

# Supercomputing Frontiers and Innovations

2023, Vol. 10, No. 4

## Scope

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

## Editorial Board

### Editors-in-Chief

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

### Editorial Director

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

### Associate Editors

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

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

## Subject Area Editors

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

## Technical Editors

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

# Contents

<b>Numerical Simulation of Supersonic Free-Flow Stationary Disturbances Caused by Two-Dimensional Roughness in a Turbulent Boundary Layer</b> A.A. Yatskikh, L.V. Afanasev, A.D. Kosinov, N.V. Semionov .....	4
<b>2D Simulation of Micro-Jet Excitation by Heat Source</b> A.A. Firsov .....	12
<b>String-Wave Direct Parallel Solver for Sparse System of Linear Equations</b> A.Y. Likhosherstnyy, Y.G. Velikaya .....	21
<b>Three-dimensional Numerical Model of Kerosene Evaporation in Gas Turbine Combustors</b> A.A. Aksenov, S.V. Zhluktoy, V.S. Kashirin, M.L. Sazonova, S.G. Cherny, I.V. Zeziulin, M.D. Kalugina .....	27
<b>Analysis and Optimization of Output Operations in the INM RAS Earth System Model</b> M.A. Tarasevich, I.V. Tsybulin, E.M. Volodin, A.S. Gritsun .....	46
<b>High-Performance Computing of Microtubule Protofilament Dynamics by Means of All-Atom Molecular Modeling</b> V.A. Fedorov, E.G. Kholina, N.B. Gudimchuk, I.B. Kovalenko .....	62



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.

# Numerical Simulation of Supersonic Free-Flow Stationary Disturbances Caused by Two-Dimensional Roughness in a Turbulent Boundary Layer

*Aleksey A. Yatskikh*<sup>1</sup> , *Leonid V. Afanasev*<sup>1</sup> ,  
*Alexander D. Kosinov*<sup>1</sup> , *Nikolai V. Semionov*<sup>1</sup> 

© The Authors 2023. This paper is published with open access at SuperFri.org

Stationary disturbances in a free supersonic flow caused by two-dimensional roughness in a turbulent boundary layer on a wall were investigated using numerical simulation in the FlowVision software package. Calculations were performed for cases where the roughness heights were significantly smaller than the thickness of the boundary layer. In order to reduce the numerical oscillations, the computational grid has been adapted to the perturbation fronts by means. It was found that a disturbance in the form of a small amplitude N-wave is formed in the flow above the boundary layer. The effect of roughness height on disturbances formed in a free flow was studied. It was found that as the roughness height increases, there is an observed increase in the amplitude and spatial scale of the disturbance. It was also found that the gradient of the flow parameters between the disturbance fronts remains practically unchanged for all roughness heights considered. The numerical results were verified with experimental data. A strong agreement was achieved between the simulation and experimental result.

*Keywords:* supersonic flow, turbulent boundary layer, two-dimension roughness, N-wave.

## Introduction

The study focuses on investigating the impact of two-dimensional roughness on a wall in a turbulent boundary layer on stationary disturbances in a free supersonic flow above it. The case of a small roughness height compared to the thickness of the turbulent boundary layer is being considered. The relevance of this problem stems from the fact that roughness on the test section wall generates disturbances in the free flow during wind tunnel experiments [11]. These disturbances take the form of an N-wave, with weak shock waves forming at the fronts. When this disturbance impacts the leading edge of the model, significant distortions occur in both the mean flow and pulsation field within the laminar boundary layer on the model. Subsequently, it was experimentally shown that, in this case, the laminar-turbulent transition in the boundary layer can shift significantly upstream compared to the undisturbed case [9]. It has also been shown that the sensitivity of the boundary layer to such freestream disturbances depends on the bluntness of the leading edge [7], as well as its sweep angle [6].

In addition to experimental studies, the influence of free-stream disturbances from two-dimensional roughness on the wall of the test section on the flow in the boundary layer on the model is carried out using numerical simulation [2, 4, 5]. These works are focused on the influence of N-waves in the free stream on the boundary layer transition. The free-stream disturbance (N-wave) is specified using boundary conditions on the side boundary of the computational domain. This approach is described in detail, for example, in [3].

The influence of surface roughness on turbulent flow at supersonic speeds is studied in detail, for example, in [1, 8, 10]. However, they are primarily focused on studying the influence on the flow within the boundary layer, and the issues regarding the formation of disturbances in a free flow have not been sufficiently researched.

---

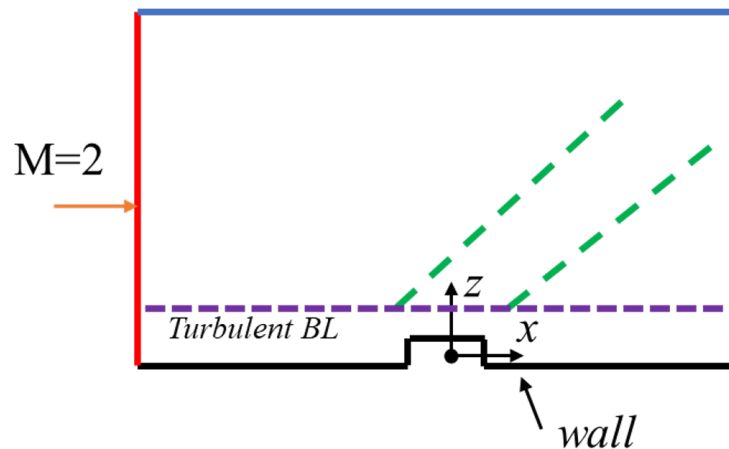
<sup>1</sup>Khristianovich Institute of Theoretical and Applied Mechanics SB RAS, Novosibirsk, Russian Federation

The amplitude and spatial characteristics of free-stream disturbances can significantly impact the transition from laminar to turbulent flow in boundary layers. Therefore, it is necessary to develop a technique for predicting the amplitude and spatial characteristics of free flow disturbances based on the parameters of two-dimensional roughness located within the turbulent boundary layer on the wall of the wind tunnel's test section.

This paper presents the results of a numerical simulation that investigates the formation of stationary free flow disturbances caused by small two-dimensional roughness in a turbulent boundary layer on a wall. The simulation was conducted using the FlowVision software package. The influence of roughness height on the generation of stationary disturbances in a free flow at Mach number  $M = 2$  is studied. Additionally, the results of the numerical simulation are validated using experimental data. The article is organized as follows. Section 1 is devoted to a description of the problem statement and the approaches used in this work. Section 2 presents the results of numerical studies. A comparison with experiments is also presented. Conclusion summarizes the study.

## 1. Simulation Set-Up

Figure 1 schematically shows the calculations set-up. In a turbulent boundary layer, a two-dimensional roughness of height  $h$  is established that is significantly less than the thickness of the boundary layer  $\delta$ . A stationary disturbance is formed in a supersonic flow above the boundary layer.



**Figure 1.** Computational domain

Numerical simulation was conducted using the FlowVision software package to study the formation of free flow disturbances caused by two-dimensional roughness on the wall. The flow is described within the framework of a continuum model for a compressible, viscous, and heat-conducting gas (air). Calculations were carried out in a two-dimensional formulation.

The entry condition was set at the left boundary of the computational domain. The flow parameters closely resemble those of the experiments conducted in the T-325 wind tunnel of the ITAM SB RAS at a Mach number of  $M = 2$ . The outlet condition was set at the upper and right boundaries. A zero temperature gradient was applied, and for velocities and pressure at the boundary, values equal to the value in the center of the boundary cell were set. This boundary condition is known as the supersonic outlet and is built into FlowVision. The wall

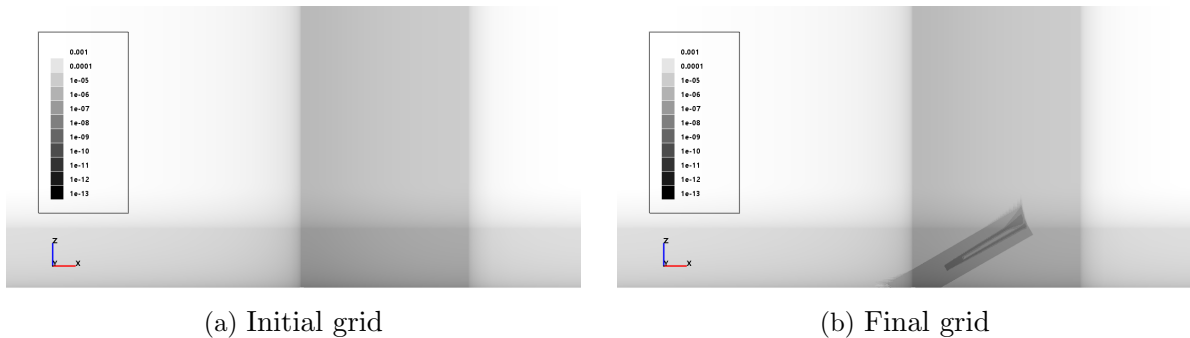
and two-dimensional roughness were modeled using the no-slip boundary condition, as well as the zero heat flux condition.

To simulate the turbulent boundary layer on the wall, the SpalartAllmaras turbulence model was used. To achieve a developed boundary layer in the roughness region, the computational domain was extended longitudinally upstream. In the roughness region, the Reynolds number, calculated from the longitudinal coordinate, was  $Re_x = 4 \cdot 10^6$ , and the thickness of the boundary layer significantly exceeded the height of the studied roughness.

As it is known from experiments, in such a setting, a disturbance in the form of an N-wave of small amplitude is formed in the flow above the boundary layer. At the N-wave fronts, the flow parameters change abruptly. If the computational grid resolution is insufficient, significant numerical oscillations can form at the fronts, which can make it difficult to determine the position of the fronts and the amplitude of the disturbances.

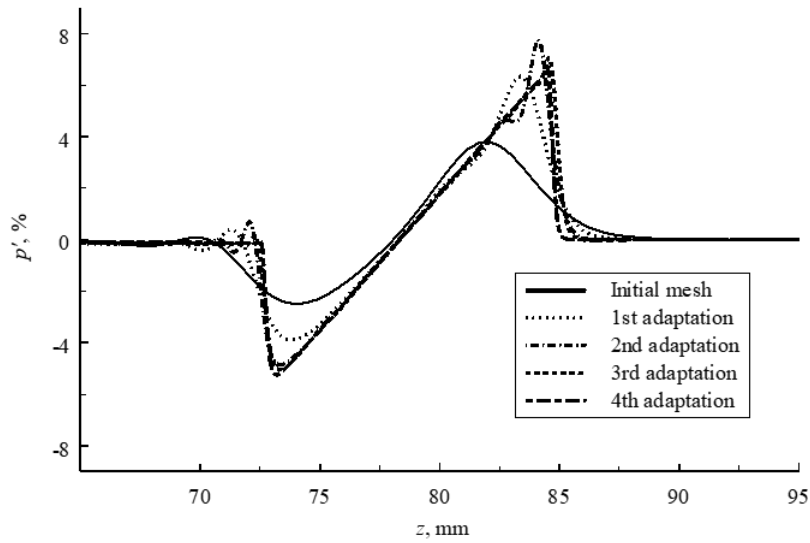
To minimize the influence of numerical oscillations, the FlowVision tool is used in this work to adapt the computational grid. In the calculation process, the computational cells were divided to account for the propagation of disturbances in the free flow and two-dimensional roughness regions. The calculation scenario was as follows. At the start, uniform initial parameters were set in the computational domain. The initial mesh was condensed in the area of the wall and roughness. After the solution reached a constant value, the first adaptation was applied: according to the condition of the density gradient, the mesh was refined in the area of roughness and disturbance in the free flow. After stabilization of the solution, another adaptation was carried out. A total of four adaptations of the computational grid were carried out.

Figure 2a shows distribution of the cell volume of initial mesh, normalized to the volume of the computational domain. Figure 2b shows distribution for the final computational grid. It can be seen that in the region of propagation of the disturbance from roughness the best mesh resolution is observed. The final computational grid had approximately 11 million cells. In the roughness region, wall distance of the boundary cells  $y^+ < 0.2$ .



**Figure 2.** Distribution of cell volumes for the initial and final computational grids

Figure 3 shows distribution of pressure disturbances in the control section at various levels of adaptation of the computational grid. With the initial grid, the N-wave fronts are blurred. As the size of the cell decreases, the disturbance fronts become sharper, but numerical oscillations appear. With a further reduction in the size of calculation cells, the numerical oscillations decrease. In the final configuration, their amplitude is significantly smaller compared to the amplitude of the disturbance caused by roughness.



**Figure 3.** Pressure disturbance at different levels of computational grid adaptation

Calculations were performed using the FlowVision software package version 3.13.02 on a computing node equipped with two 64-core AMD EPYC 7763 processors and 512 GB of RAM. The calculation for each case took approximately 30 hours.

To analyze the results in this study, we examine the distributions of pressure disturbances in the flow along the normal to the wall coordinate  $z$ . These distributions were determined as follows:

$$P' = \frac{P - P_\infty}{P_\infty} \cdot 100\%, \quad (1)$$

where  $P_\infty$  is pressure in the undisturbed flow above the boundary layer. Mass flow disturbances  $m'$  are determined similarly. The main analysis is carried out in a control section at a distance of 128 mm from the roughness.

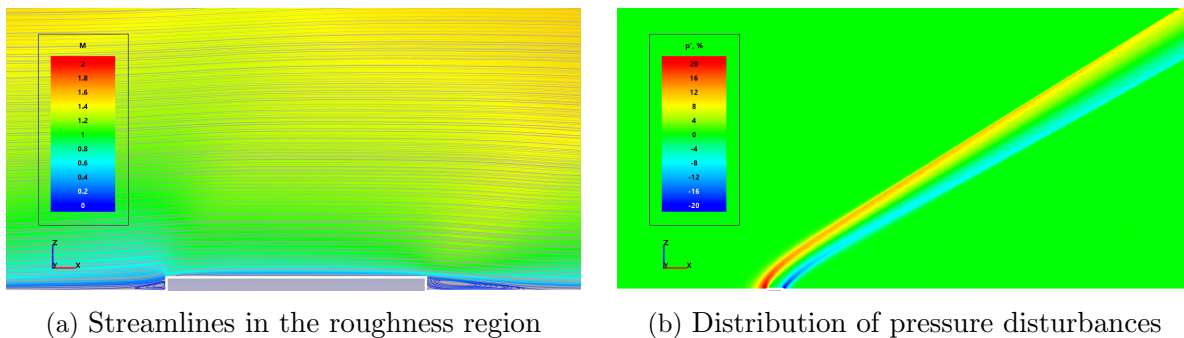
In this work, cases with different roughness sizes are considered: the length in the longitudinal direction of the roughness is constant in all cases and is equal to  $0.23 \cdot \delta$ , the height of the roughness is  $h/\delta = 0.005$  (wall distance calculated for smooth surface  $h^+ \approx 4$ ),  $0.012$  ( $h^+ \approx 8$ ),  $0.020$  ( $h^+ \approx 14$ ) and  $0.034$  ( $h^+ \approx 24$ ). All calculations were carried out at the flow Mach number  $M = 2$ .

## 2. Results

In the described research setting, it is possible to consider the propagation of disturbances caused by roughness in a turbulent boundary layer, both in the free flow and within the boundary layer. Figure 4a shows the streamlines in the region of roughness. The color of the lines represents the Mach number. Figure 4b shows the distribution of pressure disturbances.

Calculations show that flow separation zones are formed in front of and behind the roughness. The pressure fields clearly show that the disturbance fronts bend inside the boundary layer. This must be taken into account when setting up experimental studies.

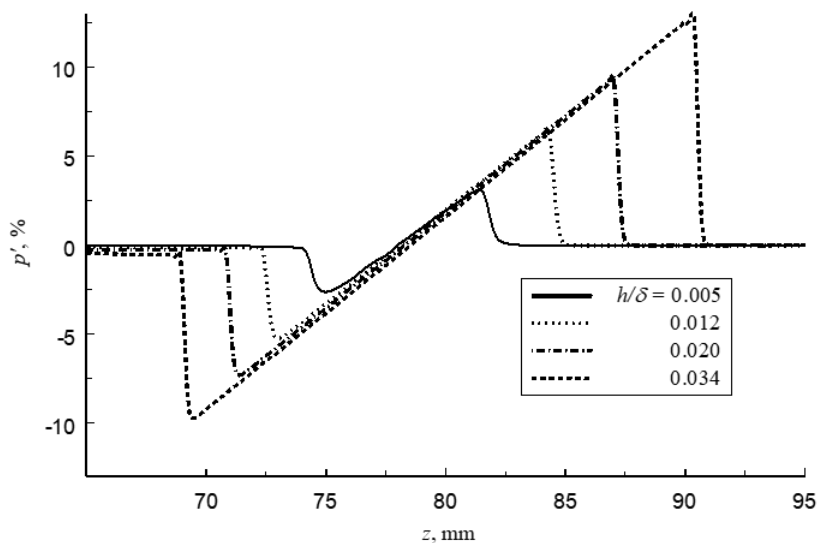
It is also clearly seen from the pressure disturbance fields that the amplitude of the disturbance generated by the roughness decays downstream. This is consistent with the results of [3],



**Figure 4.** Streamlines in the roughness region and the field of pressure disturbances.

in which the disturbance is specified by using boundary conditions. Moreover, our calculations show that the maximum pressure disturbance is observed within the boundary layer.

Calculation results for cases with varying roughness heights are presented in Fig. 5. The pressure disturbances of the free flow in the control section, located 128 mm downstream from the roughness, are shown.



