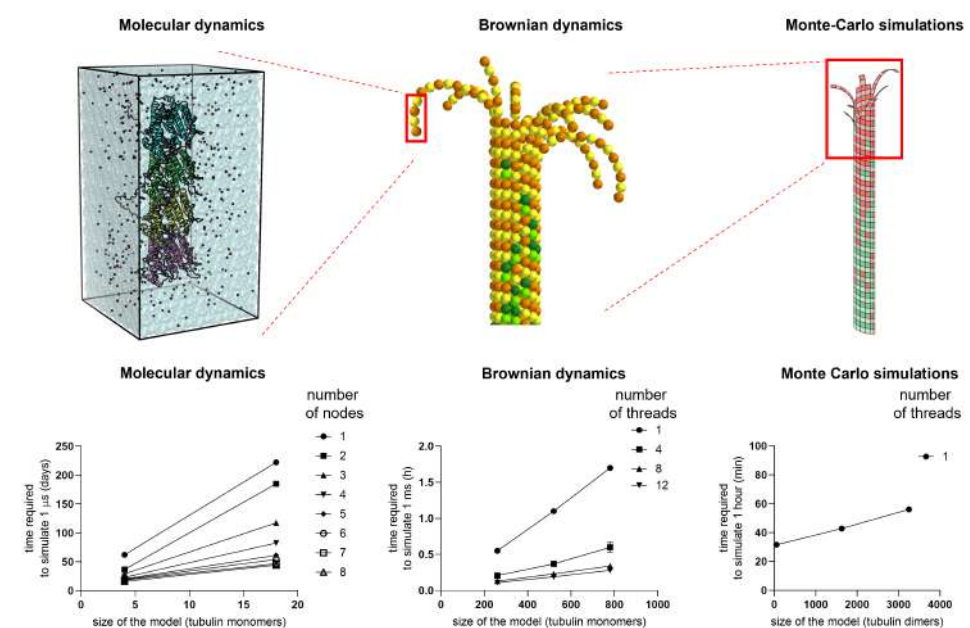
1. During microtubule assembly, tubulin dimers incorporate into the microtubule end from the solution in the GTP-bound form.
2. GTP-bound tubulins exhibit a relatively ‘straight’ equilibrium curvature in the dimeric and oligomeric form, allowing them to incorporate into the microtubule without significant strain.
3. After incorporation into the microtubule lattice, GTP hydrolysis is catalyzed, converting tubulin dimers to the GDP-bound form. Thus, the majority of tubulins in the microtubule lattice are GDP-bound, while only the freshly polymerized layers at the growing microtubule end are GTP-bound. These layers are termed the ‘GTP cap’.
4. The GTP cap protects the growing microtubule from depolymerization. The stochastic loss of the GTP cap from the microtubule tip triggers a catastrophe – a transition from growth to rapid shortening.
5. GDP-bound tubulins have ‘curved’ conformations, resulting in strain within the microtubule lattice, and accumulating the energy of mechanical deformation. During microtubule depolymerization, this energy is released as the protofilaments lose their lateral bonds and ‘unzip’ from the tip.
6. Stochastic regain of the GTP cap enables the microtubule rescue – a transition from depolymerization to growth.

Together, these six postulates summarize an effective conceptual model for the dynamic instability of microtubules, successfully describing key aspects of microtubule behavior. However, like any model, it simplifies reality. Recent observations on microtubule dynamics and structure have revealed inconsistencies, both quantitative and qualitative, with some predictions of this model (reviewed in [8, 20]). Although seemingly subtle, these new observations reveal important details, which are essential for eliciting the mechanisms of microtubule control through different factors, including: (i) associated regulatory proteins, (ii) scaffolds that couple microtubule ends to various structures in order to transmit mechanical forces, (iii) tubulin targeting drugs, representing a major type of anticancer chemotherapeutics, and (iv) tubulin modifications, such as point mutations, or post-translational changes.

Here we describe our use of the multi-scale modeling, as a powerful approach to integrate the complexity of accumulating new multi-faceted and sometimes contradictory data about microtubules into a comprehensive framework, necessitating revisions to the traditional GTP cap model. The systems simulated in this study span three orders of magnitude on the spatial scale, from tens of nanometers to micrometers, and over eight orders of magnitude on the temporal scale, from nanoseconds to hours (Fig. 1). Typical particle numbers and temporal and spatial scales for all models are summarized in Tab. 1. The simulations were conducted using the high-performance computing resources of the Moscow State University, primarily the Lomonosov-2 supercomputer [34].

The article is organized as follows. Section 1.1 is devoted to molecular dynamics simulations of tubulin dimers and oligomers. Section 1.2 describes Brownian dynamics simulations of entire

microtubule tips. Section 1.3 reviews Monte Carlo simulations of microtubule dynamic instability at the scale of tens of micrometers in length. The final section of this study summarizes the findings and suggests directions for future research.



**Figure 1.** Multi-scale models of microtubules and computational performance of the simulations on Lomonosov-2 supercomputer of the Moscow State University. The upper row of images provides schematics of molecular models at different spatiotemporal scales. The graphs below show the performance of the simulations on Lomonosov-2 (CPU: 2Intel Xeon E5-2697, GPU: Nvidia Tesla K40) as a function of the size of the modeled system and the number of nodes/computing threads. The analysis of MD simulation performance is based on data from [13]

**Table 1.** Typical spatiotemporal scales covered in each type of simulation

Model type	Typical model size (tubulin subunits)	Typical model size (number of particles)	Spatial scale	Temporal scale
Molecular dynamics (MD)	2–18 monomers	200,000–1,100,000 atoms	~10 nm	~1 $\mu$ s
Brownian dynamics (BD)	100–200 monomers	100–200 tubulin monomers	~100 nm	~1 s
Monte Carlo (MC) simulations	10–3000 dimers	100–3000 tubulin dimers	~10 $\mu$ m	~1 h

## 1. Results and Discussion

### 1.1. Molecular Dynamics Simulations of Tubulin Dimers and Oligomers

Recent structural and biochemical findings about tubulins and microtubules have challenged some of the postulates of the classical GTP cap model (reviewed in [11, 20]). Specifically, several studies have reported that GTP-bound tubulins in solution exhibit curvature similar to that of GDP-bound tubulins [2, 6, 27, 29]. In addition, a growing body of literature (reviewed in [23, 32]) has suggested that microtubule dynamics in cells are affected by posttranslational modifications

of the non-globular, intrinsically disordered tails – polypeptide regions that are neglected by the classical model.

These observations motivated our investigation of individual tubulin dimers and oligomers at the nanoscale using atomistic molecular dynamics (MD). We aimed to address two questions: (1) Is GTP-bound tubulin significantly less curved than GDP-bound tubulin? (2) Are the unstructured charged tails of tubulins involved in the microtubule assembly/disassembly process? Atomistic MD is a computational technique that represents individual atoms as particles, described by Newtonian equations of motion [31]. The forces acting on these particles are derived from a force-field – a set of pre-calibrated parameters describing bonded and non-bonded interactions between any set of atoms. Due to high-frequency bond-angle vibrations involving hydrogen atoms, the positions and velocities of atoms in protein simulations need to be updated with very short timesteps, typically about 1–2 femtoseconds (fs) or slightly longer [16]. The MD method is widely used in numerous laboratories worldwide to simulate dynamics in various molecular systems, including proteins, at the resolution of individual atoms and on timescales up to tens of microseconds. Several major software packages have been developed for efficient MD simulations. GROMACS is one of the leading packages, optimized for computations using hybrid architectures and employing multi-level parallelism [1, 28].

We created several molecular models, including (i) models of tubulin dimers, as the smallest building blocks of microtubules, (ii) models of tubulin tetramers, as the shortest possible oligomers, and (iii) models of tubulin octadecamers, representing small fragments of the microtubule wall (three laterally connected tubulin hexamers). All modeled systems included the unstructured charged regions of beta and alpha tubulins. The tubulins were solvated in water and charge-neutralized with  $K^+$  and  $Cl^-$  ions (Fig. 1).

Analysis of the MD simulation trajectories, collectively spanning approximately 30 microseconds, revealed that the overall curvatures of tubulin dimers and small oligomers were very similar in both the GTP and GDP states [15]. This finding contrasts with the postulate of the classical model but aligns with recent structural and biochemical data. Thus, it is not the curvature of tubulin that is modulated by the nucleotide to switch its properties from polymerization-competent to polymerization-incompetent. Rather, we found that the nucleotide likely alters the flexibility of the tubulin interface between tubulin dimers, making GDP-bound protofilaments softer and, therefore, easier to straighten and incorporate into the microtubule lattice.

Examination of the behavior of the disordered tails in the simulations has revealed the potential for direct interaction between the negatively charged amino acid residues of the alpha-tubulin tail and the positively charged amino acid residues on the longitudinal polymerization interface of the tubulin dimer [7]. This suggests that the tail could occlude the polymerization interface, thereby downregulating the rate of incorporation of tubulin dimers into the growing microtubule tip. This modeling result is in good agreement with experimental observations of faster microtubule growth in tubulin mutants with deleted alpha-tubulin tails [7]. Together, these theoretical and experimental data identify a direct role of unstructured charged regions of tubulin in the modulation of microtubule assembly, opening new possibilities for controlling microtubule dynamics by modifying the interactions of the tails with tubulins.

A typical MD simulation trajectory was 1 microsecond long and required approximately 10–30 days of computer time on 8 nodes of the Lomonosov-2 supercomputer, depending on the model system's size (Fig. 1). The computational performance of MD simulations of tubulins and other molecular systems has been extensively explored in a series of dedicated publications

in this journal. That analysis included assessments of the performance of MD simulations on various computational architectures and offered practical suggestions [12–14].