**Figure 5.** Free-flow pressure disturbances at various roughness heights

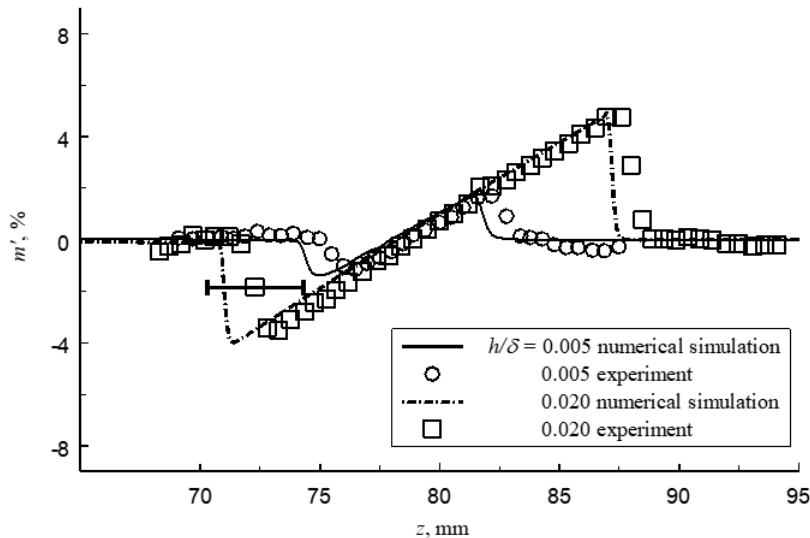
As the height of the roughness increases, there is an observed increase in the amplitude of the disturbance. At the same time, the spatial scale of the disturbed zone also increases, indicating a change in the angles of propagation of the disturbance fronts. The pressure gradient between the disturbance fronts remains practically unchanged in all cases.

The positions of the disturbance fronts in the flow differ significantly from the estimates made for Mach waves. Thus, if we consider the roughness edges as the source of Mach waves, their position in the control section should be at  $z = 75.3$  and  $73.9$  mm. This differs significantly from the results of provided calculations that take into account the turbulent boundary layer.

It should also be noted that at the first front of the disturbance (which is fixed at large values of the  $z$  coordinate), in all cases under consideration, the amplitude of changes in the flow parameters is approximately 1.3 times greater than at the trailing front of the disturbance.



Figure 6 shows a comparison between these calculations and the experimental results. The experiments were conducted in the T-325 wind tunnel at the ITAM SB RAS. A two-dimensional roughness was applied to the wall of the test section, and measurements were carried out using a constant temperature hot-wire anemometer. From the experimental data, the mass flow disturbance was determined. The mass flow rate was also determined using the numerical data for direct comparison. Figure 6 shows the perturbations of the mass flow  $m'$  for two different heights of two-dimensional roughness.



**Figure 6.** Comparison of numerical results and experimental data

The calculated data is in good agreement with the experimental results. The amplitude of disturbances, the gradient of mass flow between the disturbance fronts, and the spatial scales in calculations and experiments are similar. The differences in the spatial position of the disturbances are less than the error in determining their position in the experiment, which is attributed to the accuracy of the setting. In the experiment, the disturbance fronts are smoothed due to the physical dimensions of the sensitive element of the hot-wire probe.

## Conclusion

An efficient calculation set-up has been developed in the FlowVision software package to numerically simulate the formation of stationary free flow disturbances caused by two-dimensional roughness in a turbulent boundary layer on a wall. The case of small height roughness compared to the thickness of the boundary layer has been considered. To mitigate numerical oscillations, the FlowVision tool has been used to adapt the computational grid to disturbance fronts.

Numerical simulations have shown that flow separation zones are formed in front of and behind the two-dimensional roughness. The highest amplitude of disturbances is observed within the turbulent boundary layer. In a free flow, an observed disturbance in the form of an N-wave decays downstream.

The results of numerical modeling were verified using experimental data. A strong agreement was achieved between the simulation and experimental results, both in terms of the amplitude

and spatial position of the disturbances. It has been shown that it is possible to predict the amplitude, spatial scale, and location of a disturbance in a supersonic flow.

The influence of roughness height on the generation of stationary free flow disturbances has been studied. It was found that as the height of the roughness increases, there is an observed increase in the amplitude and spatial scale of the disturbance. In this case, the gradient of flow parameters between the disturbance fronts remains practically unchanged for all roughness height considered.

## Acknowledgements

The research was financially supported by the Russian Science Foundation (Project No. 22-19-00666, <https://rscf.ru/project/22-19-00666/>) using the equipment of the “Mechanics” Shared-Use Center, ITAM SB RAS. Numerical simulation was performed using the FlowVision software (<https://flowvision.ru/>).


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

## References

1. Aghaei-Jouybari, M., Yuan, J., Li, Z., *et al.*: Supersonic turbulent flows over sinusoidal rough walls. *Journal of Fluid Mechanics* 956, A3 (2023). <https://doi.org/10.1017/jfm.2022.1049>
2. Din, Q.H., Egorov, I.V., Fedorov, A.V.: Mach wave effect on laminar-turbulent transition in supersonic flow over a flat plate. *Fluid Dynamics* 53(5), 690–701 (Sep 2018). <https://doi.org/10.1134/S0015462818050063>
3. Dinh, H.Q., Nguyen, A.T., Egorov, I.V., Duong, N.H.: The study of Mach waves generated by a roughness element. *International Journal of Aeronautical and Space Sciences* 23(3), 511–520 (Jul 2022). <https://doi.org/10.1007/s42405-022-00471-6>
4. Egorov, I.V., Duong, N.H., Nguyen, N.C., Palchekovskaya, N.V.: Numerical simulation of the influence of a Mach wave on the laminar–turbulent transition in a supersonic boundary layer. *Doklady Physics* 67(5), 144–147 (May 2022). <https://doi.org/10.1134/S1028335822050019>
5. Khotyanovsky, D., Kudryavtsev, A., Kosinov, A.: Numerical study of the interaction of the N-wave with the plate leading edge in the supersonic stream. *AIP Conference Proceedings* 1893(1), 030051 (10 2017). <https://doi.org/10.1063/1.5007509>
6. Kocharin, V.L., Kosinov, A.D., Yatskikh, A.A., *et al.*: The impact of weak shock waves on the flow in the boundary layer of a flat plate with a variable sweep angle of the leading edge. *Thermophysics and Aeromechanics* 26(6), 803–809 (Nov 2019). <https://doi.org/10.1134/s0869864319060027>
7. Kosinov, A.D., Semionov, N.V., Yatskikh, A.A., *et al.*: Influence of the leading-edge bluntness radius of a plate on the response of flat-plate boundary layer to an N-wave at

- Mach number  $M = 2$ . *Thermophysics and Aeromechanics* 30(2), 227–237 (Aug 2023). <https://doi.org/10.1134/s086986432302004x>
8. Modesti, D., Sathyanarayana, S., Salvadore, F., Bernardini, M.: Direct numerical simulation of supersonic turbulent flows over rough surfaces. *Journal of Fluid Mechanics* 942, A44 (2022). <https://doi.org/10.1017/jfm.2022.393>
  9. Piterimova, M.V., Kosinov, A.D., Semionov, N.V., *et al.*: Experimental study of effect of a pair of weak shock waves on laminar-turbulent transition in the boundary layer of flat plate at Mach number 2. *Siberian Journal of Physics* 17(2), 30–40 (Sep 2022). <https://doi.org/10.25205/2541-9447-2022-17-2-30-40>
  10. Sun, Z., Zhu, Y., Hu, Y., Zhang, S.: Direct numerical simulation of a fully developed compressible wall turbulence over a wavy wall. *Journal of Turbulence* 19(1), 72–105 (2018). <https://doi.org/10.1080/14685248.2017.1395514>
  11. Vaganov, A.V., Ermolaev, Y.G., Kolosov, G.L., *et al.*: Impact of incident Mach wave on supersonic boundary layer. *Thermophysics and Aeromechanics* 23(1), 43–48 (Jan 2016). <https://doi.org/10.1134/s0869864316010054>

# 2D Simulation of Micro-Jet Excitation by Heat Source

Aleksandr A. Firsov<sup>1</sup> 

© The Author 2023. This paper is published with open access at SuperFri.org

Excitation of a laminar gas micro-jet by acoustic impact and by pulse-periodic heat source was simulated using the FlowVision software package in 2D formulation at normal conditions. Heat source imitates an electrical discharge. Air jet was formed by channel with inner size of 0.7 mm with the Poiseuille velocity profile at inlet boundary, the maximum profile velocity was varied in a range of 2.5–10 m/s. Influence of heat source frequency and power on the large-scale vortex formation was described. In the case of a jet with a speed of 5 m/s, the natural oscillations of the jet in response to a single pulse had a frequency  $f_{res} = 1380$  Hz, so excitation of the jet was possible at close frequencies of 1190 Hz and 1500 Hz. At the same time, at a frequency of 1000 Hz (approximately equal to  $2/3 f_{res}$ ), every second impulse acted in antiphase and the oscillations developed poorly. Dependence of flow structure from the jet velocity was obtained. The results obtained show the possibility of exciting a micro-jet using low-power electrical discharges such as spark, DBD or corona.

*Keywords: micro-jet, excitation, discharge, instability, modelling.*

## Introduction

Studies of the stability of jet flows and the mechanisms of their excitation are of significant interest for various devices, including burners and combustion chambers of various types, chemical reactors, and medical devices [1]. For example, work [2] provides data on the possibility of reducing NOx concentration during the combustion of a hydrogen jet in the air under acoustic influence. A significant part of work is devoted to the so-called micro-jets with a diameter of no more than 1 mm, which will be discussed in this work. The flow in micro-jets differs significantly from macrojets, the conditions for vortex formation change, and the penetration depth increases [3].

The closer the velocity profile is to parabolic (to the Poiseuille profile), the farther from the nozzle the laminar-turbulent transition and/or the excitation of instabilities of various types occurs. The velocity profile approaches parabolic due to the growth of the boundary layer, therefore, to obtain extended laminar jets, it is necessary to use long channels for which the diameter  $d$  is much less than the length  $L$  ( $L/d > 100$ ). In the applied sense, it is interesting both to increase the stability of such jets and to the early excitation of instabilities, leading to intensified mixing of the jet with the surrounding moving or stationary gas and intensification of combustion [4, 5].

The classical method of jet excitation is to use acoustic vibrations [6], however, in most works, the use of a plane wave is discussed, i.e. non-local influence [7–9]. Electrical discharges of various types can provide local effects: form shock waves, be a source of volumetric force and heat generation. For example, spark discharges have been considered in various settings as an actuator that intensifies the mixing of a transverse gas jet with a supersonic air flow, both experimentally [10] and using numerical modeling [11, 12]. It is noteworthy that the stability of micro-jets and the possibility of their excitation using acoustic vibrations through numerical modeling were practically not considered, with rare exceptions [13]. Therefore, it seems interesting to consider, using numerical modeling, the stability of the microjets and the possibility of its local excitation using a pulse-periodic energy input simulating a discharge.

---

<sup>1</sup>Joint Institute for High Temperatures of the Russian Academy of Sciences (JIHT RAS), Moscow, Russia

The article is organized as follows. Section 1 is devoted to description of a simulation model and boundary conditions. In section 2 we perform test of the ability to simulate the influence of an acoustic wave on a jet using FlowVision. Section 3 represents the results of simulation for fixed jet velocity at varying the frequency of the discharge and section 4 demonstrates the results for varying velocity of a discharge. Conclusion summarizes the study and reveals the possibility of using low-power electrical discharges such as spark, DBD or corona for exciting of a micro-jet.

## 1. Model and Boundary Conditions

Obtaining detailed data on the distribution of gas-dynamic quantities is extremely difficult in an experiment on excitation of a laminar micro-jet using an electric discharge. Numerical modeling can be used to obtain information about pressure and velocity fields throughout the entire study area. However, modeling has its limitations, for example, a three-dimensional problem has a significant dimension, i.e. solving such a problem requires either a large amount of computational time (weeks on available resources for each computational case), or a powerful computing cluster, but our group has limited access to such infrastructure. Therefore, a basic understanding of the influence of a weak disturbance created by an electric discharge on a laminar micro-jet was obtained in a 2D formulation.

Numerical modeling of the excitation of a jet by a thermal source simulating the operation of a discharge was performed using FlowVision 3.12 software package on a computer station with an Intel 4930K processor (6 cores) and 32 GB of RAM. The simulation is based on solving a three-dimensional non-stationary system of Navier–Stokes equations, but in one of the directions ( $z$ ) the dimension of the problem was 1 cell. The turbulence model was disabled.

The calculation area is a rectangle with a length (along the stream) of 60 mm and a width of 50 mm and presented in Fig. 1. There is a tube located along the axis of symmetry on one of the walls. Considering the 2D setting, this is a flat canal 5 mm long with an internal dimension of 0.7 mm and an external dimension of 0.9 mm. On the edge of the tube, there is a square with a side of 0.1 mm, inside of which there is a “heat source” modifier, simulating the operation of a discharge. The project used a rectangular computational mesh with adaptation – an increased level of adaptation (refinement of the computational mesh) was used in the area of jet flow. Thus, there are 96 cells per length and 50 cells per channel thickness inside the tube, and the total number of cells in the project did not exceed 0.5 million.

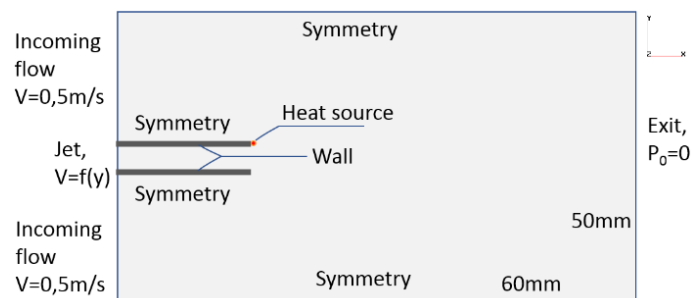


Figure 1. Calculation domain and boundary conditions

The “wall” boundary condition (no-slip boundary) was used on the inner wall of the tube and on the edge; a Poiseuille velocity profile was installed at the entrance of the tube; the maximum profile speed  $U_0$  varied in the calculations and was set to 2.5, 5 and 10 m/s. At the entrance to the computational domain from the side of the tube, a constant velocity boundary

condition of 0.5 m/s was set to minimize the mutual influence of pulsations created by the discharge. The working gas was air. Thus, the formulation of the problem is close to that described in [14, 15] for the case when the jet speed is greater than the oncoming flow speed. Therefore, the simulation results obtained for cases without discharge activation are close to the published experimental visualization. In part of the calculations, to accurately determine the concentration ratio between the jet gas and the associated gas and for better visualization, two gases were used: both having the properties of air. This made it possible to obtain visualization in the calculations similar to visualization using smoke/spray in the experiment. On the outer walls of the tube, as well as along the long boundary of the computational domain, a symmetry condition was established (a wall with slipping or a zero gradient of parameters). At the remaining boundary opposite the tube, a constant total relative pressure condition of zero was established. The relative temperature at all permeable boundaries was zero. The reference temperature and pressure were set to 273 K and 101000 Pa, respectively.

## 2. Testing the Ability to Simulate the Influence of an Acoustic Wave on a Jet Using FlowVision

To determine the possibility of modeling the influence of small pressure disturbances on the jet, a calculation was made of the acoustic excitation of a laminar jet with a speed of  $U_0 = 6$  m/s in the absence of a cocurrent flow. For this purpose, the lower and upper boundaries of the computational domain were changed from symmetry to the variable pressure domain (plane wave – sinusoidal disturbance with an amplitude of 2 Pa and a frequency of 1000 Hz) and to the non-reflective Riemann condition, respectively. In the absence of artificial pressure disturbances, the jet was deformed at a distance of more than 10 jet calibers, which is clearly visible both from the gas velocity distribution and from the pressure pulsations shown in Fig. 2. The inclusion of a source of variable pressure at the boundary of the computational domain led to the creation of a variable pressure gradient in the region of the edge of the tube, and to the development of acoustic instability of the jet, which is visible both in the velocity distribution and in the form of small-scale pressure pulsations developing from the edge of the tube along the jet. Thus, modeling the excitation of a micro-jet using small perturbations can be simulated using FlowVision CFD. Therefore, we can move on to the next part of the work – modeling excitation using a pulse-periodic heat source simulating a discharge.