## 1.2. Brownian Dynamics Simulations of Microtubule Tips

Due to the high computational complexity of MD simulations, it has not been possible to simulate whole microtubule tips at the necessary timescale of at least about a second, which is required to describe microtubule assembly, disassembly, and force generation. To address this limitation, we employed the Brownian dynamics (BD) approach [21, 25, 33, 36]. In our super-coarse-grained model, each tubulin monomer was represented as a hard sphere with four interaction centers on its surface. The interactions between these centers were described by empirical energy potentials, comprising a potential well and an activation energy barrier. Quadratic energy potentials were used to describe the bending of tubulin protofilaments. The model consisted of two layers. In the fast (‘dynamic’) layer, the updated positions of each tubulin monomer were calculated using the Ermak–McCammon algorithm for solving the overdamped Langevin equation, with a computational step of 50–100 ps [10]. In the slow (‘kinetic’) layer, new tubulins were stochastically added to the tip of the microtubule with a certain probability once per millisecond, ensuring microtubule elongation without explicitly considering the arrival of tubulin dimers from the solution. The slow layer also enabled the implementation of GTP hydrolysis as the key event switching the properties of tubulin dimers, eventually triggering an abrupt transition from assembly to disassembly. These transitions could be explored at the timescale of about a second, which was feasible with this type of modeling (Fig. 1).

The main questions we addressed with the BD model were: (i) What is the mechanism of microtubule elongation, given the curved shape of GTP-tubulin dimers and oligomers? (ii) How much force can be generated by the microtubule tip, and how does this force depend on parameters of tubulin-tubulin interactions?

The first question was partially motivated by computational and experimental data described in the previous section, including our own MD simulations. Additionally, structural findings using cryo-electron tomography from our group and others provided new evidence for the presence of curved protofilaments at the growing microtubule ends under a wide range of conditions, in various species, and in purified *in vitro* systems [21, 22, 24, 25]. The second question pertained to the ongoing debate on the mechanisms of force production by microtubules and the coupling between shortening microtubule tips and chromosome-associated kinetochore proteins – one of the longstanding problems in the biology of mitosis [5, 18, 30, 35].

The modeling offered several insights that allowed us to conceptualize a revised model of microtubule assembly. Central to this model is the computational observation that the thermal fluctuations of curved tubulin protofilaments are frequent enough to permit the straightening of protofilaments necessary for establishing lateral contacts. This reconciles the seemingly inconsistent properties of tubulin protofilaments: their curved shape and their ability to elongate microtubules. According to the model’s prediction, even relatively stiff protofilaments are expected to fluctuate sufficiently to form lateral bonds [21]. Protofilament stiffness and the activation barriers for lateral tubulin-tubulin interactions were predicted to be the two key factors responsible for the efficient conversion of the free energy of GTP hydrolysis into mechanical work and the ability of curling protofilaments to exert forces on cargoes [19, 21].

Our BD model was implemented in C++ and parallelized using OpenMP technology. In this implementation, the computational performance scaled linearly with the system size, as shown

in Fig. 1. We routinely used 14–28 simulation threads per simulation, determined by the number of cores in one Lomonosov-2 node. The use of multiple nodes, typically 20–30, of Lomonosov-2 was critical for obtaining the required statistics to extract ensemble-averaged characteristics of the simulated microtubule tips and enable comparison with experimental data.

### 1.3. Monte Carlo Simulations of Microtubule Dynamic Instability

Stochastic transitions between microtubule growth and shortening phases, termed catastrophes and rescues, are the most striking and captivating aspects of microtubule behavior. In cells, these transitions are essential, and their frequency is tightly controlled throughout the cell cycle. For example, the frequencies of catastrophes increase and the frequencies of rescues decrease by an order of magnitude when the cell enters the mitotic division stage.

Despite considerable theoretical and experimental work, our understanding of catastrophes and rescues remains incomplete. Recent observations suggest that factors beyond the GTP cap contribute to these transitions [11]. Computational modeling of microtubule catastrophes and rescues at the level of tubulin subunits (reviewed in [37]) usually involves kinetic Monte Carlo simulations, because the MD and BD models, such as described above, fail to achieve the necessary spatiotemporal scale of micrometers and minutes, dictated by the frequencies of catastrophes and rescues, observed in cells and in purified systems *in vitro*. Efficient algorithms for such simulations, pioneered by Gillespie, rely on calculating the time to the next reaction and randomly selecting the reaction to occur based on its relative probability [17].

Our motivation for creating a new Monte Carlo model of microtubule dynamics was two-fold. First, we aimed to determine whether the novel mechanism of microtubule assembly conceptualized through MD and BD simulations was consistent with the phenomenology of microtubule transitions and whether it could offer new insights into dynamic instability. Second, we sought to resolve long-standing questions about the origin of microtubule ‘aging’ and clarify the mechanism of microtubule rescue.

We developed a model that introduces two structural states of tubulin, ‘curved’ and ‘straight’, in addition to the two biochemical states, ‘GDP-bound’ and ‘GTP-bound’ [3]. Previous Monte Carlo models had only considered the two biochemical states (reviewed in [37]). This extension allowed us to incorporate information about the curved structures of GTP- and GDP-bound tubulins in solution and at dynamic microtubule ends, thereby providing a continuous description of microtubule behavior across multiple scales. Our ‘four-state’ Monte Carlo model enabled an improved description of the dependence of microtubule catastrophe frequencies on soluble tubulin concentration and microtubule polymerization time, also known as microtubule ‘aging’. Additionally, the model provided insights into the possible role of lattice defects and their repair by GTP-bound tubulins, which we further confirmed experimentally [3, 4].

The four-state Monte Carlo model was implemented in Matlab 2021a. Generally, the time step in this type of simulation depends on the specific kinetic rates. In the fully parameterized four-state model of the microtubule, the average time step was approximately equal to 1/300 of a second. Our most efficient implementation, which was not parallelized, required approximately equivalent to or even shorter clock time to simulate a second of model time (Fig. 1). This computational efficiency allowed us to collect substantial statistics within a reasonable timeframe by running several simulation trajectories in parallel or sequentially on a single node of the Lomonosov-2 supercomputer. The simulations easily covered timescales of hours in model



time, enabling the observation of repeated transitions from growth to shortening within a single simulation run.

## Conclusion

The multi-scale simulations discussed here have facilitated the construction of a coherent, integrated model of microtubule dynamics, consistent with a broad spectrum of experimental observations, including recent structural data on the conformations of tubulin dimers and oligomers in solution and at dynamic microtubule ends. The new perspective on microtubule polymerization and dynamics that emerged from the modeling has been instrumental in guiding experiments to address long-standing and newly conceived puzzles in the field. These include the mechanisms of microtubule force generation, catastrophes, aging, rescues, and the roles of unstructured tubulin regions in the control of tubulin polymerization. Collectively, this work represents a significant computational effort made possible by the high-performance supercomputing resources, and a continuous support from the Moscow State University computational facilities.

Over the past decade, there has been a dramatic increase in computational capabilities, enabling simulations of biological systems at an unprecedented scale. We anticipate that future developments in hardware architecture, and more importantly, in the development of more accurate and efficient biomolecular simulation techniques – potentially enhanced by rapidly developing machine learning approaches – will open up exciting possibilities. These advancements will enable the creation of not only qualitatively consistent but quantitatively predictive models, covering spatiotemporal scales from atoms to entire cells, and describing processes ranging from nanoseconds to hours in duration.

## Acknowledgments

The analysis of computational performance of molecular dynamics and Brownian dynamics simulations of tubulin oligomers and microtubule tips was supported by the Russian Science Foundation grant No. 23-74-00007 (<https://rscf.ru/project/23-74-00007/>). The analysis of the performance of the Monte Carlo simulations was supported by the Scientific and Educational Mathematical Center “Sofia Kovalevskaya Northwestern Center for Mathematical Research” (agreement No. 075-02-2024-1426, 28.02.2024). Simulations were carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

*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. Abraham, M.J., Murtola, T., Schulz, R., *et al.*: GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1, 19–25 (2015). <https://doi.org/10.1016/j.softx.2015.06.001>
2. Aldaz, H., Rice, L.M., Stearns, T., Agard, D.A.: Insights into microtubule nucleation from

- the crystal structure of human  $\gamma$ -tubulin. *Nature* 435(7041), 523–527 (2005). <https://doi.org/10.1038/nature03586>
3. Alexandrova, V.V., Anisimov, M.N., Zaitsev, A.V., *et al.*: Theory of tip structure-dependent microtubule catastrophes and damage-induced microtubule rescues. *Proceedings of the National Academy of Sciences* 119(46), e2208294119 (2022). <https://doi.org/10.1073/pnas.2208294119>
  4. Anisimov, M.N., Korshunova, A.V., Popov, V.V., Gudimchuk, N.B.: Microtubule rescue control by drugs and MAPs examined with in vitro pedestal assay. *European Journal of Cell Biology* 102(4), 151366 (2023). <https://doi.org/10.1016/j.ejcb.2023.151366>
  5. Asbury, C.L., Tien, J.F., Davis, T.N.: Kinetochores' gripping feat: conformational wave or biased diffusion? *Trends in Cell Biology* 21(1), 38–46 (2011). <https://doi.org/10.1016/j.tcb.2010.09.003>
  6. Buey, R.M., Díaz, J.F., Andreu, J.M.: The nucleotide switch of tubulin and microtubule assembly: a polymerization-driven structural change. *Biochemistry* 45(19), 5933–5938 (2006). <https://doi.org/10.1021/bi060334m>
  7. Chen, J., Kholina, E., Szyk, A., *et al.*:  $\alpha$ -tubulin tail modifications regulate microtubule stability through selective effector recruitment, not changes in intrinsic polymer dynamics. *Developmental Cell* 56(14), 2016–2028 (2021). <https://doi.org/10.1016/j.devcel.2021.05.005>
  8. Cleary, J.M., Hancock, W.O.: Molecular mechanisms underlying microtubule growth dynamics. *Current Biology* 31(10), R560–R573 (2021). <https://doi.org/10.1016/j.cub.2021.02.035>
  9. Desai, A., Mitchison, T.J.: Microtubule polymerization dynamics. *Annual Review of Cell and Developmental Biology* 13(1), 83–117 (1997). <https://doi.org/10.1146/annurev.cellbio.13.1.83>
  10. Ermak, D.L., McCammon, J.A.: Brownian dynamics with hydrodynamic interactions. *The Journal of Chemical Physics* 69(4), 1352–1360 (1978). <https://doi.org/10.1063/1.436761>
  11. Farmer, V.J., Zanic, M.: Beyond the GTP-cap: Elucidating the molecular mechanisms of microtubule catastrophe. *Bioessays* 45(1), 2200081 (2023). <https://doi.org/10.1002/bies.202200081>
  12. Fedorov, V.A., Kholina, E.G., Gudimchuk, N.B., Kovalenko, I.B.: High-performance computing of microtubule protofilament dynamics by means of all-atom molecular modeling. *Supercomputing Frontiers and Innovations* 10(4), 62–68. <https://doi.org/10.14529/jsfi230406>
  13. Fedorov, V.A., Kholina, E.G., Kovalenko, I.B., Gudimchuk, N.B.: Performance analysis of different computational architectures: Molecular dynamics in application to protein assemblies, illustrated by microtubule and electron transfer proteins. *Supercomputing Frontiers and Innovations* 5(4), 111–114. <https://doi.org/10.14529/jsfi180414>

14. Fedorov, V.A., Kholina, E.G., Kovalenko, I.B., *et al.*: Update on performance analysis of different computational architectures: Molecular dynamics in application to protein-protein interactions. *Supercomputing Frontiers and Innovations* 7(4), 62–67. <https://doi.org/10.14529/jsfi200405>
15. Fedorov, V.A., Orekhov, P.S., Kholina, E.G., *et al.*: Mechanical properties of tubulin intra- and inter-dimer interfaces and their implications for microtubule dynamic instability. *PLoS Computational Biology* 15(8), e1007327 (2019). <https://doi.org/10.1371/journal.pcbi.1007327>
16. Feenstra, K.A., Hess, B., Berendsen, H.J.: Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems. *Journal of Computational Chemistry* 20(8), 786–798 (1999). [https://doi.org/10.1002/\(SICI\)1096-987X\(199906\)20:8<786::AID-JCC5>3.0.CO;2-B](https://doi.org/10.1002/(SICI)1096-987X(199906)20:8<786::AID-JCC5>3.0.CO;2-B)
17. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* 22(4), 403–434 (1976). [https://doi.org/10.1016/0021-9991\(76\)90041-3](https://doi.org/10.1016/0021-9991(76)90041-3)
18. Grishchuk, E.L., Efremov, A.K., Volkov, V.A., *et al.*: The Dam1 ring binds microtubules strongly enough to be a processive as well as energy-efficient coupler for chromosome motion. *Proceedings of the National Academy of Sciences* 105(40), 15423–15428 (2008). <https://doi.org/10.1073/pnas.0807859105>
19. Gudimchuk, N.B., Alexandrova, V.V.: Measuring and modeling forces generated by microtubules. *Biophysical Reviews* 15(5), 1095–1110 (2023). <https://doi.org/10.1007/s12551-023-01161-7>
20. Gudimchuk, N.B., McIntosh, J.R.: Regulation of microtubule dynamics, mechanics and function through the growing tip. *Nature Reviews Molecular Cell Biology* 22(12), 777–795 (2021). <https://doi.org/10.1038/s41580-021-00399-x>
21. Gudimchuk, N.B., Ulyanov, E.V., O’Toole, E., *et al.*: Mechanisms of microtubule dynamics and force generation examined with computational modeling and electron cryotomography. *Nature Communications* 11(1), 3765 (2020). <https://doi.org/10.1038/s41467-020-17553-2>
22. Höög, J.L., Huisman, S.M., Sebö-Lemke, Z., *et al.*: Electron tomography reveals a flared morphology on growing microtubule ends. *Journal of Cell Science* 124(5), 693–698 (2011). <https://doi.org/10.1242/jcs.072967>
23. Janke, C., Magiera, M.M.: The tubulin code and its role in controlling microtubule properties and functions. *Nature Reviews Molecular Cell Biology* 21(6), 307–326 (2020). <https://doi.org/10.1038/s41580-020-0214-3>
24. Kukulski, W., Schorb, M., Welsch, S., *et al.*: Correlated fluorescence and 3D electron microscopy with high sensitivity and spatial precision. *Journal of Cell Biology* 192(1), 111–119 (2011). <https://doi.org/10.1083/jcb.201009037>

25. McIntosh, J.R., O'Toole, E., Morgan, G., *et al.*: Microtubules grow by the addition of bent guanosine triphosphate tubulin to the tips of curved protofilaments. *Journal of Cell Biology* 217(8), 2691–2708 (2018). <https://doi.org/10.1083/jcb.201802138>
26. Mitchison, T., Kirschner, M.: Dynamic instability of microtubule growth. *Nature* 312(5991), 237–242 (1984). <https://doi.org/10.1038/312237a0>
27. Nawrotek, A., Knossow, M., Gigant, B.: The determinants that govern microtubule assembly from the atomic structure of GTP-tubulin. *Journal of Molecular Biology* 412(1), 35–42 (2011). <https://doi.org/10.1016/j.jmb.2011.07.029>
28. Páll, S., Zhmurov, A., Bauer, P., *et al.*: Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS. *The Journal of Chemical Physics* 153(13) (2020). <https://doi.org/10.1063/5.0018516>
29. Pecqueur, L., Duellberg, C., Dreier, B., *et al.*: A designed ankyrin repeat protein selected to bind to tubulin caps the microtubule plus end. *Proceedings of the National Academy of Sciences* 109(30), 12011–12016 (2012). <https://doi.org/10.1073/pnas.1204129109>
30. Powers, A.F., Franck, A.D., Gestaut, D.R., *et al.*: The Ndc80 kinetochore complex uses biased diffusion to couple chromosomes to dynamic microtubule tips. *Cell* 136(5), 865 (2009). <https://doi.org/10.1016/j.cell.2008.12.045>
31. Rapaport, D.C.: *The art of molecular dynamics simulation*. Cambridge University Press (2004)
32. Roll-Mecak, A.: The tubulin code in microtubule dynamics and information encoding. *Developmental Cell* 54(1), 7–20 (2020). <https://doi.org/10.1016/j.devcel.2020.06.008>
33. Ulyanov, E.V., Vinogradov, D.S., McIntosh, J.R., Gudimchuk, N.B.: Brownian dynamics simulation of protofilament relaxation during rapid freezing. *Plos One* 16(2), e0247022 (2021). <https://doi.org/10.1371/journal.pone.0247022>
34. Voevodin, V.V., Antonov, A.S., Nikitenko, D.A., *et al.*: Supercomputer lomonosov-2: large scale, deep monitoring and fine analytics for the user community. *Supercomputing Frontiers and Innovations* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>
35. Volkov, V.A., Zaytsev, A.V., Gudimchuk, N., *et al.*: Long tethers provide high-force coupling of the Dam1 ring to shortening microtubules. *Proceedings of the National Academy of Sciences* 110(19), 7708–7713 (2013). <https://doi.org/10.1073/pnas.1305821110>
36. Zakharov, P., Gudimchuk, N., Voevodin, V., *et al.*: Molecular and mechanical causes of microtubule catastrophe and aging. *Biophysical Journal* 109(12), 2574–2591 (2015). <https://doi.org/10.1016/j.bpj.2015.10.048>
37. Zakharov, P.N., Arzhanik, V.K., Ulyanov, E.V., *et al.*: Microtubules: dynamically unstable stochastic phase-switching polymers. *Physics-Uspekhi* 59(8), 773 (2016). <https://doi.org/10.3367/UFNe.2016.04.037779>

# Numerical Analysis of OECD/NEA HYMERES Project Benchmark Tests Using CABARET-SC1 CFD Code

*Anton A. Kanaev<sup>1</sup>, Vyacheslav Yu. Glotov<sup>1</sup>*

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

The benchmark tests carried out within the OECD/NEA HYMERES (Hydrogen Mitigation Experiments for Reactor Safety) international project allowed to assess the capability of computational tools and to develop methodology for improving the modelling of complex safety issues relevant for the analysis and mitigation of a severe accident leading to hydrogen release into a nuclear plant containment. The paper presents the results of numerical simulation of two OECD/NEA HYMERES benchmark tests using CABARET-SC1 code. The code is based on the eddy resolving CABARET technique, which allows implicit modeling of the subgrid turbulence scales without using tuning parameters (ILES approximation). The absence of tuning parameters in the numerical approach allowed evaluating the influence of a separate physical phenomenon of radiative heat transfer. The influence of the mesh resolution in flow regions with complex geometries and the use of a porous medium model was also investigated.