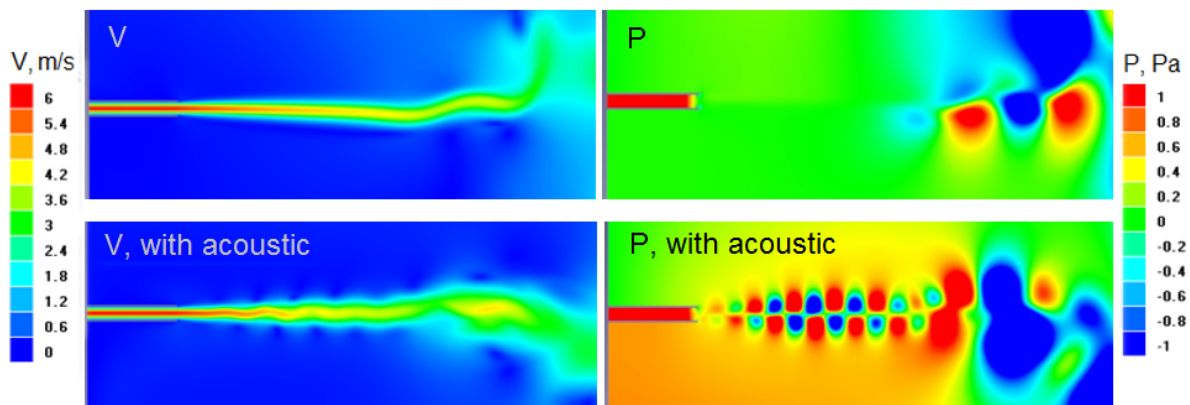
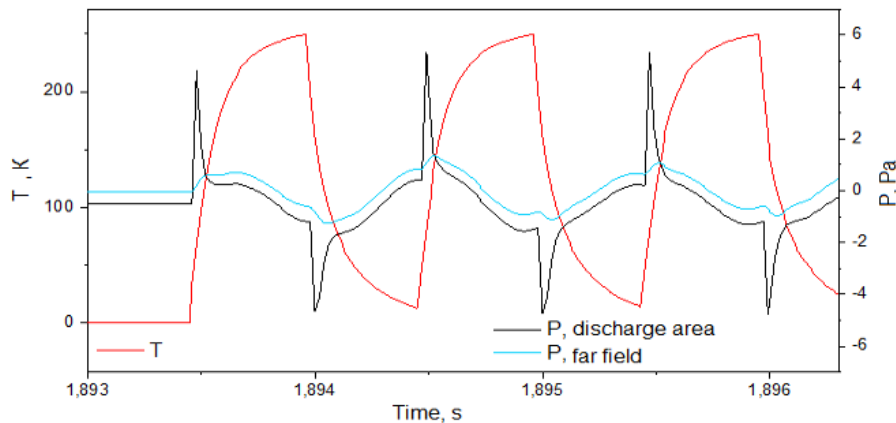


Figure 2. Velocity and pressure without and with acoustic excitation

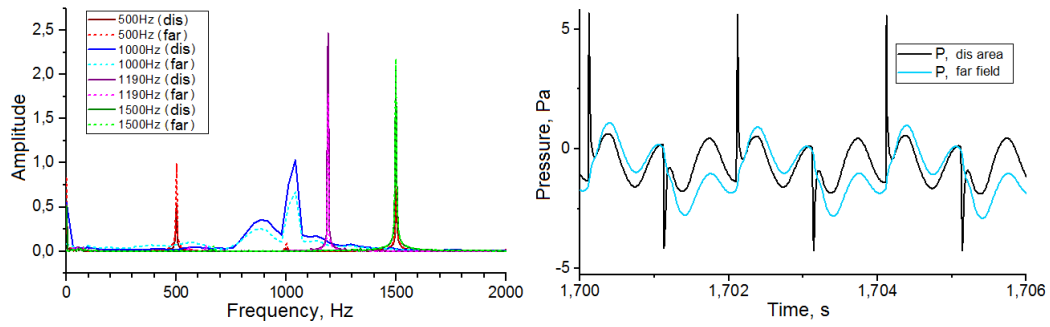
### 3. Results for Fixed Jet Velocity at Varying the Frequency of the Discharge

The discharge was simulated by a volumetric heat source located on the edge of the tube in a square of 0.1 mm in size. The discharge heated a small volume of gas to the maximum relative temperature of 200–600 K. The discharge power at the time of operation was 0.005–0.012 W. During a series of tests of various energy input profiles, it was found that the most accentuated pressure pulse occurs when the heat source is instantly turned on/off. Thus, Fig. 3 shows a graph of the relative temperature in the discharge area with a heating power of 0.006 W, operating for 0.5 ms with a period of 1 ms ( $f = 1000$  Hz) at a jet speed of 5 m/s. Turning on (and off) the heat source leads to pressure pulses in the near zone (i.e. in the area of the heat source), which are practically not felt at a distance from the source (in the far zone, 9 mm from the edge of the nozzle along the jet). However, as it can be seen in Fig. 3, in the jet region, these pressure pulses lead to the development and maintenance of pressure oscillations close to sinusoidal.

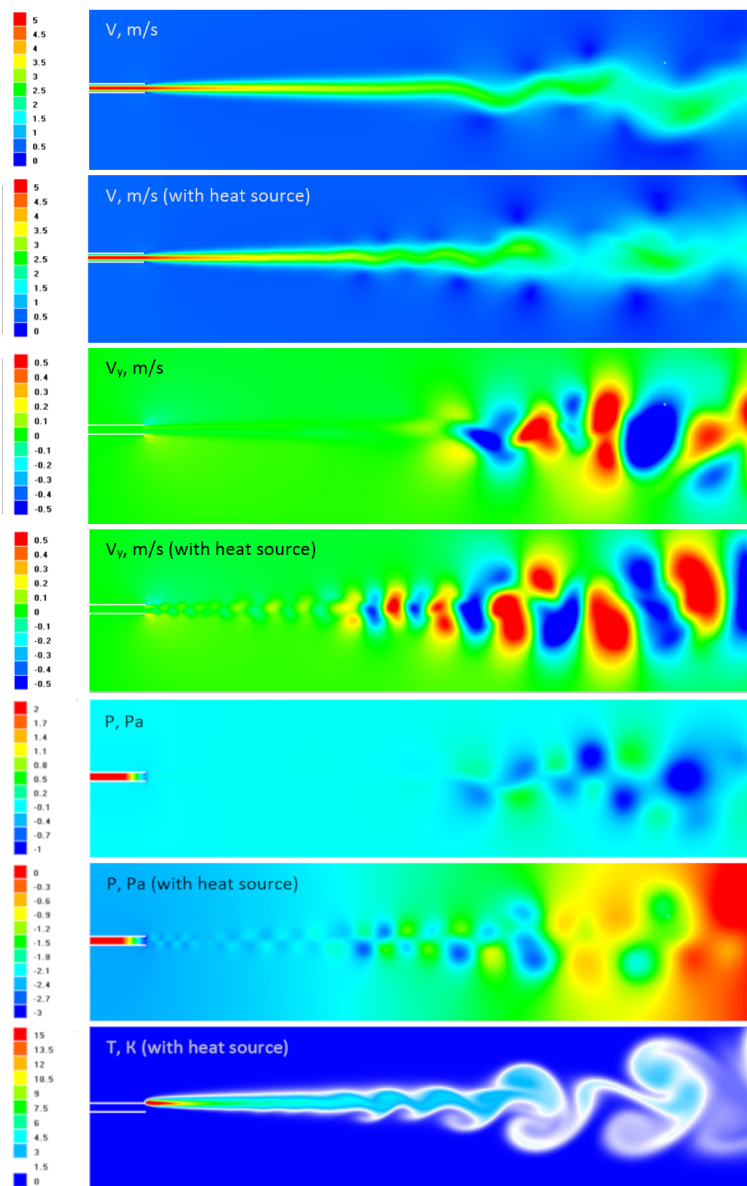


**Figure 3.** Time dependences of temperature and pressure caused by pulse-periodic heat source operation at  $f = 1000$  Hz

The pulse repetition rate was varied at a jet speed of 5 m/s. Frequencies of 500, 1000, 1190 and 1500 Hz were considered, the spectra of the resulting pressure pulsations are shown in Fig. 4. Thus, at a frequency of introduced disturbances of 500 Hz, pressure oscillations occurred with a frequency of 1500 Hz (see Fig. 4) – the second pressure pulse, corresponding to turning off the heat source, was in phase with the oscillations initiated by the first pulse, corresponding to turning on. As a result, this mode of operation led to the development and maintenance of jet oscillations. At a frequency of introduced disturbances of 1000 Hz, the second pressure pulse, on the contrary, was in antiphase to the initiated oscillations, which greatly affected the resulting spectrum of pressure oscillations (see Fig. 4). When pulsations were introduced at frequencies of 1190 and 1500 Hz, pressure oscillations occurred at the same frequencies. The simulation results for the case of excitation of a jet by a discharge with a frequency of 1500 Hz are presented in Fig. 5. It can be seen that in the absence of a discharge, the jet's own instability is observed at a considerable distance from the tube, and the inclusion of a discharge introduces disturbances in the zone closest to the tube. It is important to note that the frequency of pressure oscillations resulting from the response to a single pulse (switching on and keeping the heat source switched on) was 1380 Hz.



**Figure 4.** Spectra of pressure fluctuations in the jet with varying frequency of exposure and Pressure pulsations caused by a periodic heat source with a frequency of 1500 Hz

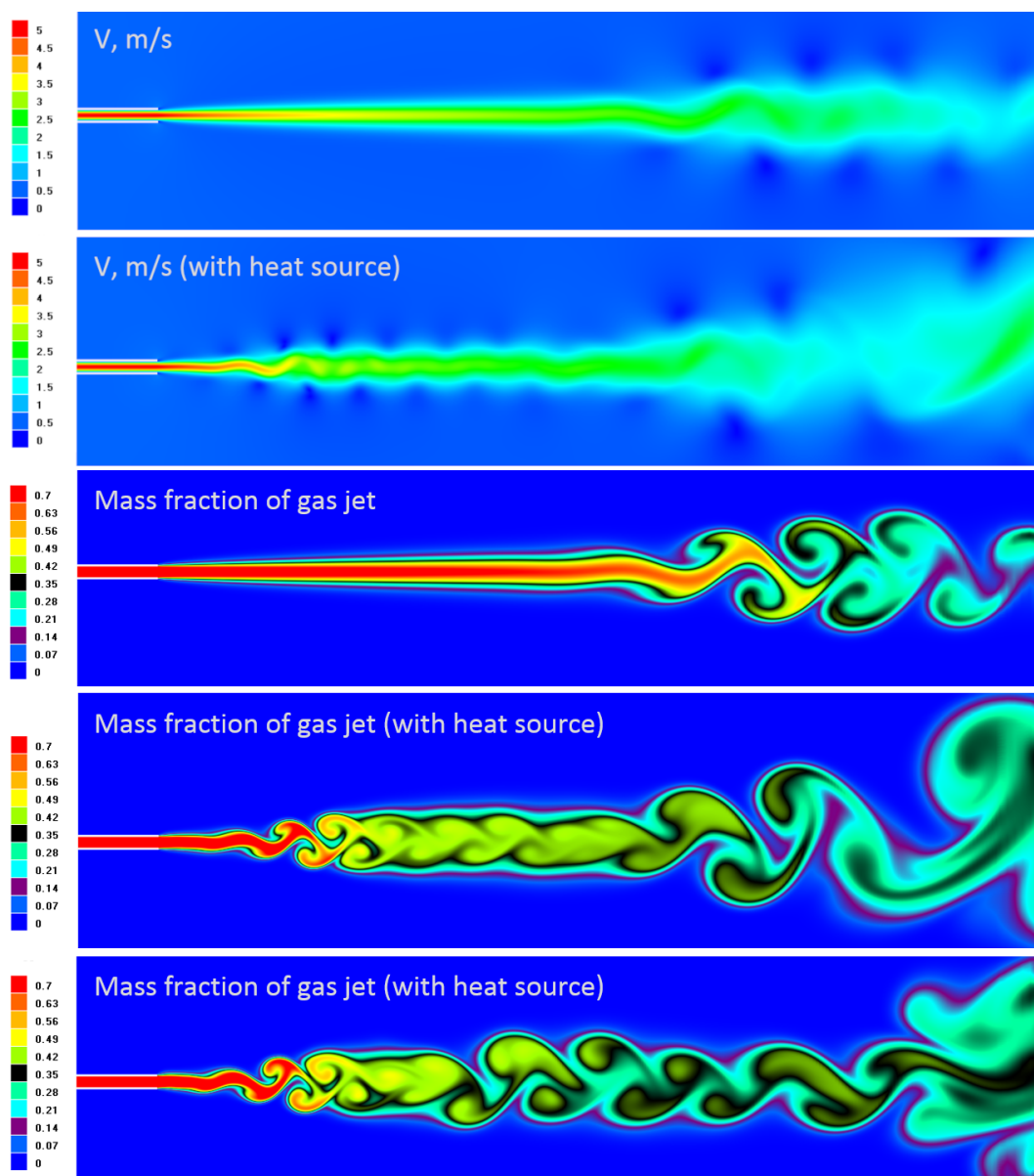


**Figure 5.** Results of simulation of jet instability at velocity  $V = 5$  m/s for case without and with excitation of a jet by a heat source with a frequency of 1500 Hz



#### 4. Results for Varying Velocity of the Discharge

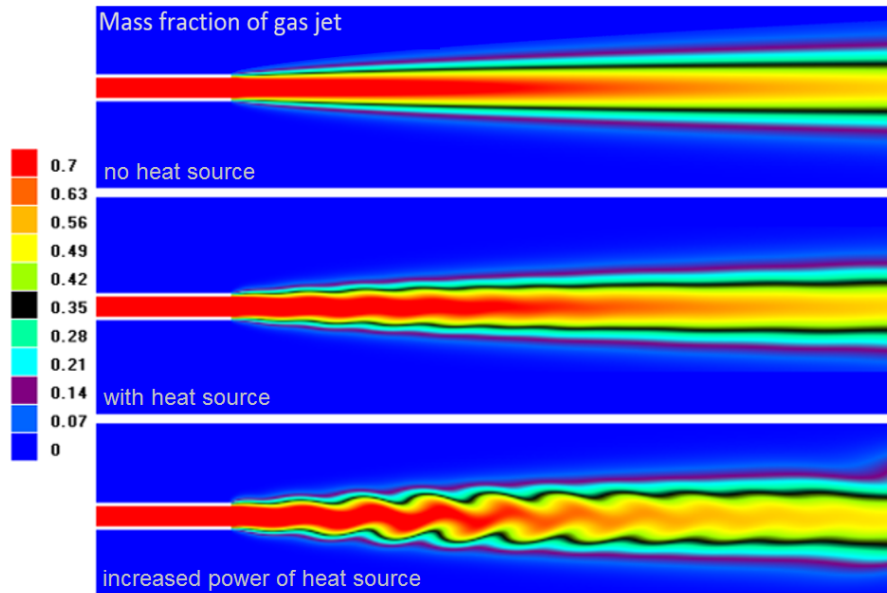
The velocity of the gas jet was varied, and for ease of visualization, the calculation was carried out using two identical gases (both with the properties of air). Velocities of 2.5, 5 and 10 m/s were considered. At a speed of 5 m/s, the jet, as mentioned earlier, has its own velocity pulsations at a considerable distance from the tube, which are clearly visualized using the mass fraction of the jet substance, similar to visualization in the experiment. Turning on the discharge (frequency 500 Hz) led to the development of jet instability starting from the edge of the tube. The gas in the discharge area was heated by +250 K, an increase in the energy input into the discharge, such that the maximum heating relative to the reference data was +640 K, briefly led to a change in the flow structure, but after a short time (70 ms) the flow pattern became close to that obtained with a lower energy input. Visualization of the jet flow and a typical instability pattern at 5 m/s are shown in Fig. 6.



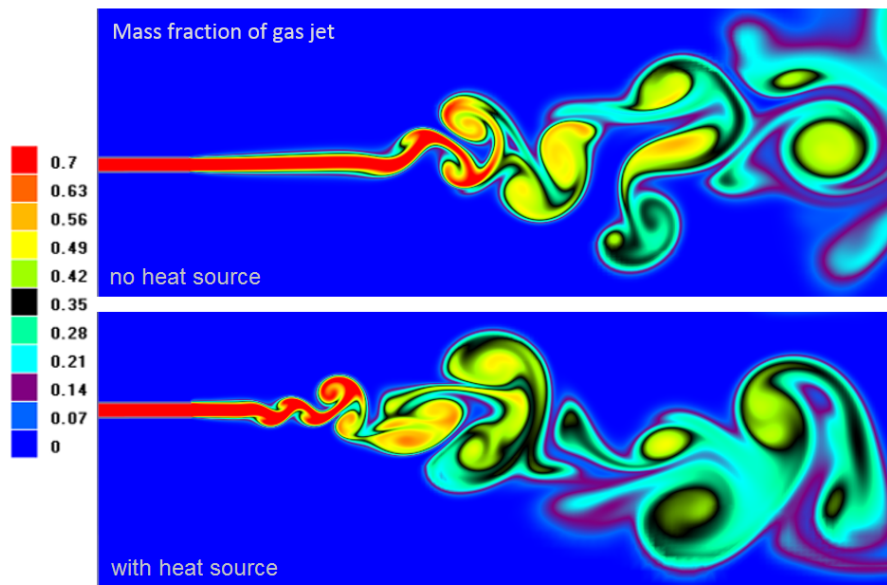
**Figure 6.** Results of simulation of jet instability at velocity  $V = 5$  m/s for case without and with excitation of a jet by a heat source with a frequency of 500 Hz. Bottom image shows a temporary change in the flow structure caused by an increase in energy input

Reducing the jet speed to 2.5 m/s leads to stabilization of the jet flow and the absence of its own instability at a distance limited by the computational area. The inclusion of a pulse-periodic heat source leads to the appearance of pressure pulsations and a slight distortion of the jet shape, which increases with increasing energy input into the discharge, described for 5 m/s. The simulation results for a 2.5 m/s jet are presented in Fig. 7.

A jet with a speed of 10 m/s is extremely unstable; significant pulsations of speed, pressure and distortion of the shape of the jet are observed already at a distance of about 10 mm from the tube. Turning on the discharge leads to significant pulsations starting from the very edge of the tube, as shown in Fig. 8.



**Figure 7.** Visualization of a jet without a discharge, with a discharge (heat source) and with increased energy input. Results of simulation at velocity  $V = 2.5$  m/s



**Figure 8.** Visualization of a jet without a discharge and with a discharge simulated by heat source. Results of simulation at velocity  $V = 10$  m/s

## Conclusion

The acoustic and pulse-periodic heat source excitation of a laminar gas micro-jet is simulated using the FlowVision software package. Using numerical modeling, it is shown that in a two-dimensional formulation with a jet width of 0.7 mm and velocities of 2.5, 5 and 10 m/s, it is possible to excite jet oscillations using a pulse-periodic heat source with a power of a unit of milliwatts (5–12 mW). In the case of a jet with a speed of 5 m/s, the natural oscillations of the jet in response to a single pulse had a frequency  $f_{res} = 1380$  Hz, so excitation of the jet was possible at close frequencies of 1190 Hz and 1500 Hz. At the same time, at a frequency of 1000 Hz (approximately equal to  $2/3 f_{res}$ ), the second pulse caused by turning off the heat source acted in antiphase with the pressure oscillations excited by the first pulse from turning on the heat source, and at this frequency the oscillations were initiated worse. At the same time, at a heat source frequency of 500 Hz (approximately equal to  $1/3 f_{res}$ ), the second pulse caused by turning off the heat source acted in phase with pressure oscillations excited by the first pulse from turning on the heat source, as a result of which the oscillations were excited no worse than at the main frequency of 1500 Hz. Dependence of flow structure from the jet velocity was obtained. It was shown that increase of jet velocity results in increase of natural oscillations and also disturbances of the jet excited by heat source. In all cases, at the heat source operation the jet oscillation zone moves closer to the tube. The results obtained show the possibility of exciting a micro-jet using low-power electrical discharges such as spark, DBD or corona.

## Acknowledgements

This work was performed within the State Assignment No. 075-01129-23-00 by the Ministry of Science and Higher Education of the Russian Federation.