*Keywords: CFD modeling, NPP hydrogen safety, ILES.*

## Introduction

During a severe accident at a nuclear power plant (NPP) with water-cooled reactors, a significant amount of hydrogen can be produced due to the oxidation of the zirconium cladding of the fuel rods at high temperatures. The release of hydrogen into the containment volume can lead to the formation of explosive mixtures of hydrogen and air, the combustion and detonation of which pose a serious threat to the integrity of the containment. During the Three Mile Island NPP accident in 1979, about 350 kg of hydrogen burned, fortunately causing no damage to the containment and thus not leading to significant radiological consequences for the environment or the population [3]. In more recent instance, during the Fukushima Daiichi NPP accident [6], a series of hydrogen explosions occurred, damaging the reactor buildings and resulting in the release of radioactivity into the atmosphere.

The distribution of hydrogen in the containment and the potential for the formation of localized areas with high hydrogen concentration are determined by complex thermal and hydraulic processes occurring at different stages of a severe accident. Therefore, ensuring the hydrogen explosion safety of NPPs during severe accidents represents a complex scientific and technical problem. Its solution requires a comprehensive approach, including both analytical and experimental research.

The first (2013–2016) [9] and second (2017–2021) [10] phases of the international OECD/NEA HYMERES project included experimental studies on the large-scale PANDA facility (PSI, Switzerland). The project aimed to improve understanding of thermal-hydraulic processes in severe accident scenarios involving hydrogen release into NPP containment, with an emphasis on the potential for mixing the areas of increased hydrogen concentration. Helium was used in all experiments to simulate hydrogen. All experiments were accompanied by analytical studies.

At the Nuclear Safety Institute (IBRAE), a non-parametric approach based on the CABARET method [1] is being developed for modeling turbulent flows in multicomponent media. CABARET belongs to eddy-resolving schemes with implicit subgrid turbulence modeling (ILES). It allows to carry out computations using meshes that do not fully resolve turbulence scales without using

---

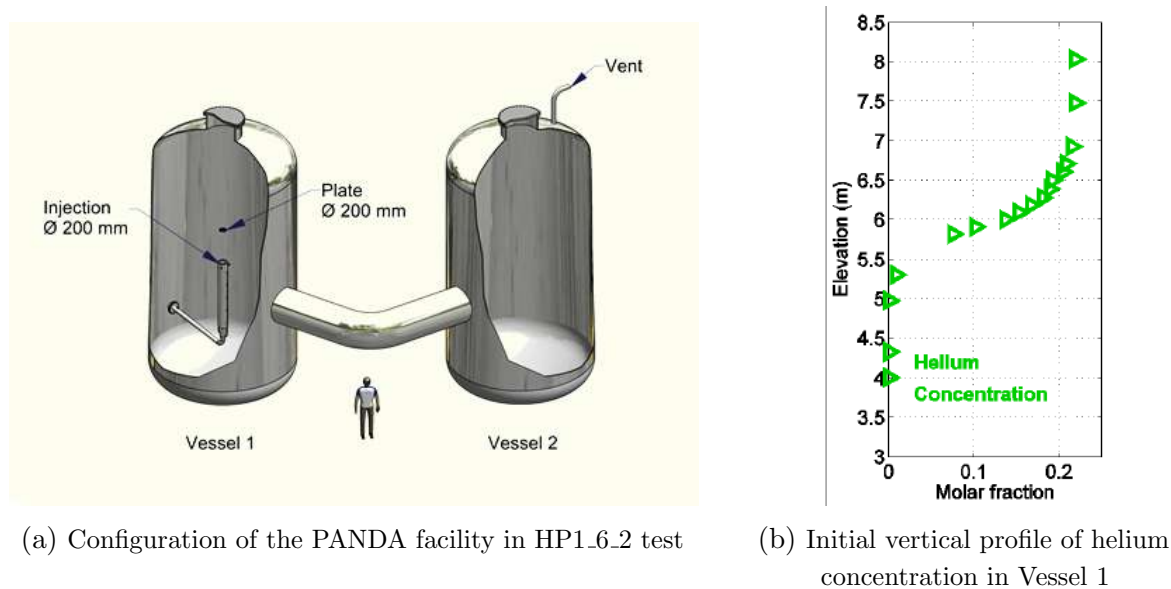
<sup>1</sup>Nuclear Safety Institute (IBRAE), Moscow, Russian Federation

tuning parameters. The only source of uncertainties is the mesh resolution, the selection criterion of which is based on the analysis of solution convergence. The CABARET methodology forms the basis of the CFD code CABARET-SC1 hydrodynamic solver [4, 7].

Within each phase of the OECD/NEA HYMERES project, one of the experiments investigating the mixing process of a helium-rich region by a flow formed after the interaction of a vertical steam jet with an obstacle was chosen as a benchmark test. This paper contains two main sections devoted to the description of the experimental setup, the approaches used in modeling and the results of the numerical analysis of the benchmark tests from the first (HP1\_6\_2 test) and second (H2P1\_10 test) phases of the OECD/NEA HYMERES project using CABARET-SC1 CFD code.

## 1. HP1\_6\_2 Test

One of the experiments on the PANDA facility with a horizontal obstacle shaped as a flat disk on the path of a vertical steam jet (HP1\_6\_2 test) was chosen as the benchmark test for the first phase of the OECD/NEA HYMERES project.



**Figure 1.** HP1\_6\_2 test setup

The configuration of the PANDA facility in HP1\_6\_2 test is shown in Fig. 1a. The height of the volumes (vessels) of the facility is  $\sim 8$  m, the diameter of each vessel is 4 m. Vessels are connected by a large (1 m) diameter Interconnecting Pipe (IP). During the pre-conditioning phase the vessels were filled with steam at  $108^\circ\text{C}$  and a helium-rich layer was created in the upper part of Vessel 1 (Fig. 1b). During the experiment, steam is injected from a round pipe located on the axis of Vessel 1. The outlet of the pipe is 2 m below the lower boundary of the helium-rich layer (which starts at 6 m from the bottom of Vessel 1). The steam flow rate is 60 g/s at a temperature of  $150^\circ\text{C}$ . Mixing is slowed down by a circular plate with a diameter of 20 cm, also located on the axis of the vessel at a distance of 1 m from the steam jet outlet. The pressure in the vessels during the experiment is maintained constant at 1.3 bar by venting the medium to the atmosphere through a valve at the top of Vessel 2. Before the start of the experiment,

the walls of the vessels (as well as the obstacle plate) were heated to the target temperature, ensuring the absence of condensation on the walls during the transient process.

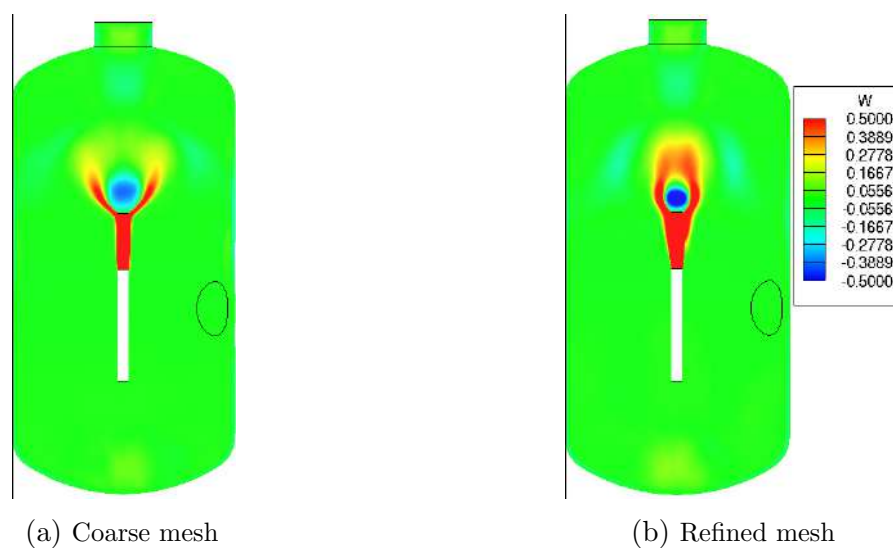
During the experiment, the distribution of thermal-hydraulic variables in the volume of the experimental setup is measured using the PANDA instrumentation consisting in a variety of sensor types. To measure the temperature in Vessels 1 and 2 and in the connecting pipeline, 374 thermocouples were installed. Helium, steam, and air concentrations at the PANDA facility are measured using two mass spectrometers. The gas mixture was sampled through capillaries (139 in total) which were installed near the thermocouple locations. The distribution of concentration and temperature measurement points was chosen to obtain detailed information about the flow structures and the erosion of the helium-rich layer. Helium concentration was measured at six levels above the steam injection level.

To carry out the simulation two computational meshes were constructed: coarse ( $\sim 1$  million hexahedral cells) and refined in the area of the steam jet propagation and in the area of the obstacle ( $\sim 3$  million hexahedral cells).

The initial conditions were set as the approximation of the experimental measurements [11]. Thermal insulation of the PANDA facility was not modeled directly. A third-kind boundary condition  $q = h \cdot (T - T_{ref})$  with a heat transfer coefficient obtained from the ad hoc heat loss measurements [12]  $h = 5.77 \cdot 10^{-3} \cdot (T[K] - 1.66)$  was set on the external boundaries of the steel walls of the Vessels and the IP, with a reference temperature  $T_{ref} = 20^\circ\text{C}$ .

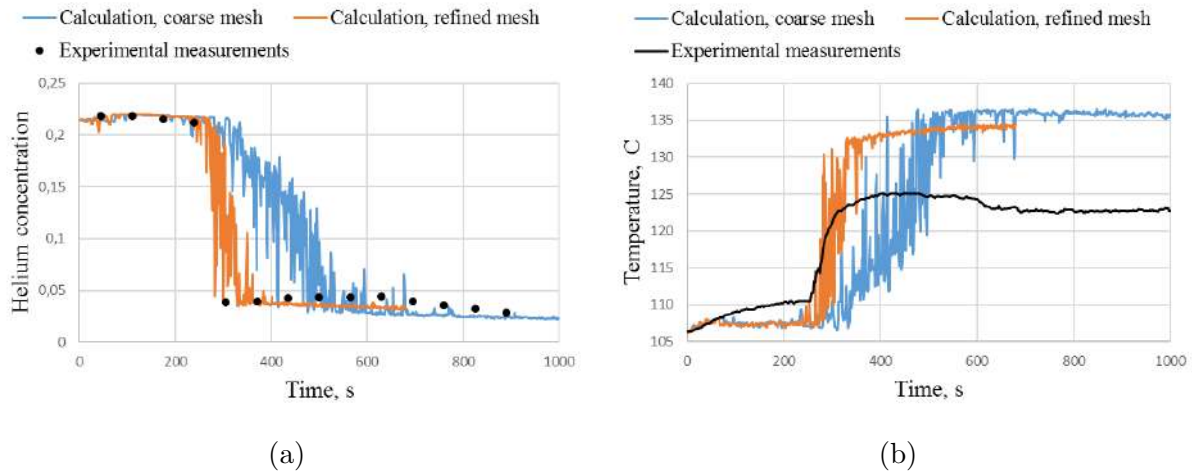
The computation of 1000 seconds of HP1.6.2 test on the Lomonosov-2 supercomputer [13] using CABARET-SC1 in the coarse mesh took about a week on 560 processors, and in the refined mesh – about two weeks on 980 processors. The time required for calculation is proportional not only to the overall number of cells but also to the numerical time steps which are reduced for the refined mesh in accordance with the CFL limitation.

Figure 2 shows a comparison of the vertical velocity distribution in experiment HP1.6.2 in the coarse and in the refined meshes. Different flow patterns of the steam jet in the area of the circular obstacle were observed. This is due to insufficient expansion of the modeled jet before the obstacle in the coarse mesh caused by insufficient resolution of the vortex flows formed in the mixing layer. In the refined mesh, the jet expands better and “reassembles” after the obstacle.



**Figure 2.** Distribution of the time-averaged vertical velocity in the calculations

Figure 3 shows the evolution of helium volume concentration and temperature at the point located on the axis of Vessel 1 at 2926 mm above the steam injection level for two meshes. It can be seen that the calculated helium concentration on the refined grid is significantly closer to the experimental measurements. Further calculations were conducted using the refined mesh.



**Figure 3.** Comparison of the calculated evolution of helium concentration (a) and temperature (b) with the experimental measurements

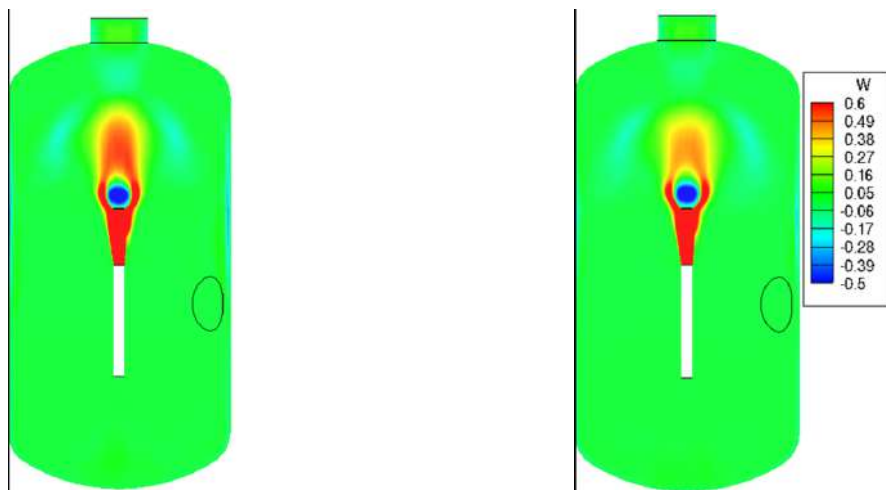
At the same time, it can be seen from Fig. 3b that the temperature is overestimated in the calculations for both meshes. One of the significant outcomes of the first phase of the OECD/NEA HYMERES project's benchmark test was the understanding that in tests with high steam content on the PANDA facility the heat transfer by radiation from the heated gas medium to the internal surface of the walls plays a significant role [12]. Experiments on the PANDA facility are conducted at relatively low steam temperatures ( $\sim 100\text{--}150^\circ\text{C}$ ), but even at these temperatures, radiation is significant [5]. Estimates show that the medium is optically dense, so diffusion approximations can be used as a radiation model, in particular, the Rosseland model.

Inclusion of the Rosseland radiation heat transfer model in the calculation of HP1\_6.2 test not only led to a good match between the calculated temperature and measurement results but also to a better approximation of the experimental helium concentration evolution in the upper area of Vessel 1. Due to more efficient heat removal by radiation from the gas medium to the walls in the calculation with the included Rosseland model, the vertical velocity of the steam flow after the obstacle is higher than in the calculation without considering radiation heat transfer due to its increased buoyancy (see Fig. 4).

The dynamics of the helium-enriched layer dissolution can be assessed by the decrease in helium concentration and the increase in temperature at sensors located on the axis of Vessel 1 at different heights above the steam injection level. Figures 5–7 show a comparison of helium concentration and temperature evolution in the calculation with the experimental measurements in the upper area of Vessel 1.

Observed differences in the time of the helium layer erosion at the two upper levels may be associated with increased heat losses in the area of the Vessel 1 manhole noted by the experimenters. These losses can be taken into account in the modeling using the measurements obtained with the wall thermocouples.

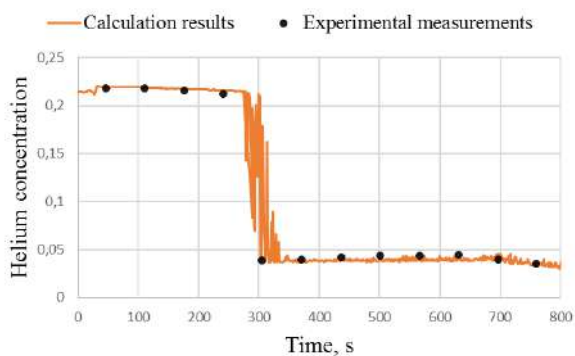




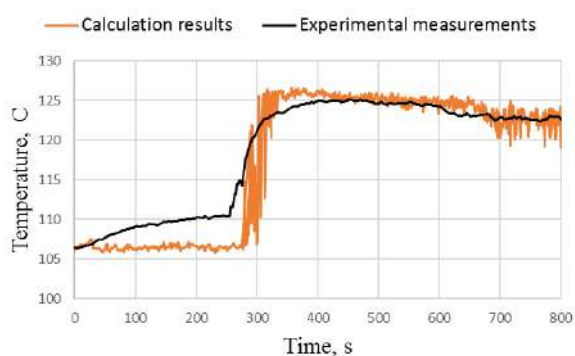
(a) Radiation heat transfer model included

(b) No radiation heat transfer model

**Figure 4.** Distribution of the time-averaged vertical velocity in the calculations

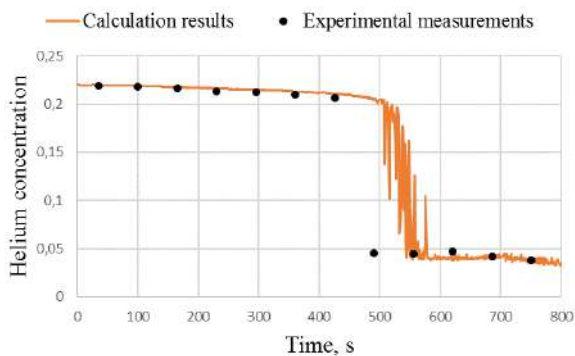


(a)

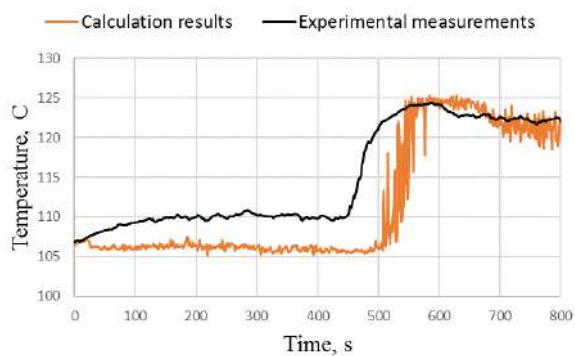


(b)

**Figure 5.** Comparison of the evolution of calculated helium concentration (a) and temperature (b) with the experimental measurements at 2926 mm above the steam injection level

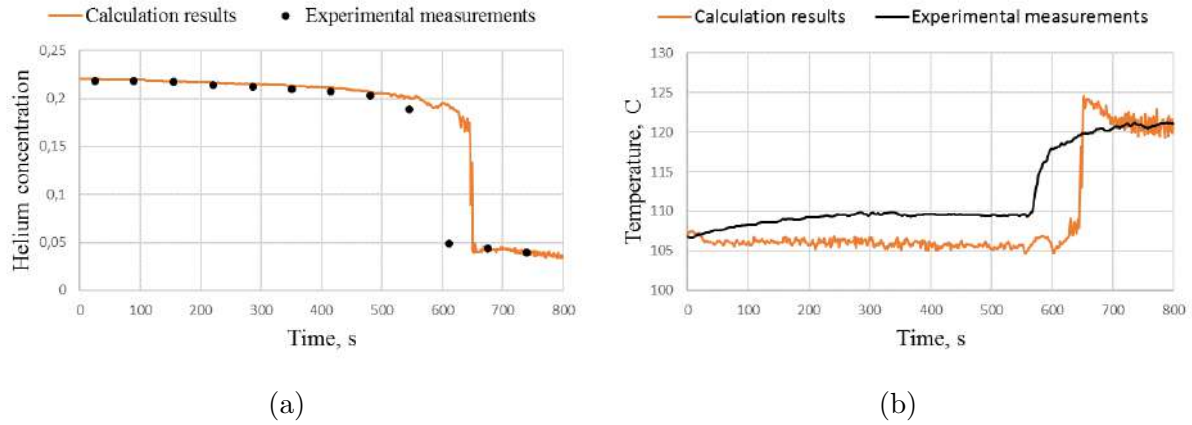


(a)