*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. Reynolds, W.C., Parekh, D.E., Juvet, P.J.D., Lee, M.J.D.: Bifurcating And Blooming Jets. Annual Review of Fluid Mechanics 35(1), 295–315 (2003). <https://doi.org/10.1146/annurev.fluid.35.101101.161128>
2. Oh, J., Heo, P., Yoon, Y.: Acoustic excitation effect on NOx reduction and flame stability in a lifted non-premixed turbulent hydrogen jet with coaxial air. International Journal of Hydrogen Energy 34(18), 7851–7861 (2009). <https://doi.org/10.1016/j.ijhydene.2009.07.050>
3. Gau, C., Shen, C.H., Wang, Z.B.: Peculiar phenomenon of micro-free-jet flow. Physics of Fluids 21(9), 092001 (2009). <https://doi.org/10.1063/1.3224012>
4. Yoshida, H., Koda, M., Ooishi, Y., *et al.*: Super-mixing combustion enhanced by resonance between micro-shear layer and acoustic excitation. International Journal of Heat and Fluid Flow 22(3), 372–379 (2001). [https://doi.org/10.1016/S0142-727X\(01\)00103-5](https://doi.org/10.1016/S0142-727X(01)00103-5)

5. Krivokorytov, M.S., Golub, V.V., Volodin, V.V.: The effect of acoustic oscillations on diffusion combustion of methane. *Technical Physics Letters* 38(5), 478–480 (2012). <https://doi.org/10.1134/S1063785012050240>
6. Kozlov, G.V., Grek, G.R., Sorokin, A.M., Litvinenko, Yu.A.: Influence of initial conditions at the nozzle exit on the structure of round jet. *Thermophysics and Aeromechanics* 15(1), 55 (2008). <https://doi.org/10.1134/S0869864308010046>
7. Kozlov, V.V., Grek, G.R., Litvinenko, Y.A., *et al.*: Subsonic Round and Plane Jets in the Transversal Acoustic Field. *Siberian Journal of Physics* 5(2), 28–42 (2010). <https://doi.org/10.54362/1818-7919-2010-5-2-28-42>
8. Krivokorytov, M.S., Golub, V.V., Moralev, I.A.: The evolution of instabilities in gas microjets under acoustic action. *Technical Physics Letters* 39(9), 814–817 (2013). <https://doi.org/10.1134/S1063785013090186>
9. Kozlov, V.V., Grek, G.R., Dovgal, A.V., Litvinenko, Y.A.: Stability of Subsonic Jet Flows. *Journal of Flow Control, Measurement & Visualization* 01(03), 94–101 (2013). <https://doi.org/10.4236/jfcmv.2013.13012>
10. Firsov, A., Savelkin, K.V., Yarantsev, D.A., Leonov, S.B.: Plasma-enhanced mixing and flameholding in supersonic flow. *Philos. Trans. R. Soc. A* 373(2048), 20140337 (2015). <https://doi.org/10.1098/rsta.2014.0337>
11. Dolgov, E.V., Kolosov, N.S., Firsov, A.A.: The study of the discharge influence on mixing of gaseous fuel jet with the supersonic air flow. *Computer Research and Modeling* 11(5), 849–860 (2019). <https://doi.org/10.20537/2076-7633-2019-11-5-849-860>
12. Volkov, L.S., Firsov, A.A.: Modeling the influence of repetitively pulsed heating on the formation of perturbations at the boundary of a transverse jet in a supersonic crossflow. *Computer Research and Modeling* 15(4), 845–860 (2023). <https://doi.org/10.20537/2076-7633-2023-15-4-845-860>
13. Shevchenko, A.K., Yakovenko, S.N.: Simulation of Instability Development in a Plane Submerged Jet at Low Reynolds Numbers. *Siberian Journal of Physics* 13(4), 35–45 (2018). <https://doi.org/10.25205/2541-9447-2018-13-4-35-45>
14. Boedicker, C., Planeaux, J., Chen, T., Schmoll, W.: Interactions between jet and annulus flows in the presence of a two-dimensional bluff body. *1st National Fluid Dynamics Conference* (1988). <https://doi.org/10.2514/6.1988-3689>
15. Roquemore, W.M., Britton, R.L., Tankin, R.S., *et al.*: Interaction of 2-D Wake and Jet Plume. *The Physics of Fluids* 31(9), 2385–2385 (1988). <https://doi.org/10.1063/1.4738820>

# String-Wave Direct Parallel Solver for Sparse System of Linear Equations

Alexey Y. Likhoshesterov<sup>1,2</sup> , Yana G. Velikaya<sup>2</sup> 

© The Authors 2023. This paper is published with open access at SuperFri.org

The article discusses a parallel algorithm of solving linear algebraic equations systems for symmetric sparse matrices, which allows to split a large task into many small subtasks, thereby both increasing performance and reducing memory consumption. It is based on a method of simultaneous calculation of intermediate values during matrix factorization with maintaining load balancing on processors so that when the final result of the left parts of the factorization is obtained, the right parts of the factorization do not depend on them. This approach allows the initial stiffness matrix to be represented as a product of a large number of simple matrixes and solve a system of linear algebraic equations in the form of a sequence of solutions by substitution. To reduce the filling of sparse factorization matrices, an approximate minimum degree method was used, which, in addition to being one of the most efficient and fastest ones existing at the moment, allows the developed algorithm to distribute the load of calculations more evenly. The developed method is implemented in APM Ltd. software products for systems with shared memory, but it can also be performed for distributed memory systems.

*Keywords:* approximate minimum degree, string-wave algorithm, finite element method, system of linear equations, matrix factorization.

## Introduction

Currently, the finite element method (FEM) is actively used for engineering calculations in the field of construction and machine-building [1]. The use of FEM involves solving systems of linear algebraic equations of large dimension. The matrices of such systems, called stiffness matrixes, have a number of properties that depend on the structure and number of nodes of the finite element grid, as well as the nature of the problem. These matrices are symmetric and sparse to a considerable extent.

With the help of the system of linear algebraic equations, such tasks as: static, dynamic calculation, calculation of natural frequencies, stability, nonlinear calculations, etc. are solved. In general, system of linear algebraic equations has the following form [2].

$$K_{n,n}X_{n,m} = F_{n,m}, \quad (1)$$

where  $K$  – is the stiffness matrix,  $X$  – are displacement vectors,  $F$  – are load vectors for a finite element system,  $n$  – is the dimension of the stiffness matrix;  $m$  – is the number of loads.

Due to the fact that the stiffness matrix is sparse, it is presented in the CSR (Compressed Sparse Rows) [3] string sparse form, where the storage structure uses three one-dimensional arrays: 1) an array of non-zero elements of the stiffness matrix line by line; 2) an array of column numbers of non-zero elements line by line; 3) an array of the location of the first non-zero element in each row. This format allows to reduce the amount of source data and decrease the number of operations performed on them significantly, i.e., when calculating, memory consumption is reduced and performance increases.

The article is organized as follows. Section 1 is devoted to the problem of factorization of symmetric sparse matrices and the need to reorder them. In section 2 we describe the main

<sup>1</sup>APM Ltd., Korolev, Russian Federation

<sup>2</sup>Belgorod State University, Belgorod, Russian Federation

methods for reordering sparse matrices and compares the performance of the AMD and Cuthill–McKee algorithms. Section 3 contains a description of the developed string-wave algorithm for solving system of linear algebraic equations and computational experiments of its application. Conclusion summarizes the study and points directions for further work.

## 1. Matrix Factorization

Today, two different approaches to solving systems of linear algebraic equations are used: direct solution and iterative methods. Each approach has certain advantages, but direct methods are of the most common use. The main problem of iterative methods is the frequent poor conditionality of stiffness matrices of large dimension when its determinant is equal to a small value or tends to zero. And even the preconditioning process does not absolutely guarantee the convergence of the solution. In order to simplify the solution of system of linear algebraic equations with a symmetric stiffness matrix, the following factorization is used [4].

$$K_{n,n} = L_{n,n}D_{n,n}L_{n,n}^T, \quad (2)$$

where  $L$  – is the lower triangular matrix (stored in sparse format), in which all diagonal elements are equal to one,  $D$  – is the diagonal matrix (stored as a one-dimensional array).

The solution of system of linear algebraic equations  $LDL^T X = F$  in this case is reduced to the sequential solution of three simplest issues by the substitution method.

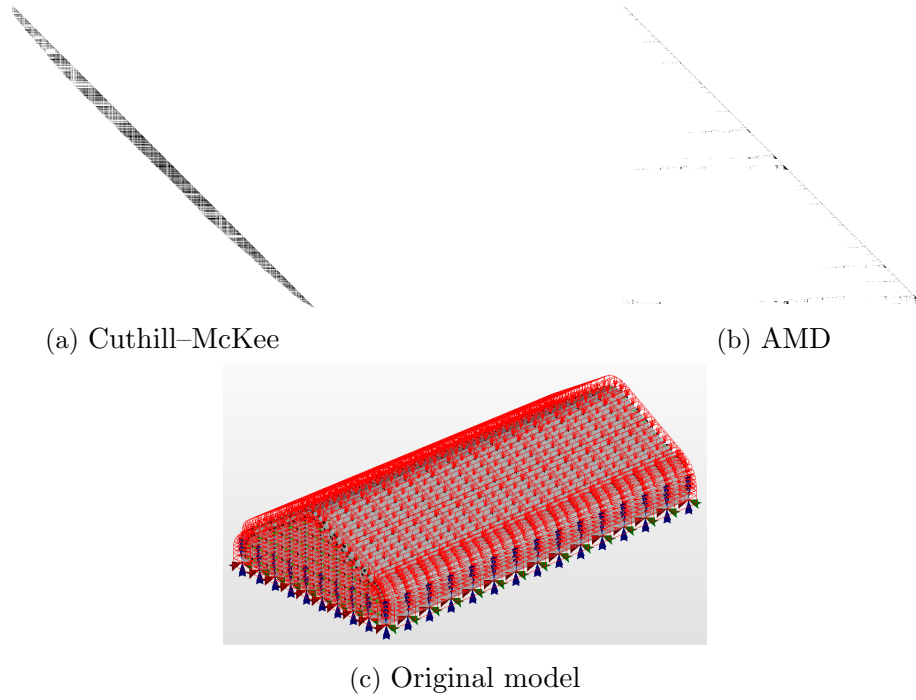
$$\begin{cases} L_{n,n}Y_{n,m} = F_{n,m}; \\ D_{n,n}Z_{n,m} = Y_{n,m}; \\ L_{n,n}^T X_{n,m} = Z_{n,m}. \end{cases} \quad (3)$$

The main problem with this factorization is filling a sparse triangular matrix  $L$  with a large number of non-zero elements, which entails significant memory consumption and an increase in the number of computational operations on the elements of this matrix. Therefore, before the factorization procedure, the rows and columns of the original stiffness matrix are reordered. To do this, we find a special permutation matrix  $P$  such that

$$(P_{n,n}K_{n,n}P_{n,n}^T)(P_{n,n}X_{n,m}) = P_{n,n}F_{n,m}. \quad (4)$$

## 2. Permutation of Stiffness Matrix

The permutation matrix  $P$  is a matrix in which each row and column contains only one element equal to one, and the others equal to zero. This matrix is stored as an array of column numbers with single elements. To find the permutation matrix, the Cuthill–McKee algorithm is often used due to the simplicity of its implementation, but such an algorithm is not optimal from the point of reducing the filling of factorization matrices with nonzero elements. Algorithms based on the application of the minimum degree method are more effective. In practice, this method, in its original form, is not used due to its considerable complexity. Two of its modifications are widespread: the Multiple Minimum Degree method (MMD) and the Approximate Minimum Degree method (AMD) [5]. The latter is used in the software products of the company APM Ltd. Figure 1 shows the result of calculating the lower triangular matrix  $L$  with permutation using the Cuthill–McKee algorithm and the approximate minimum degree method.



**Figure 1.** The result of factorization after reordering the matrix

Here the black dots denote the nonzero elements of the resulting factorization matrix  $L$ . Table 1 shows the number of received non-zero elements of this matrix.

**Table 1.** Results of matrix reordering algorithms

Initial stiffness matrix	Cuthill-McKee	AMD
279966	4510257	469879

After the permutation procedure, the factorization of the stiffness matrix is performed directly. In the classical version,  $LDL^T$  factorization calculated as follows

$$\begin{cases} D_j = K_{jj} - \sum_{k=1}^{j-1} L_{jk} D_k L_{jk}^T; \\ L_{ij} = \frac{K_{ij} - \sum_{k=1}^{j-1} L_{ik} D_k L_{jk}^T}{D_j}; \\ i = 1, \dots, n; j = 1, \dots, n. \end{cases} \quad (5)$$

### 3. Description of the Algorithm

Expression (5) does not lend itself well to parallelization due to the large number of information dependencies. In addition, the calculations are not balanced, which significantly reduces their efficiency [6]. To eliminate these problems, an algorithm adapted for parallel computing was developed. The triangular matrix of factorization  $L$  can be visually represented as a musical instrument like a harp (Fig. 2). Each harp string corresponds to a column of the matrix  $L$ . The strings are played alternately. Playing on the  $j$ -th string means finding the final values of the elements of the  $j$ -th column  $L$ . At the same time, each string propagates an undamped sound wave towards the strings located to the right of it. That is, each  $j$ -th column of the matrix  $L$

partially changes the values of the elements of all columns with a number higher than  $j$ . Thus, it is possible to calculate the effect of the current column on all columns located to the right in parallel.

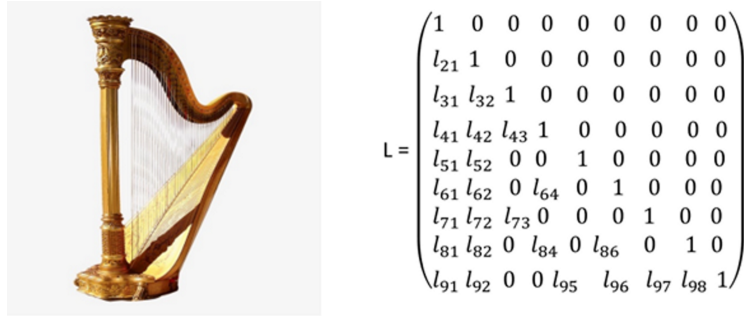


Figure 2. Visual representation of the factorization matrix in the form of a harp

Mathematically, the string-wave direct solver can be described as follows. First, the view matrices are assigned the corresponding values of the stiffness matrix.

$$\begin{cases} D_{ii} = K_{ii}; \\ L_{ij} = K_{ij}; \\ i = 1, \dots, i - 1; j = 1, \dots, i - 1. \end{cases} \quad (6)$$

Further, the elements of these matrices are changed with accordance to the algorithm described above.

$$\begin{cases} L_{ij} = \frac{L_{ij}}{D_{jj}}; \\ L_{ik} = L_{ik} - L_{kj}D_{jj}L_{ij}; \\ D_{ii} = D_{ii} - L_{ij}D_{jj}L_{ij}; \\ j = 1, \dots, n; i = j + 1, \dots, n; k = 2, \dots, i - 1. \end{cases} \quad (7)$$

Since the factorization matrix  $L$  is sparse, not all right columns are affected by the current one. In the course of the conducted research, it was revealed that only those columns which number corresponds to the row numbers of non-zero elements of the current column change. This circumstance makes it possible to eliminate unnecessary iterations of the parallel algorithm cycle. When solving system of linear algebraic equations of large dimensions by a direct method, there is often not enough memory. The string-wave algorithm allows to reduce the size of the problem, for example, by applying the following factorization.

$$K = L_1 B L_1^T, \quad (8)$$

where  $L_1$  are the first columns of the triangular factorization matrix  $L$ ,  $B$  is a submatrix of smaller dimension obtained by the influence of  $L_1$  (Fig. 3). In turn, a similar factorization can

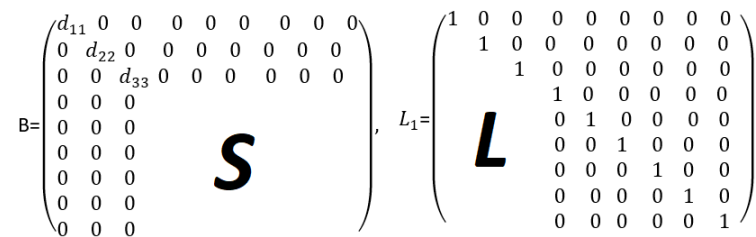


Figure 3. Reducing the dimension of the problem



be applied to the matrix  $B$ . As a result, the string-wave direct solver allows us to obtain the following expression

$$K = L_1 L_2 \dots L_{n-1} L_n D L_n^T L_{n-1}^T \dots L_2^T L_1^T. \quad (9)$$

This factorization of the stiffness matrix allows to split a large problem into many small subtasks, where the solution of system of linear algebraic equations is also found by substitution

$$\left\{ \begin{array}{l} L_1 X_n = F; \\ L_2 X_{n-1} = X_n; \\ \dots\dots\dots \\ L_n X_{i-1} = X_{i-2}; \\ DX_i = X_{i-1}; \\ L_n^T X_{i+1} = X_i; \\ \dots\dots\dots \\ L_2^T X_2 = X_3; \\ L_1^T X_1 = X_2. \end{array} \right. \quad (10)$$

Speedup and efficiency parameters were used to evaluate the efficiency of parallel computing [7]. Speedup refers to the execution time ratio of a sequential algorithm to the execution time of a parallel algorithm on  $p$  processors

$$S_p = \frac{T_1}{T_p}. \quad (11)$$

The efficiency of using processors by a parallel algorithm in solving a problem is defined as the ratio of acceleration to the number of processors

$$E_p = \frac{S_p}{p}. \quad (12)$$

The results of computational experiments evaluating the effectiveness of the parallel algorithm are presented in Tab. 2. The algorithm is implemented using OpenMP [8]. Calculations were performed on a six-core Intel Core i7-8700K processor with 32 GB of DDR4 RAM. To conduct computational experiments, the APM Structure3D module of APM WinMachine software was used, in which models of various dimensions were loaded and when measuring the execution time of the parallel algorithm, the time for the formation of the stiffness matrix was taken into account.

**Table 2.** Evaluating the effectiveness of a parallel algorithm

Dimension of the stiffness matrix, $n$	Speedup, $S_p$	Efficiency, $E_p$
94,692	3.54	0.59
174,267	3.75	0.63
244,620	4.86	0.81
527,919	4.72	0.79
755,181	4.78	0.80

Thereby, the string-wave direct solver effectively works in conjunction with the AMD re-ordering algorithm, breaking a large task into many small subtasks with the least information dependencies, which allows its parallel implementation to achieve optimal values of speedup and efficiency parameters.

## Conclusion

The article presents a new algorithm for solving systems of linear algebraic equations for symmetric sparse matrices. This algorithm includes the AMD method for reordering sparse matrices and the string-wave method for solving system of linear algebraic equations of our own design. It is successfully used in software products of the company APM Ltd. The developed solver is used in such software modules as static and dynamic calculations, calculation of natural frequencies and stability when solving a generalized eigenvalue problem, calculation of harmonic vibrations and many others.

The developed algorithm is parallel and implemented using OpenMP technology. Computational experiments show that with an increase in the number of processors and the volume of input data, the speedup increases without a decrease in efficiency. That is the developed parallel algorithm is scalable.








Further research will focus on improving its performance and implementation using CUDA technology for NVIDIA graphics accelerators and MPI technology for distributed memory systems.

*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. Bathe, K.J., Wilson, E.L.: Numerical methods in finite element analysis. Stroyizdat, Moscow (1982).
2. Demmel, J.: Computational linear algebra. Theory and applications. Word, Moscow (2001).
3. Pissanetsky, S.: Sparse Matrix Technology. Academic Press Inc (1984).
4. Andreev, B.: Numerical methods. Publishing department of the Faculty of Computational Mathematics and Mathematics of Lomonosov Moscow State University, Moscow (2013).
5. Heggenes, P., Eisenstat, S.C., Kumfert, G.: The Computational Complexity of the Minimum Degree Algorithm. In: NASA/CR-2001-211421ICASE Report No. 2001-4, Langley Research Center, Hampton, Virginia (2001). <https://doi.org/10.2172/15002765>
6. Voevodin, V., Voevodin, Vl.: Parallel Computing. BHV, Saint Petersburg (2002).
7. Gergel, V.P.: High-performance computing for multiprocessor, multi-core systems. Moscow University Publishing House, Moscow (2010).
8. Gergel, V.P.: Modern languages and parallel programming technologies. Moscow University Publishing House, Moscow (2012).

# Three-dimensional Numerical Model of Kerosene Evaporation in Gas Turbine Combustors

*Andrey A. Aksenov*<sup>1</sup> , *Sergey V. Zhluktov*<sup>1</sup> , *Vladimir S. Kashirin*<sup>1</sup> ,  
*Marina L. Sazonova*<sup>1</sup> , *Sergey G. Cherny*<sup>2</sup> , *Ilia V. Zeziulin*<sup>3</sup> ,  
*Maria D. Kalugina*<sup>1</sup> 

© The Authors 2023. This paper is published with open access at SuperFri.org

A three-dimensional model of the multiphase flow based on the Eulerian–Eulerian approach was implemented using the FlowVision CFD package and, on this basis, a numerical algorithm for study of evaporation of liquid fuel was developed. The created high-performance complex for the carrier and dispersed phases interaction simulation was validated against the well-studied experimental problem of the evaporation and mixing of kerosene emerging from a flat pre-filming airblast atomizer for gas turbine combustors. In this work, the carrier phase is supposed to be air and kerosene vapors, and the dispersed phase is selected as liquid kerosene. Based on the calculated kerosene evaporation drops distributions, an important parameter that characterizes the spray fineness, Sauter mean diameter, is determined. Numerically calculated in the developed model the evaporation rate and Sauter mean diameter of fuel droplets agreed well with the experimental data. In famous works, the air temperature and pressure varied during the experiments. At the same time, in comparison with the calculated data, a stronger influence on the kerosene evaporation was obtained by air temperature than pressure. The dependence on pressure can be seen in the case of taking into account the corresponding changes in the liquid fuel properties. It is also noted that the initial fuel temperature is an important parameter for evaporation. This can be seen in the results of the kerosene evaporation numerical simulation carried out in this study.

*Keywords:* mathematical modeling, FlowVision, verification, dispersed phase, Eulerian–Eulerian approach, liquid fuel, droplets evaporation, Sauter mean diameter.

## Introduction

1) *Numerical modeling is the most appropriate tool to study liquid fuel evaporation.* In gas turbine design, it is often necessary to develop devices for the atomization, evaporation, and mixing of liquid fuel within a given fuel droplet size range and uniformity of droplet distribution in the gas stream. At the same time, as a rule, a detailed experimental study of atomization and evaporation using real nozzles is difficult due to the wide range of particle sizes and the density of filling the gas stream with fuel particles. Therefore, the results of the evaporation and kerosene mixing detailed study under conditions similar to aircraft engines and stationary gas turbines demonstrated in [7] are of particular interest.

2) *The experimental data is for validation of numerical modeling.* One of the reasons for the experimental data given in [7] significance is their potential for validation (debugging, calibration, calculation methods development) of programs for fuel evaporation process numerical modeling. Numerical modeling is the most appropriate way (tools) to understand (comprehend) the flow structure and identify the factors that affect the fuel atomization and evaporation efficiency, and, therefore, the gas turbine energy conversion efficiency. For the evaluation of the mixing quality in the combustion chambers of aircraft and rocket engines, numerical modeling of the droplet jet formation is currently used to determine the fineness of atomization by the calculation of

---

<sup>1</sup>“TESIS” LTD, Moscow, Russia

<sup>2</sup>Kutateladze Institute of Thermophysics of the Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia

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

Sauter mean diameter, taking into account primary and secondary decay, fusion and collision of droplets, which makes it possible to evaluate liquids atomization in a wide range of devices and modes. In turn, the correct evaluation of the fuel atomization is the fundament for a correct (detailed) description of its evaporation. This knowledge is very much in demand for device design. Therefore, at the stages of design, modernization, and optimization of these installations' operating modes, it is impossible to do without mathematical modeling of the processes occurring in these installations. The reliability of the results obtained in this case can only be guaranteed by the validation of the numerical model and its software implementation carried out on reliable experimental data. Validation of an analytical technique is experimental proof that the technique is suitable for solving the intended problems.

3) *FlowVision CFD is validated on experimental data.* In [7], a model of a pre-evaporator channel with full optical access operating at a pressure of up to 15 bar and a temperature of up to 850 K was created to conduct experiments on kerosene evaporation. During the experiment, the pressure, temperature, and airflow velocity, as well as the initial jet temperature and fuel consumption, were changed. It means there are unique data for the calibration of complex three-dimensional numerical models of kerosene atomization, evaporation, and mixing under conditions similar to aircraft engines and gas turbines. In this study, the FlowVision software package (FlowVision PC) is validated [3, 4] based on the unique experimental data from [7] in terms of the reliability of the results obtained on its basis for solving phase transition problems.

4) *Effective technique for numerical simulation fuel evaporation is of interest.* Even though a large number of works have been carried out on experimental and numerical studies of the processes of liquid evaporation in gas and the dependencies of the quality of gas, fuel and its evaporation products on the main parameters of fuel input have been compiled, interest in a more thorough study of the vaporization process is constantly increasing. New models are being developed, validated, and verified for both primary and secondary decay and droplet fusion, as well as evaporation and fuel combustion. The works considered below are the closest in terms of the studied processes and research methods. In [8], an effective technique for numerical simulation of a liquid jet decay in a compressed gas stream based on the Reynolds equations and the volume of fluid (VOF) method was developed. The technique describes the formation of primary droplets, and their transition to the Eulerian dispersed phase, where they undergo the secondary break-up and evaporation. In [11], Eulerian multicomponent model of liquid atomization and evaporation based on the WAVE decay model is presented. Special attention is paid to the modeling of thermal processes occurring inside moving droplets. A successful simulation of the dense liquid jets complex dynamic behavior has been carried out. In [9, 10], the large eddy simulation (LES) was used to describe the spray. In [9] a special attention is paid to the setting of turbulent boundary conditions. The comparison of the results obtained experimentally and numerically by the proposed model showed its good accuracy. A similar study was carried out in [10], which also investigated the effect of the mesh resolution on the spray global characteristics such as the penetration depth. In [12], direct numerical simulation (DNS) was applied to describe the sputtering of the flame of n-heptane in a vortex combustion chamber. Using the model, the contribution of the combustion of the preliminary mixture and the diffusion flame to the total heat release is estimated. Studies [13, 14] are devoted to the research of the fuel spray evolution in the engines combustion chambers. The CFD analysis carried out in [13] turned out to be the only chance to get a complete view on the fuel spray process, which made it possible to design the optimal geometry of the chamber. In [14], calculations were carried out for chambers

of various working volumes; it is noted that the simulation results show a good agreement with experimental data. Works by S. Tonini et al. [21, 22] are also devoted to modeling the spraying of dense diesel fuel compressed to high pressure – in [21] a description of the model, the approaches used, validation of the model is presented, and in [22] the influence of the number of fuel injections and the time between injections on the development of a cloud of droplets in the working chamber is studied. Several papers [18, 19] are devoted to the study of the flow parameters inside the nozzle influence on the fuel atomization in the working chamber. In [18], the effect of primary droplet decay on the characteristics of atomization and fuel combustion in diesel engines was studied. In particular, the KH-ACT model is considered, taking into account the contributions of cavitation and turbulence inside the nozzle to the droplets primary break-up. The results show improved agreement with the experiments in comparison with modeling using the KH model. In [19], the Eulerian model based on the VOF method and implemented in the CONVERGE CFD code was investigated. For the model, the influence of the nozzle outlet geometry on both the flow inside the nozzle and the further spray in the chamber is studied. The influence of the hole geometry on the droplets primary break-up of was established, which, in turn, affects further processes and characteristics, such as the fuel penetration depth into the working area, fuel evaporation, its mixing with air, and the spray fineness. A hybrid procedure for numerical simulation of two-phase flows in the combustion chamber is proposed in [17]. It is a two-stage combination of Eulerian and Lagrangian approaches. The Euler method gives a preliminary two-phase flow field. The Lagrangian method tracks the particle movement with a detailed description of the droplets secondary break-up. A 30% calculation of the spraying acceleration has been achieved compared to modeling based only on the Lagrangian method.

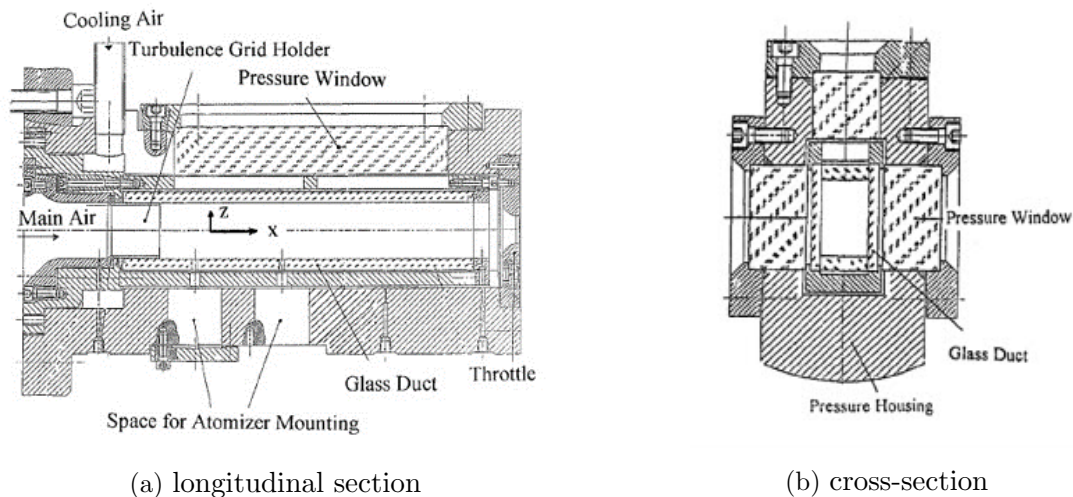
5) *The work is aimed at the fuel evaporation calculation and validation.* It should be noted that the works given in our brief review, as well as the works [3, 4], describe methods for calculating both atomization and evaporation of the dispersed phase. The FlowVision PC has a model of particle atomization during the secondary droplet decay and, in general, the secondary decay can be calculated in the package. However, the present work is aimed at the results of the fuel evaporation calculation validation. Therefore, the characteristics (parameters) of the fuel spray are taken from [7] for the calculation of the experimental setup [7]. At the same time, the possibility of varying the kerosene droplet diameter spectrum set at the chamber inlet is presented in the formulation of the problem. Analyzing the solution sensitivity to the spectrum, the range of changes in the diameters of kerosene droplets supplied to the chamber is selected, which ensures the evaporation rate and Sauter mean diameter of fuel droplets as much as possible coinciding with their values obtained experimentally. The proof of the FlowVision PC suitability for solving the vaporization of the dispersed phase problem is relevant for numerical studies presented in [3, 4] since the correctness of the carbon black production reactor operation evaluation obtained in these works significantly depends on this.

6) *The article is organized as follows.* Section 1 is described a problem statement as experimental setup and assumptions that we used for mathematical model. Section 2 contains description of mathematical model into FlowVision CFD. That model contains continues and dispersed phased that are described by different set of equations. In section 3 we showing results of numerical modeling in comparison of physical experiment. Several hypotheses were tested in additional to the comparison of mathematical model and experiment. Conclusion summarizes the study and results from section 3.

## 1. Problem Statement

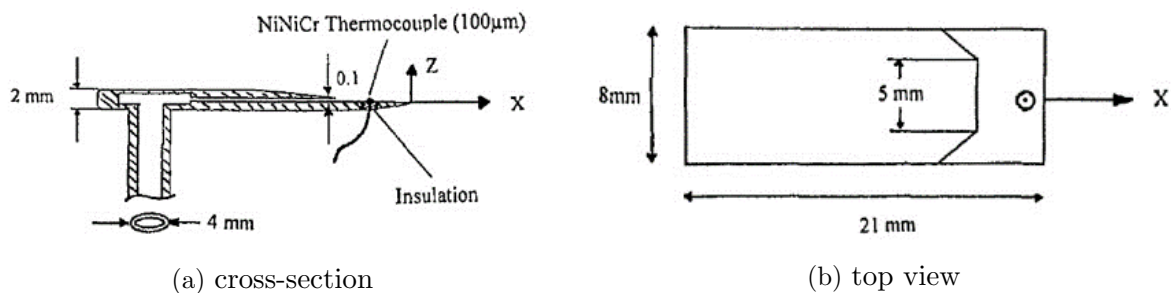
### 1.1. Experimental Setup

The experimental setup from [7] is shown in Fig. 1. The experimental setup working section is a rectangular duct with a cross-section of 25 mm × 40 mm and an observable streamwise pathlength of about 180 mm. To prevent overheating of the pressure housing, the quartz glass duct was surrounded by a cooling airflow. At a given temperature of the main air, the pressure inside the duct was regulated by changing the air flux and using throttles of different diameters at the duct outlet.



**Figure 1.** The experimental setup [7]

The setup allows to install a liquid fuel atomizer inside the channel in two axial positions. The atomizer shape is shown in Fig. 2. There was used an atomizer with a flat film section. The film was atomized by a high-speed incoming airflow. The atomizer width is 8 mm, the slit height is 0.1 mm. The atomizer material is stainless steel. The atomizer film section coincides with the horizontal symmetry plane of the duct XY. The origin of the coordinate system is located in the middle of the atomizer edge. To measure the fuel temperature, a thermocouple was installed on the atomizer film section surface. To protect against the atomizer housing thermal conductivity, the thermocouple was insulated with a ceramic bond. More than 70% of the sensor volume was located in the insulator so that the film flow on the atomizer surface was not disturbed by the sensor.



**Figure 2.** The airblast atomizer

Jet A1 kerosene, whose physical properties are of kerosene itself and its vapors, presented in [15], was used as a liquid fuel during the experiment. The parameters of the experimental mode, defined as the basic ones for numerical simulation, are given in Tab. 1.

**Table 1.** Experimental mode parameters defined as the basic for the numerical simulation

Physical property	Value
Air pressure	9 bar
Air temperature	750 K
Air speed (ratio of volumetric airflow to duct cross-sectional area)	120 m/s
Liquid fuel temperature (measured on the atomizer surface)	453 K
Fuel mass flow	1 g/s (125 g/s/m per unit length of the atomizer edge)

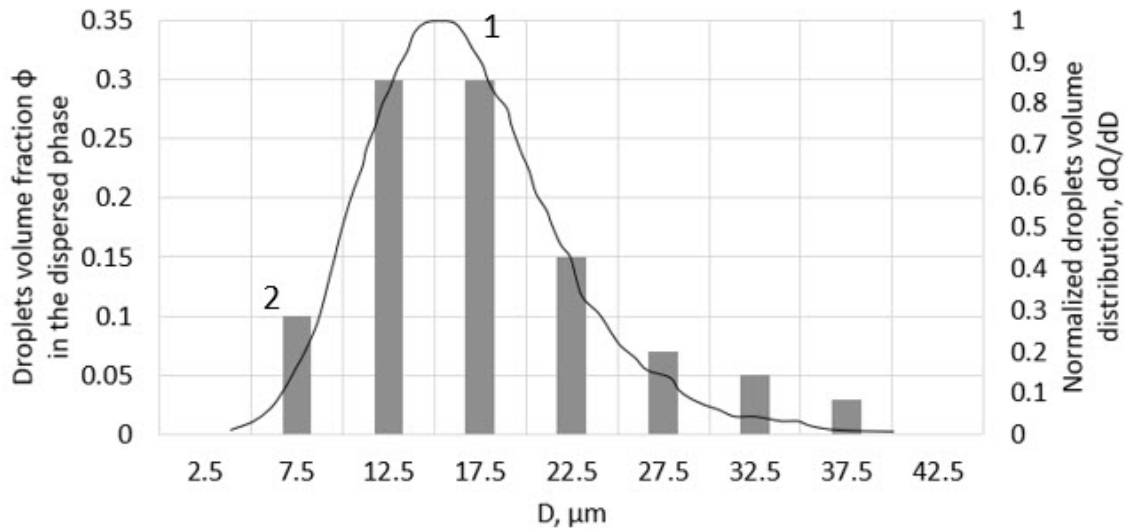
The control sections of the computational domain at which the flow characteristics are analyzed are located at distances from the sprayer  $x = 30$  mm, 60 mm, 100 mm, 150 mm. The computational domain is sectioned and flow characteristics are analyzed at distances of  $x = 30$  mm, 60 mm, 100 mm, 150 mm from the sprayer. The fuel evaporation rate  $V_{evap}/V_0$  (the ratio of the difference in the fuel droplets volume flows in the input section  $x_0$  and control sections  $x_k$  to the flow in  $x_0$ ) and droplets Sauter mean diameter are taken as control flow characteristics (hereinafter referred to as “control parameters”).