(b)

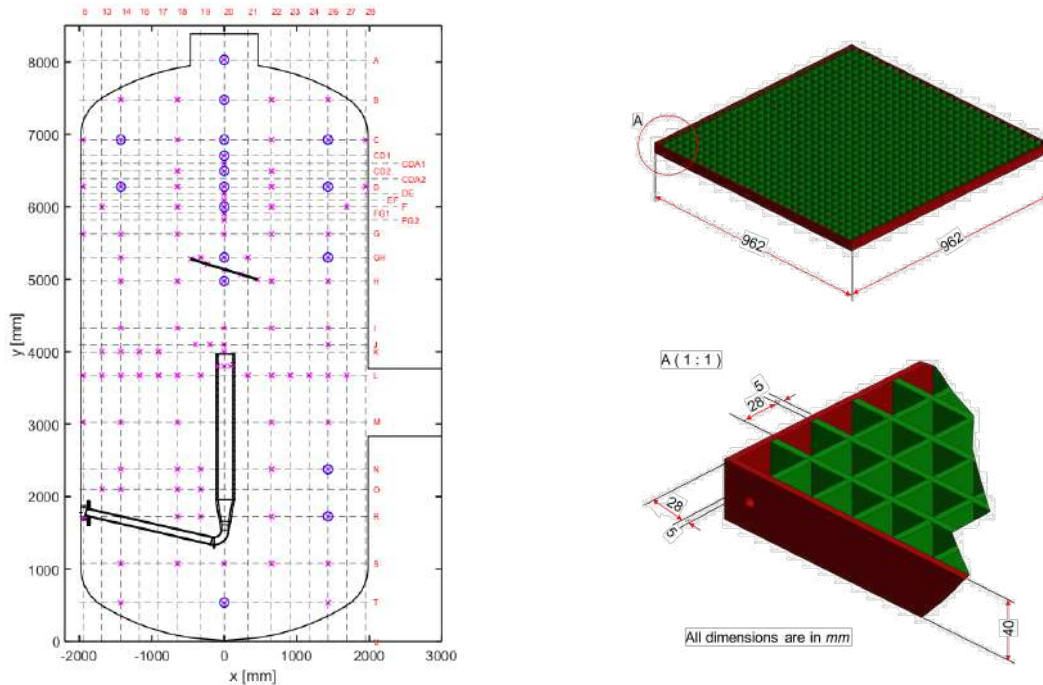
**Figure 6.** Comparison of the evolution of calculated helium concentration (a) and temperature (b) with the experimental measurements at 3478 mm above the steam injection level



**Figure 7.** Comparison of the evolution of calculated helium concentration (a) and temperature (b) with the experimental measurements at 4030 mm above the steam injection level

## 2. H2P1\_10 Test

As a benchmark test for the second phase of the OECD/NEA HYMERES project experiment H2P1\_10 was selected. The setup of H2P1\_10 test differs from that of HP1.6\_2 by the type of the obstacle in the path of the steam jet, which is a metal grid inclined at an angle of  $17^\circ$  to the horizontal direction (Fig. 8a). The center of the grid is located at the height of 1.138 meters above the steam injection pipe. Unlike the first phase of the project, all experiments of the second phase were conducted in a single vessel of the PANDA facility (Vessel 1). The pressure during the experiments was relieved through a valve located at the bottom of Vessel 1.

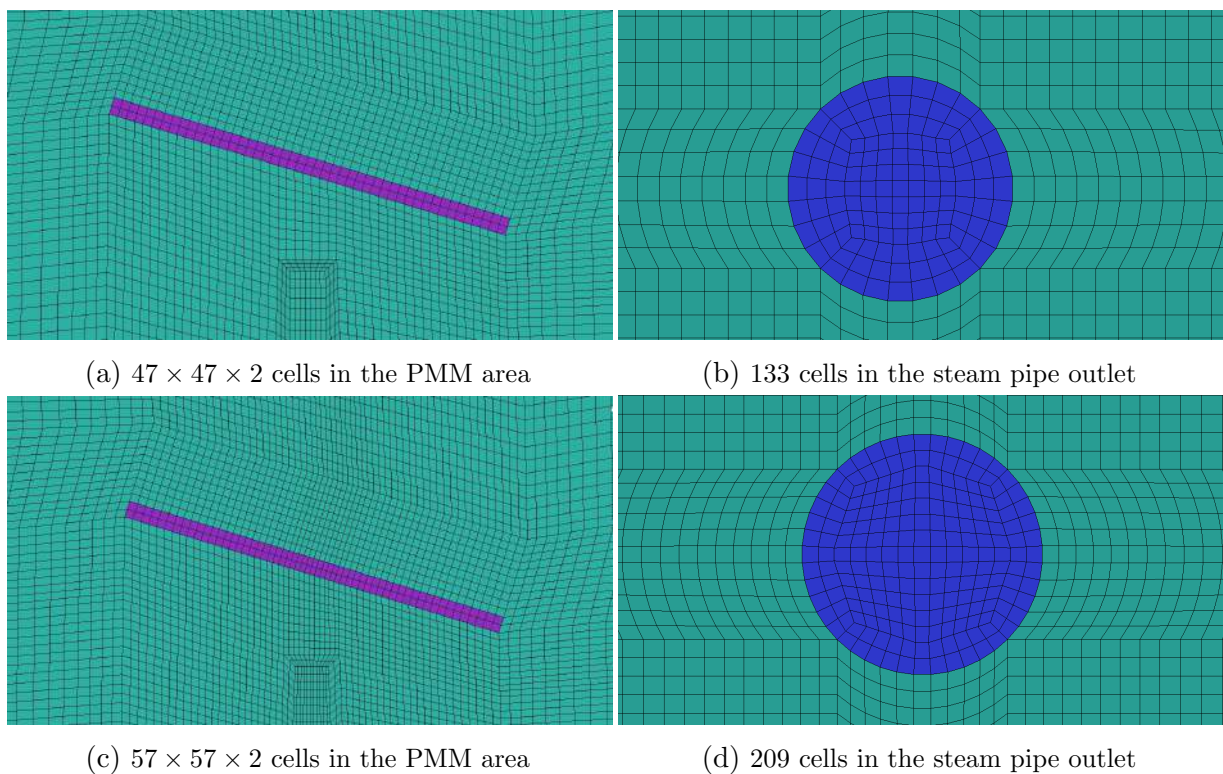


(a) Configuration of the PANDA experimental facility and the location of temperature and concentration measurement systems (b) Geometrical characteristics of the inclined grid

**Figure 8.** H2P1\_10 test setup

A complete simulation of the H2P1\_10 test was carried out using a porous medium model (PMM) simulating the grid's resistance to the jet flow. The parameters of the PMM were adjusted using the averaged results of a detailed calculation of 40–70 seconds time interval with the direct mesh resolution of the steam flowing through the grid. For the reference detailed calculation of H2P1\_10 test, a mesh with a resolution of  $3 \times 3 \times 4$  cells for each grid hole and 777 cells in the steam injection pipe outlet, totaling  $\sim 9$  million cells was constructed.

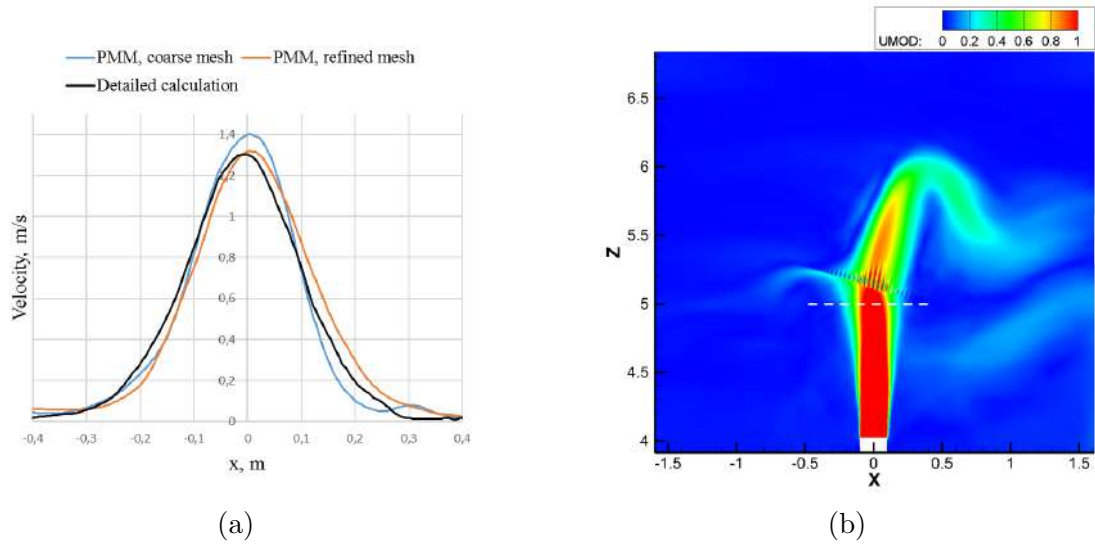
The number of computational cells in the mesh where the flow through the grid is not directly resolved is significantly lower than in the detailed mesh. For the calculation using PMM, the coarse ( $\sim 1.75$  million hexahedral cells,  $47 \times 47 \times 2$  cells in the porous model area, 133 cells in the steam injection pipe outlet) and the refined ( $\sim 2.7$  million hexahedral cells,  $57 \times 57 \times 2$  cells in the porous model area, 209 cells in the steam injection pipe outlet) meshes were constructed (Fig. 9).