## 1.2. Numerical Model Assumptions

As noted in the introduction, atomization of the film is not considered in the computational model. A discrete distribution of droplet diameters by size – a droplet size spectrum – is set at the chamber inlet (the atomizer exit). The fuel inlet is established as the zero  $x$ -coordinate (atomizer edge), though according to experimental materials it is noted that the film primary atomization is completed at the distance of 10 mm from the atomizer. The velocity of the dispersed phase (droplets)  $V_d$  at the fuel inlet is assumed to be 10 m/s based on the calculation data in [7]. The fuel phase volume fraction at the inlet is determined by taking into account the selected speed and the experimental mass flow rate (Tab. 1). The distribution function  $\frac{dQ}{dD}$  of the volume occupied by droplets with diameter  $D$  and normalized to its maximum value  $\frac{dQ}{dD}$  is taken from the results of the “cold” experiment in [7]. This distribution function is shown in Fig. 3 as a solid line, and its values are shown on the right.

According to the given curve, a discrete distribution of the droplet volume fraction  $\phi$  is calculated. To do this,  $N_d$  (the number of droplets with a diameter  $d_i$  in the diameter spectrum) is determined, for which the value  $\frac{dQ}{dD}(d_i)$  is calculated. After that, the volume fraction  $\phi_i$  of droplets with a diameter  $d_i$  in the dispersed phase is calculated

$$\phi_i = \frac{\frac{dQ}{dD}(d_i)}{\sum_{i=1}^{N_d} \frac{dQ}{dD}(d_i)}. \quad (1)$$



**Figure 3.** Droplets diameter spectrum at the chamber entrance: 1 – diameter distribution function obtained in the “cold” experiment [7], 2 – selected for FlowVision calculation values

Expression (1) ensures the fulfillment of the condition  $\sum_{i=1}^{N_d} \phi_i = 1$ . Volume fraction values and the corresponding droplet diameters of the fuel droplet size spectrum at the inlet, taken as the basic, are shown in Tab. 2. Also, a histogram in Fig. 3 shows the discrete droplet diameter distribution, which was selected for calculation in FlowVision, these values are shown on the left. At the fuel (dispersed phase) inlet, its volume fraction  $\alpha$  is set. At a given kerosene velocity at the atomizer exit (fuel inlet)  $V_d = 10$  m/s, the volume fraction  $\alpha$  is calculated from experimental data (Tab. 1) as

$$\alpha = \frac{Q_d}{\rho(T_d) \cdot V_d \cdot S},$$

where  $Q_d$  is the kerosene mass flux,  $S$  is the atomizer exit area. The droplet concentration value for each family (set)  $n_{d,i}$  at the fuel inlet is determined for the concentration equation by  $\alpha$ , as well as by the distribution  $\phi_i$  depending on  $d_i$ :

$$n_{d,i} = \alpha \cdot \frac{\phi_i(d_i)}{V(d_i)}. \quad (2)$$

Here  $V(d_i)$  is the drop of diameter  $d_i$  volume. It is assumed in this model that all drops are balls of diameter  $d_i$ . The equations given in paragraphs 2.1–2.2 are solved separately for each particles family  $i$ . It should be noted that in the case of solving this problem based on the Eulerian–Lagrangian approach, the number of fuel droplets  $N_i$  passed through the atomizer exit per unit of time in each family (set) corresponding to the current diameter  $d_i$  of a given fuel droplet size spectrum may be calculated from (2) as

$$\frac{dN_i}{dt} = n_i \cdot V_d \cdot S = \alpha \cdot \frac{\phi_i(d_i)}{V(d_i)} \cdot V_d \cdot S. \quad (3)$$

The inlet gas velocity is selected based on preliminary calculations of the carrier phase without taking into account the dispersed phase results at a distance of 10 mm from the atomizer near its film section level. Due to the lack of experimental data on the effect on the problem solution changes of:



**Table 2.** Droplet diameter spectrum at the chamber entrance

Family number, $i$	1	2	3	4	5	6	7
Family particle diameter $d_i$ , $\mu\text{m}$	7.5	12.5	17.5	22.5	27.5	32.5	37.5
Family volume fraction in the dispersed phase $\phi_i$	0.1	0.3	0.3	0.15	0.07	0.05	0.03

- fuel velocity in the flow after the film is torn off the edge of the atomizer;
- droplet size distribution (diameter) after the film atomization;
- degree of turbulence in the incoming airflow;

the accepted assumptions are verified in this study by investigation of the sensitivity of the problem solution to change:

- droplet velocity at the fuel inlet;
- droplet size range at the fuel inlet;
- turbulence parameters at the air inlet.

## 2. Mathematical Model

The study of the liquid fuel evaporation is carried out in the FlowVision software package [1–4]. The evaporation of fuel droplets in the air stream is modeled using the Eulerian approach to describe the interaction between the continuous (carrier) and dispersed phase [20]. The following physical processes are modeled in the continuous phase: motion, turbulence, heat transfer, mass transfer; in the dispersed phase – motion, mass transfer, heat transfer. A detailed description of the mathematical model is given in [3], this paper only provides the short form of the main equations used in obtaining the problem solution.

### 2.1. Continuous Phase

The equations for continuity and momentum conservation in a continuous medium are

$$\frac{\partial(\phi_c \rho_c)}{\partial t} + \nabla \cdot (\phi_c \rho_c \mathbf{V}_c) = Q_d^{mass},$$

$$\frac{\partial(\phi_c \rho_c \mathbf{V}_c)}{\partial t} + \nabla \cdot (\phi_c \rho_c \mathbf{V}_c \cdot \mathbf{V}_c) = -\phi_c \nabla p + \nabla \cdot (\phi_c \hat{\tau}_{\text{eff}}).$$

Here  $\phi_c$ ,  $\rho_c$ ,  $\mathbf{V}_c$ ,  $p$  are the volume fraction, the density, the velocity, and the pressure of the continuous phase,  $Q_d^{mass}$  is the source term,  $n_d$  and  $M_d$  are the dispersed phase particle concentration and mass,  $\tau_{\text{eff}}$  is the effective viscous stress tensor, which depends on the continuous phase dynamic coefficient of molecular and turbulent viscosities. Turbulence is described using the KEFV model [3, 5, 20–23]. The dispersed phase influences the carrier phase through the terms on the right-hand side of the Navier–Stokes equations describing the motion of the carrier phase (coolant gas). The equation for the energy balance of a continuous phase, written with respect to the total enthalpy  $H_c$ , has the form

$$\frac{\partial(\phi_c \rho_c \mathbf{H}_c)}{\partial t} + \nabla \cdot (\phi_c \rho_c \mathbf{V}_c H_c) = \frac{\partial(\phi_c P)}{\partial t} - \nabla \cdot (\phi_c \mathbf{J}_q) + \phi_c Q_{vis,G} - Q_d^{enth}.$$

Here  $\mathbf{J}_q$  is the effective enthalpy diffusion flux,  $Q_{vis,G}$  and  $Q_d^{enth}$  are source terms. The mass fraction of the evaporated raw material is determined from

$$\frac{\partial(\phi_c \rho_c Y_i)}{\partial t} + \nabla \cdot (\phi_c \rho_c Y_i \mathbf{V}) = -\nabla \cdot (\phi_c \mathbf{J}_{i,\text{eff}}) + Q_{d,i}^{mass},$$

$$\mathbf{J}_{i,\text{eff}} = -(\rho D_i + \frac{\mu_t}{Sc_t}) \nabla Y_i = -(\frac{\mu_t}{Sc_i} + \frac{\mu_t}{Sc_t}) \nabla Y_i,$$

where  $Y_i$  and  $\mathbf{J}_{i,\text{eff}}$  are the continuous phase mass fraction and the effective diffusion flux of the  $i$ -th substance,  $Q_{d,i}^{mass}$  is the continuous phase  $i$ -th substance mass source due to dispersed phase mass change (if the dispersed phase consists of one substance, then  $Q_{d,i}^{mass} = Q_d^{mass}$ ),  $Sc_i = \frac{\mu}{\rho_c D_i}$  and  $Sc_t = \frac{\mu_t}{\rho_c D_t}$  are the molecular and turbulent Schmidt numbers and  $D_i$  is the continuous phase diffusion coefficient of the  $i$ -th substance. The evaporated raw materials are added to the heat carrier gas, forming the carrier phase.

## 2.2. Dispersed Phase

### 2.2.1. Approaches to modeling evaporation/condensation of particles (droplets)

Evaporation/condensation processes of particles on a surface are described by the particle mass transfer equation for variable  $M_d n_d$ :

$$\frac{\partial(M_d n_d)}{\partial t} + \nabla \cdot (\mathbf{V}_d M_d n_d) = \nabla \cdot (\frac{\nu_{t,d}}{Sc_{t,d}} \nabla (M_d n_d)) - Q_d^{mass} + M_d \dot{n}_d.$$

The values of particle concentration  $n_d^n$ ,  $n_d^{n+1}$  on two adjacent layers in time are assumed to be known during the equation integration. The source member  $Q_d^{mass}$  is figured out as

$$Q_d^{mass} = n_d \pi d^2 \dot{m}_d = n_d \pi d^2 \sum_{i=1}^{N_d} \dot{m}_{d,i}.$$

Here  $\dot{m}_d$  is the specific rate of the particle phase mass change,  $\dot{m}_{d,i}$  is the specific rate of  $i$ -th substance of the particle phase mass change,  $N_d$  is the number of the dispersed phase substances,  $d$  is the particle diameter.

### 2.2.2. Particle mass, momentum, and energy conservation equations

The particles transport is described by the inhomogeneous convective-diffusion equation for the particles concentration:

$$\frac{\partial(n_d)}{\partial t} + \nabla \cdot (\mathbf{V}_d n_d) = \nabla \cdot (\frac{\nu_{t,d}}{Sc_{t,d}} \nabla n_d) + \dot{n}_d,$$

$$\nu_{t,d} = \nu_{t,c},$$

where  $\mathbf{V}_d$  is the dispersed phase (particles) velocity,  $\nu_{t,d}$  and  $\nu_{t,c}$  are the dispersed and continuous phase kinematic coefficients of the turbulent viscosity and  $Sc_{t,d}$  – turbulent Schmidt number. The velocity of the dispersed phase (particles) in the main flow  $\mathbf{V}_d$  is determined by the particle momentum transfer equation, which is a non-uniform convective-diffusion equation for the conservative variable  $\mathbf{V}_d M_d n_d$ :

$$\frac{\partial(\mathbf{V}_{d,i} M_d n_d)}{\partial t} + \nabla \cdot (\mathbf{V}_d \mathbf{V}_{d,i} M_d n_d) = \nabla \cdot (\frac{\nu_{t,d}}{Sc_{t,d}} \nabla (\mathbf{V}_{d,i} M_d n_d)) - n_d \frac{\pi d^3}{6} \nabla_i p + F_{D,i}.$$

The force  $\mathbf{F}_D$  should be understood as the total force acting on the particle. It is obtained from the addition of repulsive, lift and drag forces. The values of the particle concentration  $n_d^n$ ,  $n_d^{n+1}$  and their masses  $M_d^n$ ,  $M_d^{n+1}$  on two adjacent layers in time are assumed to be known during the equation integration. The dispersed phase energy transfer is described by an inhomogeneous convective-diffusion equation for the conservative variable  $h_d M_d n_d$ :

$$\frac{\partial(h_d M_d n_d)}{\partial t} + \nabla \cdot (\mathbf{V}_d h_d M_d n_d) = \nabla \cdot (\frac{\nu_{t,d}}{Sc_{t,d}} \nabla (h_d M_d n_d)) + Q_d^{enth},$$

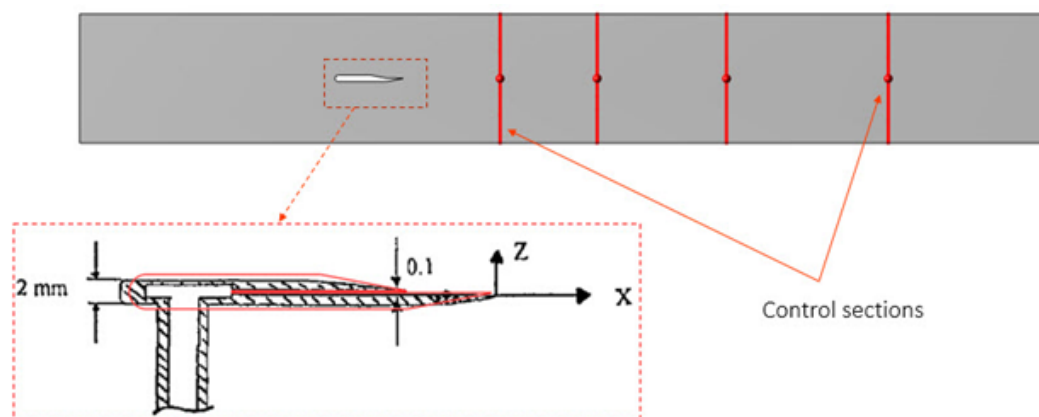
where  $h_d$  is the thermodynamic enthalpy of the particle phase.

### 3. Results and Discussion

#### 3.1. Model Calibration under the Basic Mode Conditions

##### 3.1.1. Computational grid

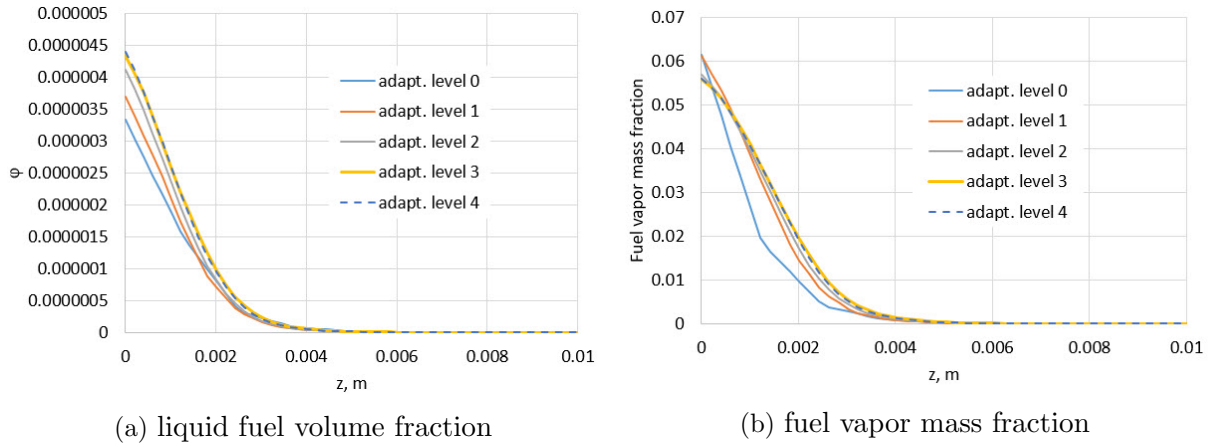
Grid convergence studies were performed for a 2D formulation of the problem. In this case, the central cross-section of the rectangular chamber was selected as the computational domain. The shape of the atomizer cross-section was modeled as illustrated in Fig. 4.



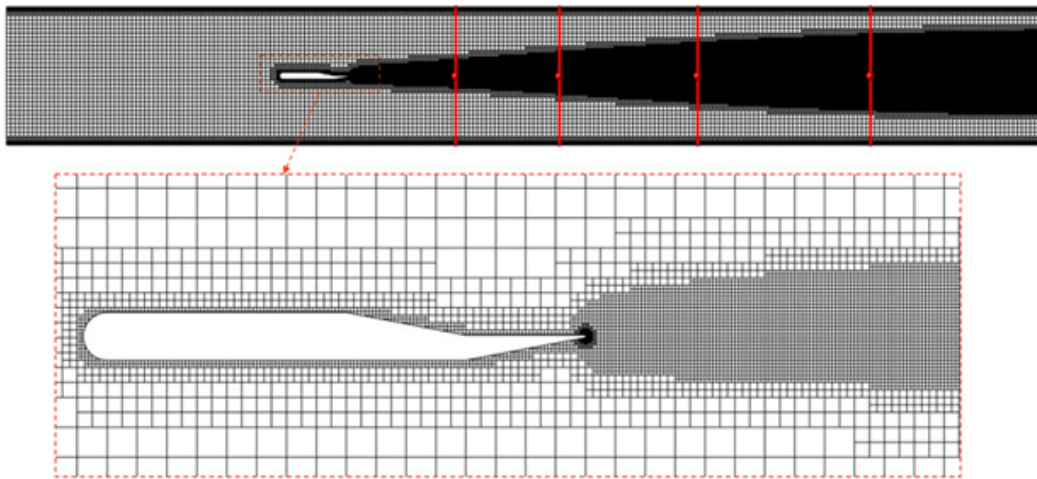
**Figure 4.** The computational domain with a schematized atomizer in the two-dimensional problem formulation

The initial mesh was generated with a uniform characteristic cell size of about 1.25 mm. Next, mesh adaptation of the computational domain volume was applied near the atomizer inlet and channel walls. In the FlowVision software, one level of adaptation involves dividing a Cartesian cell into 8 equal ones. As the computational domain was refined, the pressure drop change in the channel was monitored. As a result of the grid convergence studies, the third level of adaptation was chosen to be used near atomizer and channel walls. Selecting the fourth level of adaptation gives about a 3% change in the control parameter. The average  $Y+$  value was about 50. The single-phase flow of air alone is firstly considered under the conditions specified in Table 1 on the grid constructed as mentioned below. The turbulence intensity at the channel entrance was set to 3%, and the turbulence scale was 2.5 mm. In the simulation of single-phase flow, the air velocity obtained at a distance of 10 mm from the atomizer, near its film section, was 106 m/s. This value was set as the gas boundary condition in the fuel inlet section for the grid design refinement taking into account the dispersed phase. At the same kerosene inlet, the volume fraction of the dispersed phase was set equal to 0.1844, which provides the required fuel consumption when its velocity equals 10 m/s. In two-phase calculations of the gas-plus-kerosene system, additional adaptation of the computational mesh was applied behind the atomizer. The level of additional adaptation was based on a condition of dispersed phase volume fraction – only regions where the dispersed phase volume fraction had values in the range  $10^{-10}$  to 1 were adapted. Figure 5 shows the profiles of the liquid-fuel volume fraction and the mass fraction of its vapors, obtained at different adaptation levels in the control section furthest from the atomizer, at  $x = 150$  mm.

The third level of the adaptation with the dispersed phase volume condition was selected for use in further calculations. The cell count was 150.4 thousand. The resulting grid, which was used in calculations, is shown in Fig. 6.

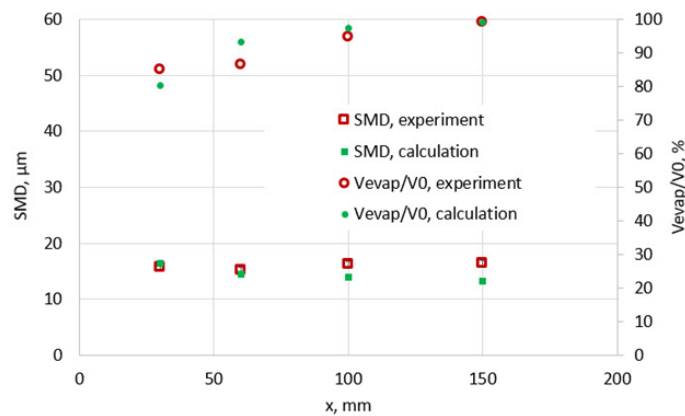


**Figure 5.** Dependence on the grid adaptation levels of the fuel (a) and vapor (b) distributions in the cross-section  $x = 150$  mm along the duct length from the center line



**Figure 6.** A grid for a two-dimensional formulation based on the results of the numerical solution convergence study

The calculated values of the control parameters – kerosene evaporation rate and Sauter mean diameter of fuel droplets are shown in Fig. 7 in comparison with the experimental data for the basic mode (Tab. 1).



**Figure 7.** Fuel droplets evaporation rate and Sauter mean diameter in basic calculation mode

3.1.2. Assessment of the influence of atomizer shape

Due to the lack of precise dimensions parameters of the atomizer used in the experiments, taking into account the accepted method of the fuel supply (droplets form from the atomizer edge), the transition to a simplified atomizer model with a rectangular cross-section, which sizes are equal to the atomizer length and the kerosene supply gap size (Fig. 8), is considered. The negligible change in the control parameters as a result of calculations with a simplified geometry indicates a weak influence of its shape on the solution in the given formulation. Subsequent calculations model the atomizer with a rectangular cross-section (simplified geometry).

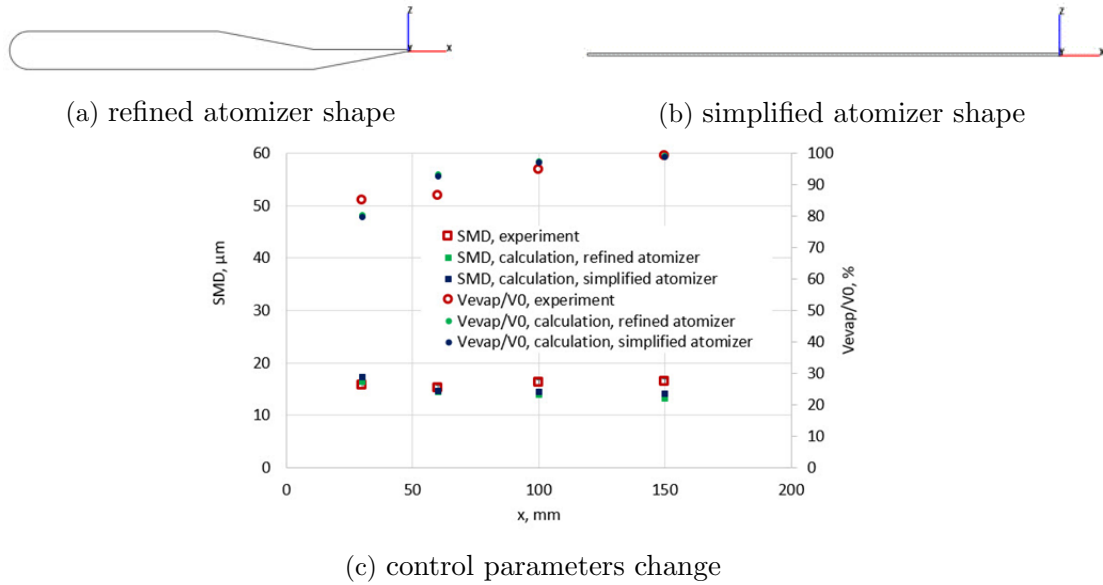


Figure 8. Assessment of the influence of atomizer shape

3.1.3. Assessment of the influence of droplet inlet velocity

Two types of fuel inlet boundary conditions were considered:

- inlet droplets velocity 10 m/s, dispersed phase volume fraction 0.1844, air velocity 106 m/s (droplet velocity values based on calculations in [7]);
- inlet droplets velocity 1.844 m/s, dispersed phase volume fraction 1, air velocity 0 m/s (droplet velocity values based on the experimental fuel flow rate).

The respective calculation results are shown in Fig. 9. These results indicate that determining kerosene inlet velocity from its flow rate is both acceptable and justifiable.

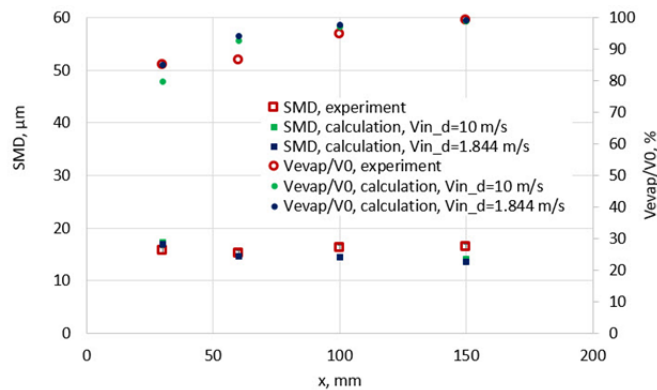


Figure 9. Assessment of the influence of droplet inlet velocity

3.1.4. Assessment of the influence of the spectrum of inlet fuel droplets

As observed in the calculation results presented above, the deviation of the droplet Sauter mean diameter from experimental values increases in control sections located further away from the atomizer. To identify the source of the deviation of this control parameters calculated values from experimentally obtained ones, the computational model solution sensitivity analysis to the spectrum of kerosene droplet diameters, which are set as inlet parameter, was carried out. The basic spectrum of fuel droplet diameters was obtained from the results of the cold experiments [7]. The considered spectrum variations, which were obtained by changing the values of the largest particle group volume fractions in the dispersed phase at the fuel inlet, are shown in Tab. 3. The basic spectrum corresponds to the set of droplets denoted D1.

**Table 3.** Variations of the particle diameter spectrum at the inlet

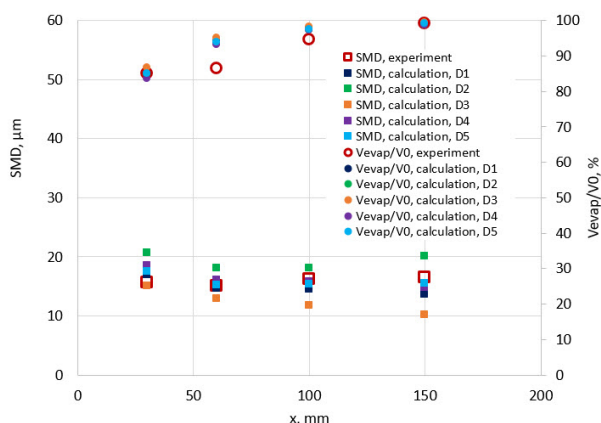
Particle group diameter D, $\mu\text{m}$	Spectrum				
	D1	D2	D3	D4	D5
Particle group volume fraction $\phi$					
2.5	0	0	0.03	0	0
7.5	0.1	0.1	0.1	0.07	0.1
12.5	0.3	0.3	0.3	0.3	0.3
17.5	0.3	0.3	0.3	0.3	0.3
22.5	0.15	0.15	0.15	0.15	0.15
27.5	0.07	0.07	0.07	0.07	0.07
32.5	0.05	0.05	0.05	0.06	0.05
37.5	0.03	0.01	0	0.05	0.01
40	0	0	0	0	0.01
45	0	0	0	0	0.01
50	0	0.01	0	0	0
60	0	0.01	0	0	0

A comparison of the control parameters calculated for the spectra of Tab. 3 is shown in Fig. 10. The spectrum of fuel droplet sizes at the input has the greatest influence on the solution out of all the considered modeled input parameters, ambiguously determined from [7]. The closest match of SMD control cross-sections to experimental data was obtained with the D5 spectrum. Subsequent studies were carried out using the inlet kerosene droplet diameter spectrum denoted D5 in Tab. 3.

3.1.5. Assessment of jet spray width

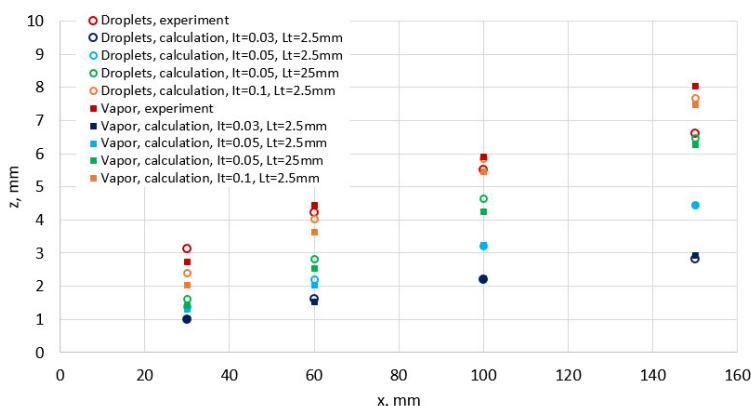
According to the experimental data in [7], the control sections are given a value Z0.9, defined as the half-width of the profile which contains 90% of the measured volumetric fuel flow rate (liquid and vapor). The factors affecting the spray width of the jet are the following:

- gas turbulence [7, 16];
- fuel droplets, which are tearing off atomizer edge, fluctuations [7, 16];
- main stream velocity fluctuations [6].



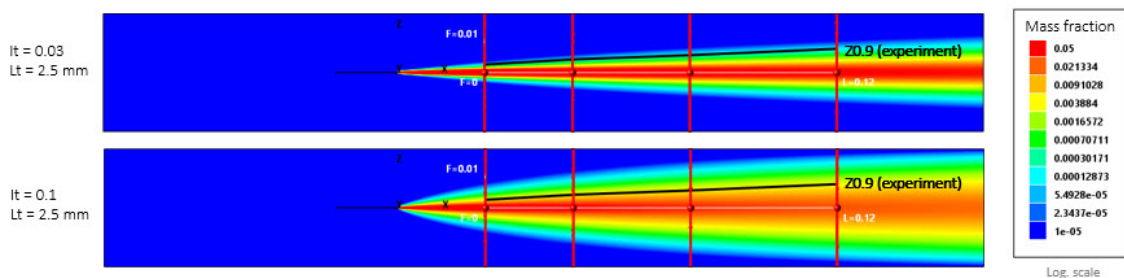
**Figure 10.** Droplets inlet size spectrum change estimation

To study the effect on the jet spray width in the computational model, it was opted to alter the parameter of inlet air turbulence. Figure 11 shows the magnitude of Z0.9 values obtained from calculations using different values of pulsations and turbulence scales in the air flow at the channel entrance.



**Figure 11.** The half-width of the profile, which contains 90% of the measured volumetric fuel consumption (liquid and vapor)

Increasing the incoming flow pulsations from 3% to 10% achieves an up to  $2\times$  increase in profile width, which contains 90% of the measured volumetric fuel consumption, thus matching the experimentally measured values. At the same time, the change in the control characteristics of evaporation rate and droplet Sauter mean diameter is negligible. Figure 12 shows changes in the kerosene vapors mass fraction distribution as a result of an increase in the airflow turbulence intensity at the inlet.



**Figure 12.** Kerosene vapors mass fraction distribution for different airflow turbulence parameters

### 3.2. Model Analysis under Different Experimental Conditions

#### 3.2.1. Influence of airflow temperature regime and pressure

In the basic experimental scenario, the air temperature is equal to 750 K and pressure is 9 bar in the duct setup. [7] presents results with these parameters changed to the following values:

- air temperature 650 K, 850K;
- air pressure 6 bar, 12 bar.

Numerical simulations with these conditions were performed using the computational model constructed in the FlowVision CFD package. Simulations were performed with a simple change in air temperature and pressure, but also with an adjustment of fuel parameters to account for the change in its properties under the new conditions. Airflow under different temperature regime interactions leads to a change in fuel temperature on the atomizer surface. Values measured with a thermocouple under different experimental conditions are given in [6]. The corresponding correction of the liquid kerosene thermophysical properties (in particular, its surface tension) determines the change in its dispersed composition after the film spraying. In order to confirm the droplet diameter size distribution, which was accepted as an input parameter from the cold experiment (paragraph 2.1) based on the [7], the droplet size diameter adjustment is defined as

$$D_{new} = k \cdot D_{ini},$$

$$k = \frac{\left(\frac{\sigma_{fuel}}{\rho_{air}^{0.3}}\right)_{new}}{\left(\frac{\sigma_{fuel}}{\rho_{air}^{0.3}}\right)_{ini}},$$

where  $D_{new}$  is the new diameter value,  $D_{ini}$  is the original diameter value,  $\sigma_{fuel}$  is the liquid fuel surface tension, and  $\rho_{air}$  is the air density. The results of control parameters (the evaporation rate and droplet Sauter mean diameter) calculations under another set of experimental conditions are shown in Fig. 13.

The calculation results approach the experimental data by adjusting the input data for fuel droplets, taking into account changes in air pressure and temperature. However, the chosen method of the correction probably does not accurately describe real conditions (in particular, the droplet size spectrum), which is the reason for the observed deviations in the control parameter values.

#### 3.2.2. Influence of the dimensionality in the problem statement

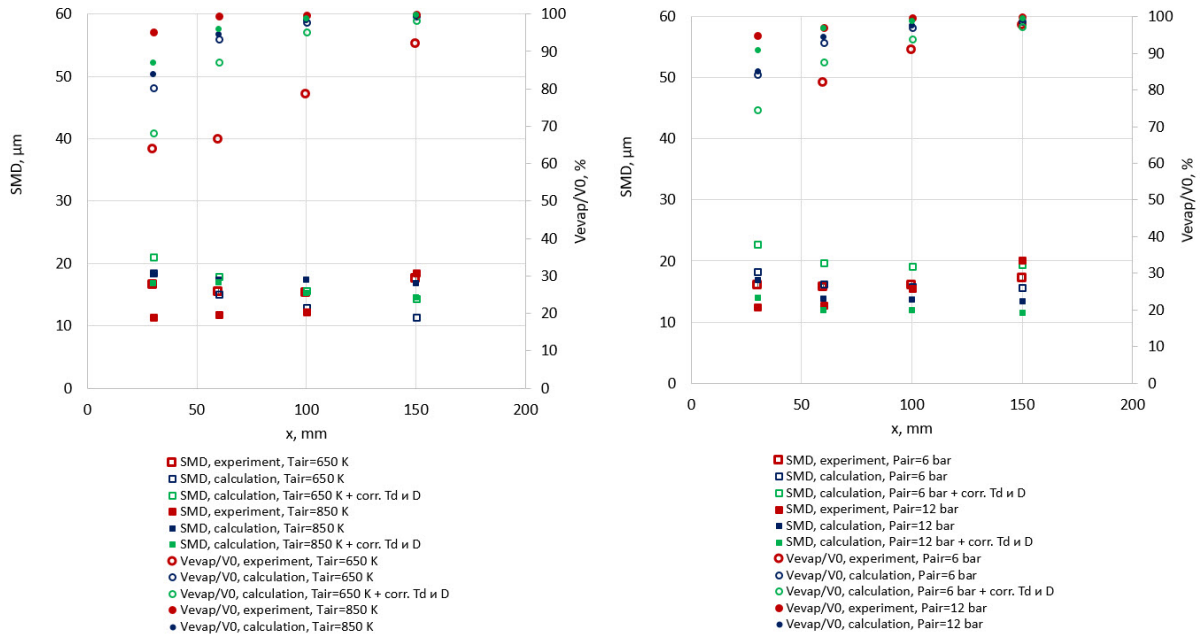
The study of kerosene evaporation in the premix duct for a gas turbines combustion chamber at high pressure is also considered as a three-dimensional formulation within the scope of this work. The parameters of the derived two-dimensional model were transferred to a three-dimensional formulation for the rectangular duct. The atomizer shape was modeled as the simplified one obtained as a result of studies carried out in paragraph 3.1.2. Figure 14 shows the isometry of the spatial computational domain and boundary conditions.

FlowVision provides the following options for boundary conditions of dispersed phase particles near solid walls.

Particles can:

- bounce elastically off the wall;
- bounce inelastically;
- stick to it.