**Figure 9.** Comparison of the coarse (a, b) and the refined (c, d) meshes

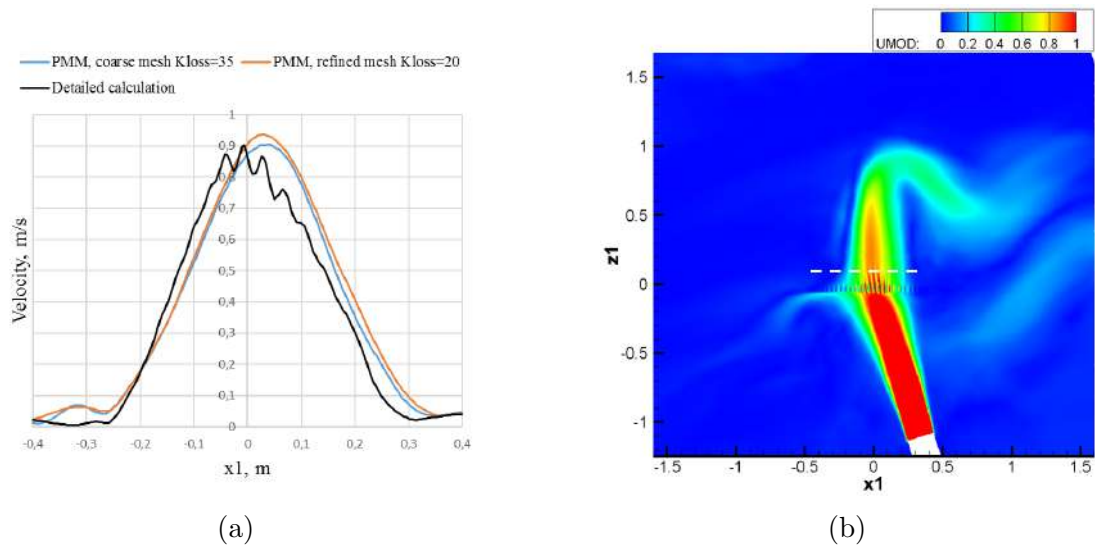
The detailed calculation of the first 70 seconds of H2P1\_10 test on the Lomonosov-2 super-computer using CABARET-SC1 took about 70 hours on 1120 processors. The calculation using PMM in the refined mesh required approximately 6.7 times less core-hours than the detailed calculation.

Figure 10a shows the comparison of time-averaged velocity magnitude values in the jet along the horizontal line at an altitude of 5000 mm above the bottom point, just before the jet contacts the inclined grid (Fig. 10b) in the detailed calculation and in the calculations with PMM. The maximum velocity magnitude values in the detailed calculation and in the PMM calculation on the refined grid coincide, while in the PMM calculation on the coarse grid, due to insufficient resolution in the jet area, the peak velocity magnitude is overestimated by  $\sim 8\%$ .



**Figure 10.** Distribution of the velocity magnitude (a) in the jet before the obstacle (b)

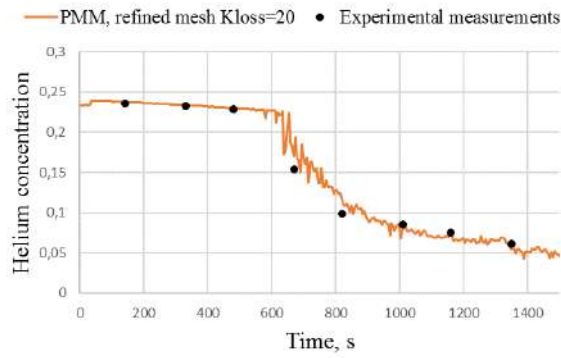
Figure 11a shows the comparison of the time-averaged velocity magnitude values in a coordinate system tied to the inclined grid ( $x_1 = x \cdot \cos(17^\circ) - z \cdot \sin(17^\circ)$ ,  $z_1 = x \cdot \sin(17^\circ) + z \cdot \cos(17^\circ)$ ), along a line parallel to the inclined grid at a distance of 10 cm from it (Fig. 11b) in the detailed calculation and in the calculations using PMM. The hydrodynamic loss coefficient of the flow in  $z_1$  direction, perpendicular to the grid,  $K_{loss}$  was adjusted using a series of calculations to achieve the best match with the detailed calculation results for the same averaging time interval ( $K_{loss} = 35m^{-1}$  in the coarse mesh,  $K_{loss} = 20m^{-1}$  in the refined mesh).



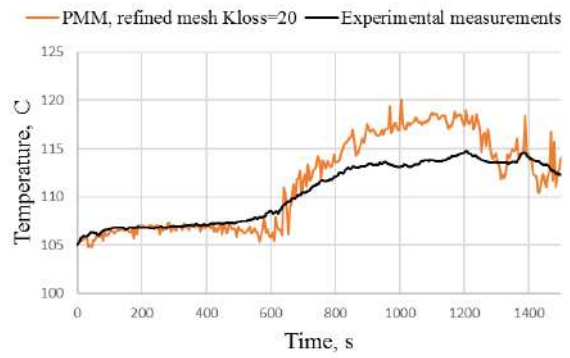
**Figure 11.** Distribution of the velocity magnitude (a) in the flow above the inclined grid (b)

The maximum velocity drop in the flow passing through the inclined grid in the detailed calculation is 29.58%; the maximum velocity drop in the flow passing through the PMM area is 36.79% for the coarse mesh and 30.38% for the refined mesh.

The calculation results of H2P1\_10 test with the CABARET-SC1 code using the porous medium model with the adjusted hydrodynamic loss coefficient  $K_{loss} = 20m^{-1}$  in the refined mesh led to a good agreement in the evolution of the helium-rich layer erosion and the temperature distribution with the experimental data (Figs. 12–14).

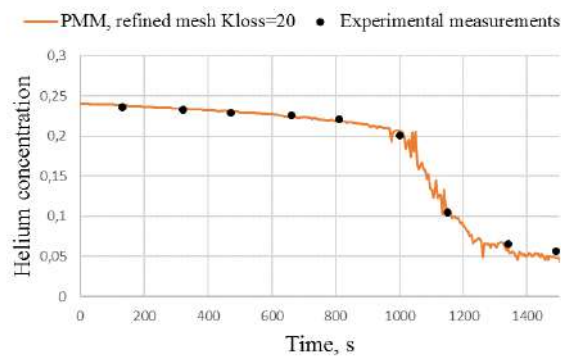


(a)

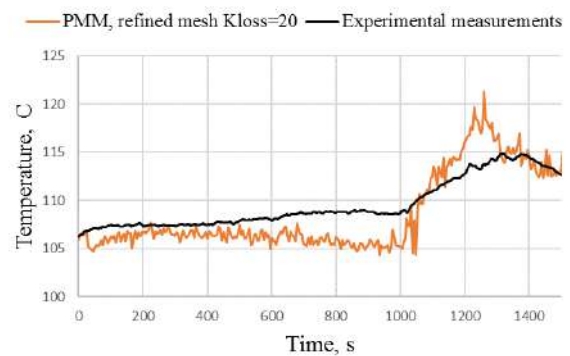


(b)

**Figure 12.** Comparison of the evolution of calculated helium concentration (a) and temperature (b) with the experimental measurements at 2926 mm above the steam injection level

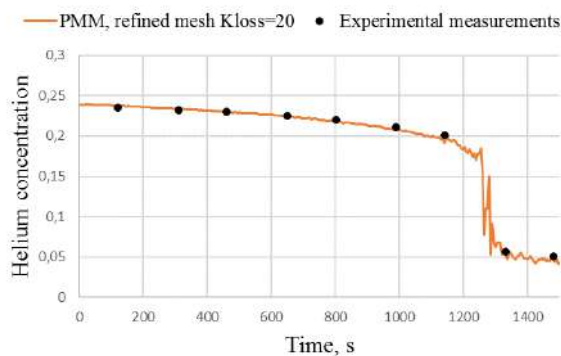


(a)

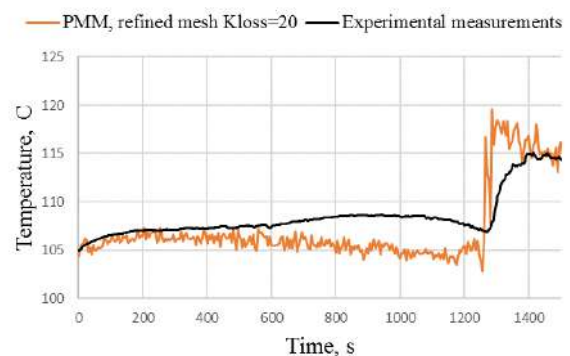


(b)

**Figure 13.** Comparison of the evolution of calculated helium concentration (a) and temperature (b) with the experimental measurements at 3478 mm above the steam injection level



(a)



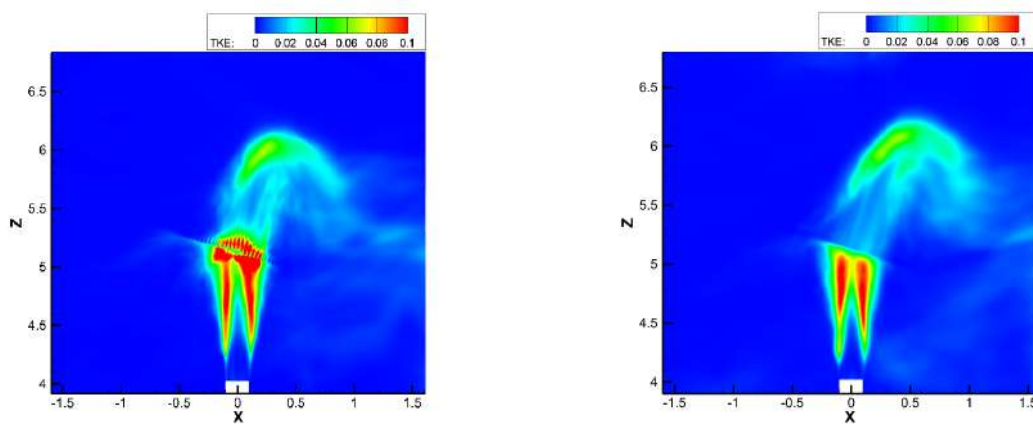
(b)

**Figure 14.** Comparison of the evolution of calculated helium concentration (a) and temperature (b) with the experimental measurements at 4030 mm above the steam injection level

In the developed porous medium model, heat losses on the grid were not modeled. This causes the temperature of the steam after passing through the grid and then in the area of contact with the helium-rich layer to be overestimated. Also, after passing through the PMM, the flow velocity oscillations are almost nullified. Figures 15 and 16 show a qualitative comparison of the TKE distribution in the detailed calculation, PMM calculation and the experiment. The TKE value is determined as:

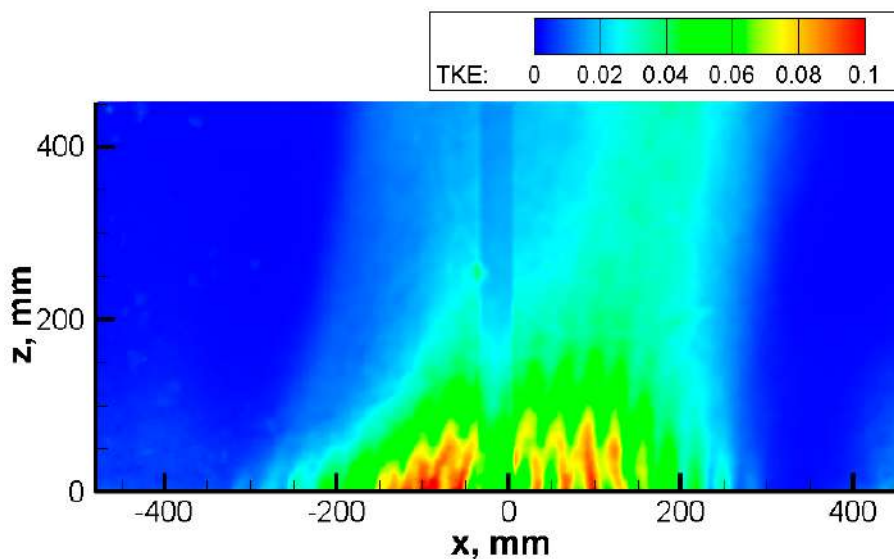
$$TKE = \frac{1}{2} \left( (\sigma V_x)^2 + (\sigma V_z)^2 + \frac{1}{2} [(\sigma V_x)^2 + (\sigma V_z)^2] \right),$$

where  $\sigma V_x$  and  $\sigma V_z$  are the standard deviations of the velocity components in  $y = 0$  plane. In the experiment, two-dimensional velocity fields in the rectangular area above the grid (the field of view) were recorded using the PIV (Particle Image Velocimetry) system during several time periods.



(a) Detailed calculation averaged over 40–70 s    (b) PMM refined grid  $K_{loss} = 20m^{-1}$  averaged over 40–70 s

**Figure 15.** Calculated TKE distributions



**Figure 16.** TKE distribution in the PIV field of view. Measurements were recorded over a time period of 743.6–948.4 s

Due to the absence of velocity oscillations behind the grid, the deflected flow mixes with surrounding atmosphere less intensively and its velocity in the area of contact with the helium-rich layer is overestimated. One of the OECD/NEA HYMERES project participants conducted H2P1\_10 test simulation in the RANS approximation [8] also using PMM. Taking into account heat losses on the grid and with the adjusted velocity oscillation distribution behind the grid, the hydrodynamic loss coefficient of the flow in the direction perpendicular to the inclined grid was estimated as  $K_{loss} = 15m^{-1}$ .

## Conclusion

The experiments conducted in the OECD/NEA HYMERES project covered a wide range of complex interconnected processes and phenomena. The most comprehensive understanding of the physical phenomena observed in experiments. The calculations of complex flows with obstacles in the OECD/NEA HYMERES benchmark tests with the CABARET-SC1 CFD code showed that insufficient local resolution of vortex structures can affect the simulated transient differently. In the HP1\_6\_2 benchmark test simulations, the resolution of the mesh in the jet area significantly affects the flow pattern behind the obstacle in the path of the jet. Insufficient mesh resolution leads to an underestimation of the helium-rich layer dissolution dynamics. In the H2P1\_10 benchmark test simulations insufficient mesh resolution in the jet area, on the contrary, leads to an overestimation of the time required for complete mixing of the helium layer. The absence of other tuning parameters in the numerical approach allows evaluating the influence of separate physical processes on the observed experimental picture. Inclusion of the radiation heat transfer model in the computational model led to a good agreement with experimental measurements for both temperature and the dynamics of the helium-rich layer mixing.

The calculation of the H2P1\_10 test was conducted using a porous medium model (PMM) simulating the resistance of a metal grid to the jet flow. The hydrodynamic loss coefficient in the PMM was adjusted using the results of a detailed calculation (with the direct mesh resolution of the steam flowing through the grid). This approach allowed achieving good agreement of the calculation results using the porous medium model with experimental measurements.

## Acknowledgements

This research uses the data from the high quality experiments conducted in PSI in the framework of HYMERES project. We thank our colleagues from HYMERES-1 and HYMERES-2 groups who provided insight and expertise that greatly assisted the research.

The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [13].

*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. Afanasiev, N., Goloviznin, V., Solovjev, A.: CABARET scheme with improved dispersion properties for systems of linear hyperbolic-type differential equations. Numerical Methods

- and Programming 22, 67–76 (2021). <https://doi.org/10.26089/NumMet.v22r105>
2. Andreani, M., Gaikwad, A.J., Ganju, S., *et al.*: Synthesis of a CFD benchmark exercise based on a test in the PANDA facility addressing the stratification erosion by a vertical jet in presence of a flow obstruction. *Nuclear Engineering and Design* 354, 110177 (2019). <https://doi.org/10.1016/j.nucengdes.2019.110177>
  3. Bal, R.S.: *Nuclear Safety in Light Water Reactors*. Elsevier, 2012. <https://doi.org/10.1016/C2010-0-67817-5>
  4. Bolshov, L., Glotov, V., Goloviznin, V., *et al.*: Cabaret-Sc1 Code Validation in Experiments on Hydrogen Explosion Safety at NPP. *Atomic Energy* 127, 216–222 (2020). <https://doi.org/10.1007/s10512-020-00613-7>
  5. Filippov, A., Grigoryev, S., Tarasov, O.: On the possible role of thermal radiation in containment thermal-hydraulics experiments by the example of CFD analysis of TOSQAN T114 air-He test. *Nuclear Engineering and Design* 310, 175–186 (2016). <https://doi.org/10.1016/j.nucengdes.2016.10.003>
  6. Jammal R., *et al.*: *The Fukushima Daiichi Accident*. IAEA, Vienna, Austria (2015).
  7. Kanaev, A.: Modeling of the influence of local heat sources on a light gas stratification formation and erosion in a large-scale experimental facility using eddy resolving numerical approach. *Nuclear Engineering and Design* 421, 113037 (2024). <https://doi.org/10.1007/s10512-020-00613-7>
  8. Kelm, S., Liu, X., Liu, X., *et al.*: The Tailored CFD Package ‘containmentFOAM’ for Analysis of Containment Atmosphere Mixing, H<sub>2</sub>/CO Mitigation and Aerosol Transport. *Fluids* 6(3), 100 (2021). <https://doi.org/10.3390/fluids6030100>
  9. Paladino, D., Mignot, G., Kapulla, R., *et al.*: OECD/NEA HYMERES Project: For the Analysis and Mitigation of a Severe Accident Leading to Hydrogen Release Into a Nuclear Plant Containment - 14322. American Nuclear Society, United States. <https://www.oecdnea.org/jointproj/hymeres.html>
  10. Paladino, D., Kapulla, R., Paranjape, S., *et al.*: PANDA experiments within the OECD/NEA HYMERES-2 project on containment hydrogen distribution, thermal radiation and suppression pool phenomena, *Nuclear Engineering and Design* 392, 111777 (2022). <https://doi.org/10.1016/j.nucengdes.2022.111777>
  11. Paranjape, S., *et al.*: OECD-NEA/HYMERES project: PANDA Test HP1.6.2 Quick-Look Report. Tech. Rep. TM-42-15-08, Rev-0, HYMERES-P-15-20 Paul Scherrer Institute (2015).
  12. Paranjape, S., *et al.*: OECD-NEA/HYMERES project: PANDA test facility description and geometrical specifications. Tech. Rep. TM-42-13-12, HYMERES-P-13-04 Paul Scherrer Institute (2013).
  13. Voevodin, Vl., Antonov, A., Nikitenko, D., *et al.*: Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. In *Journal: Supercomputing Frontiers and Innovations* 6(2), 4–11 (2019). <https://doi.org/10.14529/jsfi190201>