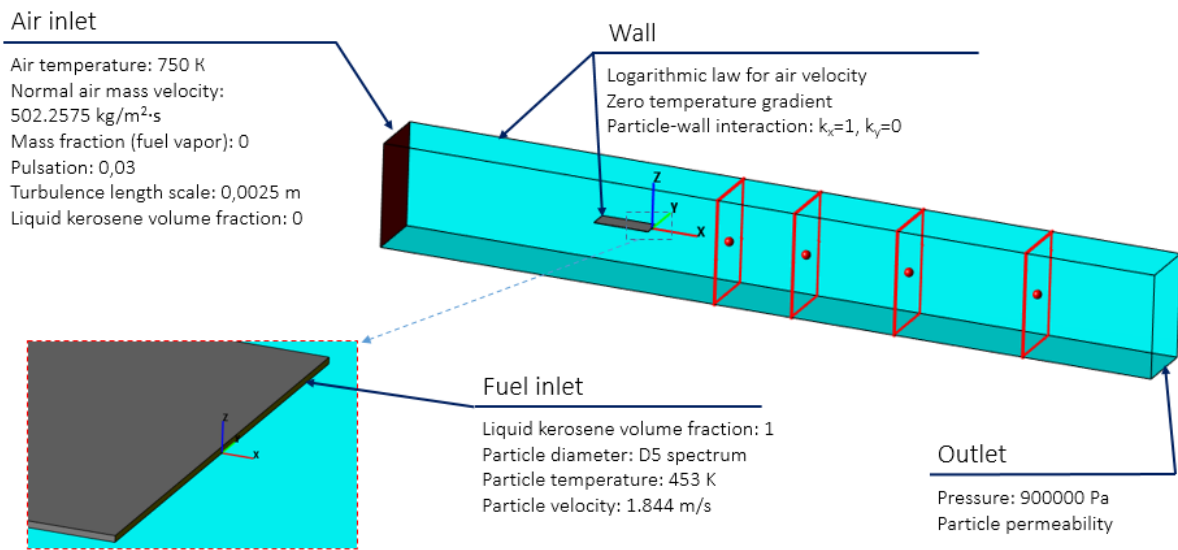




(a) air temperature: 650 K, 850 K

(b) air pressure: 6 bar, 12 bar

**Figure 13.** Calculation results for another set of experimental conditions



**Figure 14.** The computational domain and boundary conditions in the 3D problem formulation

The choice of boundary condition is determined by the tangent and normal components ( $k_x$  and  $k_y$ , respectively) of the particle momentum recovery coefficient after collision with the wall. For example, in the case of a perfectly elastic rebound conserving velocity tangential component,  $k_x = 1, k_y = 1$ ; for a perfectly inelastic rebound conserving velocity tangential component,  $k_x = 1, k_y = 0$ ; in case of sticking,  $k_x = 0, k_y = 0$ . In this study, particle interaction with the wall is defined as “perfectly inelastic rebound conserving velocity tangential component”. The adaptive grid, built for the three-dimensional problem statement, contained 3.96 million cells. Figure 15 shows the distributions of the kerosene volume fraction (top) and the droplet diameter (bottom). Figure 16 shows the distributions of the kerosene vapor mass fraction (top) and the air temperature (bottom).

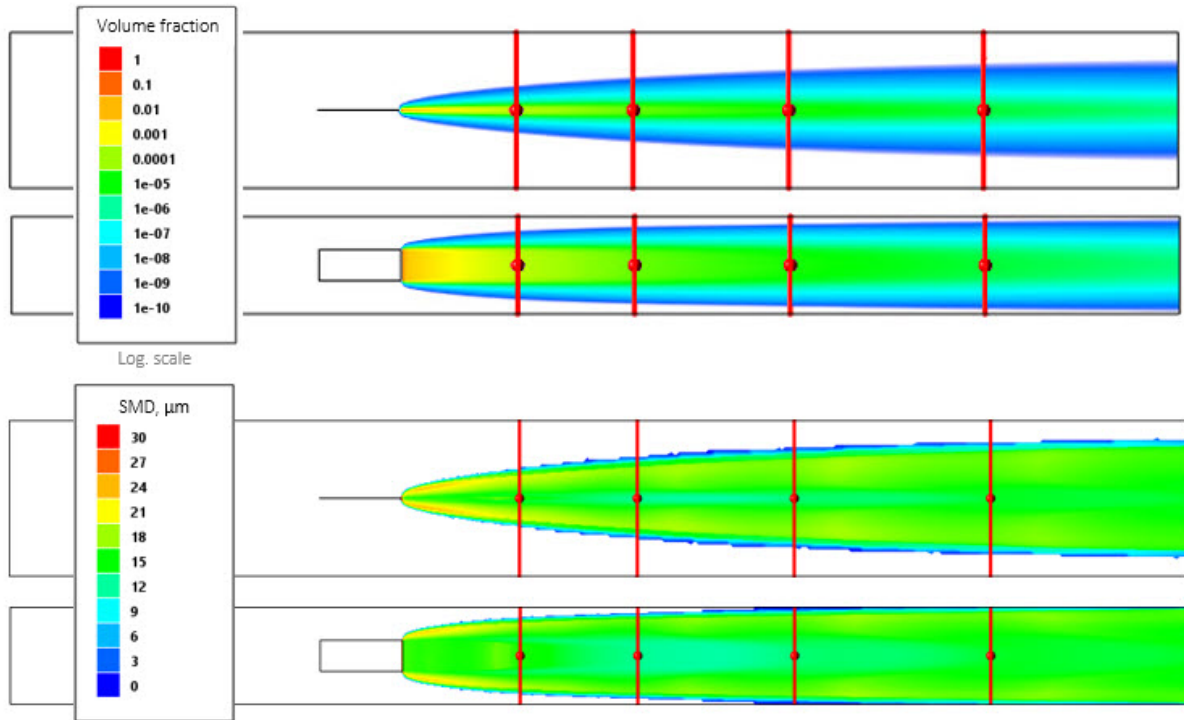


Figure 15. Liquid kerosene volume fraction (top) and SMD (bottom) distributions

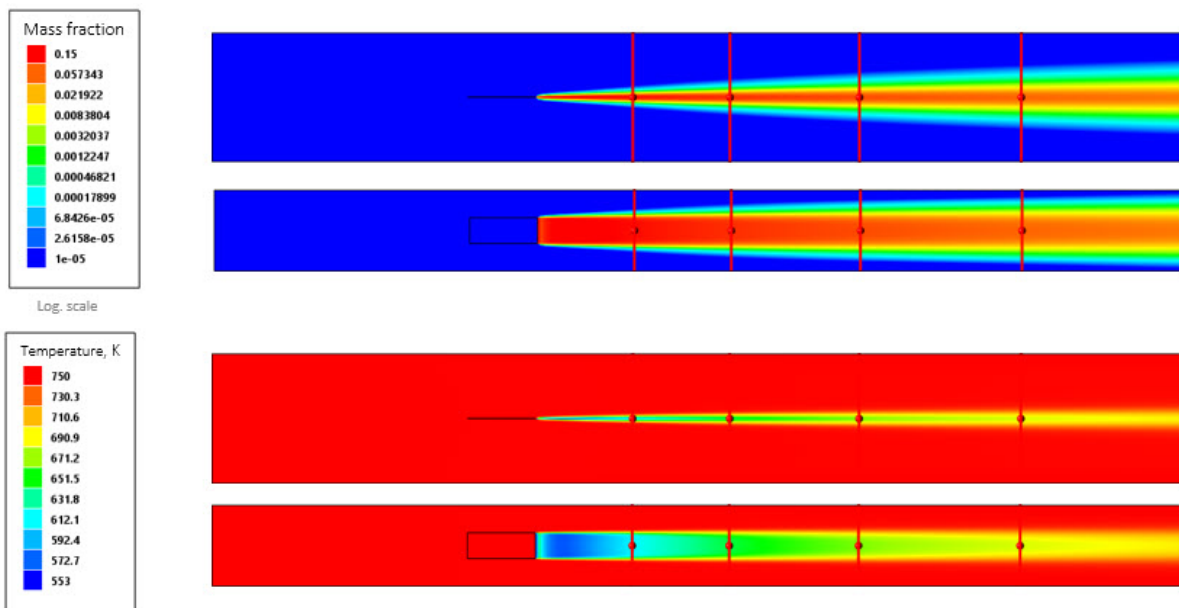


Figure 16. Kerosene vapor mass fraction (top) and air temperature (bottom) distributions

A comparison of the control parameters calculated in the three-dimensional problem statement with experimental data is shown in Fig. 17.

The resources of the Moscow State University (MSU) supercomputer complex allowed for the optimization of the time required to run the required series of test calculations and to speed up the calculation model debugging process and program validation.

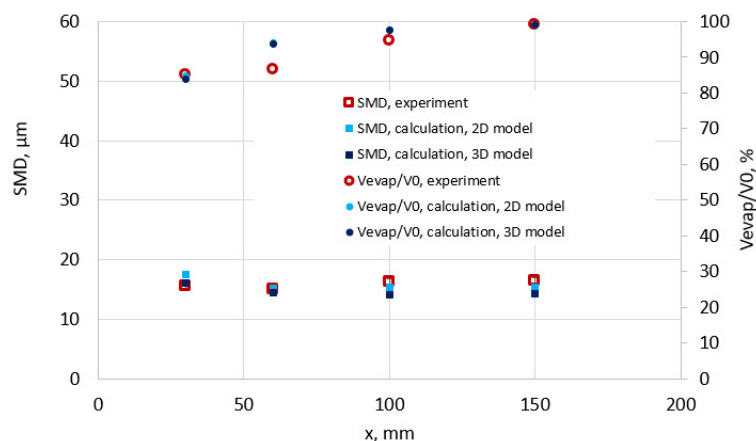


Figure 17. Calculation results on the 3D problem statement

## Conclusion

A study of kerosene evaporation in a premix duct for gas turbine combustion chambers at high pressure was carried out in the FlowVision software package based on the experimental data available in literature. Fuel droplet motion and evaporation in the airflow are modeled using the Eulerian approach to continuous- and dispersed- phase interaction. There was a study of the influence on the solution of model parameters such as the computational grid, the atomizer shape, the fuel inlet boundary conditions in terms of the velocity and droplet size, and the inlet air turbulence. In particular, it is found that

- in the modeled scenario of fuel supply in the form of droplets, the atomizer shape has negligible effect on the calculated control values;
- change in the droplet velocity at the inlet from 2 m/s to 10 m/s (in accordance with available data from other published computational studies) affects the results only in the control section closest to the atomizer, where, according to experimental data, an overestimated evaporation rate is observed;
- simulation results significantly depend on the size of fuel droplets at the inlet. It is possible to obtain a correct distribution of control parameters only for a specific droplet size spectrum, which allows to refine results by taking into account the change in droplet acceleration due to flow, the rate of their heating and evaporation depending on droplet diameter and the initial spectrum (as selected based on the cold experiment data);
- inlet air turbulence parameters affect the jet spray width. An increase in the incoming flow pulsations from 3% to 10% achieves an up to 2x increase in the profile width, which contains 90% of the measured volumetric fuel consumption, matching experimentally achieved values.

The evaporation rate and droplet Sauter mean diameter obtained in the selected observed duct sections as a result of the simulation deviate from their corresponding experimentally obtained values by 9% and 12%, respectively, for the basic calculation mode, taking into account the existing uncertainties of the initial data and the assumptions made. Based on a series of calculations for different experimental conditions, it is shown that for the constructed computational model the correct change in fuel evaporation rate with variations in temperature and air pressure is obtained.

## Acknowledgements

The authors thank the MSU Research Center and personally V. V. Voevodin for the opportunity to use the Lomonosov-2 supercomputer to obtain the results presented in this article.



*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. FlowVision 3.13.03: Users guide. [https://flowvision.ru/webhelp/fvru\\_31303](https://flowvision.ru/webhelp/fvru_31303) (2023), accessed: 2023-10-05
2. Aksenov, A.: FlowVision: Industrial computational fluid dynamics. *Computer Research and Modeling* 9(1), 5–20 (2017). <https://doi.org/10.20537/2076-7633-2017-9-5-20>
3. Aksenov, A., Zhlukto, S., Kashirin, V., *et al.*: Numerical modeling of raw atomization and vaporization by flow of heat carrier gas in furnace technical carbon production into FlowVision. *Computer Research and Modeling* 15(4), 921–939 (2023). <https://doi.org/10.20537/2076-7633-2023-15-4-921-939>
4. Aksenov, A., Zhlukto, S., Kashirin, V., *et al.*: Numerical modelling of raw materials atomization and vaporization in a heat carrier gas flow in technical carbon production based on the Euler approach. *E3S Web Conf.* 459, 04019 (2023). <https://doi.org/10.1051/e3sconf/202345904019>
5. Aksenov, A., Zhlukto, S., Platov, S.: Numerical simulation of transition at hull of a ship in FlowVision software. *Ship-building* 4, 58–60 (2013)
6. Brandt, M., Gugel, K., Hassa, C.: Experimental investigation of the liquid fuel evaporation in a premix duct for lean premixed and prevaporized combustion. *Journal of Engineering for Gas Turbines and Power* 119, 815–821 (1997). <https://doi.org/10.1115/1.2817059>
7. Brandt, M., Rachner, M., Schmitz, G.: An experimental and numerical study of kerosene spray evaporation in a premix duct for gas turbine combustors at high pressure. *Combustion Science and Technology* 138:1-6, 313–348 (1998). <https://doi.org/10.1080/00102209808952074>
8. Brinckman, K., Hosangadi, A., Ahuja, V., *et al.*: A CFD Methodology for Liquid Jet Breakup and Vaporization Predictions in Compressible Flows. 46th AIAA Aerospace Sciences Meeting and Exhibit (2008). <https://doi.org/10.2514/6.2008-1023>
9. Hoyas, S., Gil, A., Momp-Laborda, J., Khuong-Anh, D.: A large-eddy simulation of diesel-like gas jets. *Int. J. Vehicle Systems Modelling and Testing* 6(3/4), 268–282 (2011). <https://doi.org/10.1504/IJVSMT.2011.044229>
10. Irannejad, A., Banaeizadeh, A., Jaber, F.: Large eddy simulation of turbulent spray combustion. *Combustion and Flame* 162(2), 431–450 (2015). <https://doi.org/10.1016/j.combustflame.2014.07.029>

11. Keser, R., Battistoni, M., Im, H., Jasak, H.: An eulerian multi-fluid model for high-speed evaporating sprays. *Processes* 9(6), 941 (2021). <https://doi.org/10.3390/pr9060941>
12. Luo, K., Pitsch, H., Pai, M., Desjardins, O.: Direct numerical simulations and analysis of three-dimensional n-heptane spray flames in a model swirl combustor. In: *Proceedings of the Combustion Institute*. vol. 33, pp. 2143–2152 (2011). <https://doi.org/10.1016/j.proci.2010.06.077>
13. Malaguti, S., Cantore, G., Fontanesi, S., *et al.*: CFD Investigation of Wall Wetting in a GDI Engine under Low Temperature Cranking Operations. SAE Technical Paper 2009-01-0704 (2009). <https://doi.org/10.4271/2009-01-0704>
14. Perini, F., Mattarelli, E.: A large-eddy simulation of diesel-like gas jets. *International Journal of Engine Research* 12(4), 311–335 (2011). <https://doi.org/10.1177/1468087411401285>
15. Rachner, M.: Die stoffeigenschaften von kerosin Jet A-I. *DLR-Mitteilungen* 98-01 (1) (1998)
16. Rachner, M., Brandt, M., Eickhoff, H., *et al.*: A numerical and experimental study of fuel evaporation and mixing for lean premixed combustion at high pressure. *Symposium (International) on Combustion* 26(2), 2741–2748 (1996). [https://doi.org/10.1016/S0082-0784\(96\)80111-1](https://doi.org/10.1016/S0082-0784(96)80111-1)
17. Schmehl, R., Klose, G., Maier, G., Wittig, S.: Efficient numerical calculation of evaporating sprays in combustion chamber flows. *RTO AVT Symposium on “Gas Turbine Engine Combustion, Emissions and Alternative Fuels”* (1998)
18. Som, S., Aggarwal, S.: Effects of primary breakup modeling on spray and combustion characteristics of compression ignition engines. *Combustion and Flame* 157(6), 1179–1193 (2010). <https://doi.org/10.1016/j.combustflame.2010.02.018>
19. Som, S., Ramirez, A., Longman, D., Aggarwal, S.: Effect of nozzle orifice geometry on spray, combustion, and emission characteristics under diesel engine conditions. *Fuel* 90(3), 1267–1276 (2011). <https://doi.org/10.1016/j.fuel.2010.10.048>
20. Sorokin, K., Zhluktov, S., Aksenov, A.: On the implementation of the Euler approach to modeling polydisperse media in the FlowVision software package. *Thermophysics and Physical Hydrodynamics. Collection of abstracts of the VI All-Russian scientific conference with elements of the school of young scientists* p. 201 (2021)
21. Tonini, S., Gavaises, M., Theodorakakos, A.: Modelling of high-pressure dense diesel sprays with adaptive local grid refinement. *International Journal of Heat and Fluid Flow* 29(2), 427–448 (2008). <https://doi.org/10.1016/j.ijheatfluidflow.2007.11.009>
22. Tonini, S., Gavaises, M., Theodorakakos, A., Cossali, G.: Numerical investigation of a multiple injection strategy on the development of high-pressure diesel sprays. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. vol. 224, pp. 125–141 (2010). <https://doi.org/10.1243/09544070JAUTO1083>
23. Zhluktov, S., Aksenov, A., Karasev, A.: Simulation of a bypass laminar-turbulent transition in the framework of the k- $\varepsilon$ -approach. *Computer Research and Modeling* 6(6), 879–888 (2014)

# Analysis and Optimization of Output Operations in the INM RAS Earth System Model

*Maria A. Tarasevich*<sup>1,2,3</sup>, *Ivan V. Tsybulin*<sup>4</sup>, *Evgeny M. Volodin*<sup>1</sup> ,  
*Andrey S. Gritsun*<sup>1</sup> 

© The Authors 2023. This paper is published with open access at SuperFri.org

The modern development of complex Earth system models forces developers to take into account not only the computational efficiency, but also the performance of the data input and output. This work evaluates the data output performance of the INM RAS Earth system model and optimizes its weak points. The output operations were found to be surprisingly slow on the Cray XC40-LC supercomputer compared to the results obtained on the INM RAS cluster. To identify the bottleneck, the computational time, the distributed data gathering time, and the file system output time were measured separately. The distributed data gathering time was the cause of the slowdown on the Cray XC40-LC, so optimizations were made to the gathering routines without any additional rework of the existing output code. The optimizations resulted in a significant reduction in the overall model running time on the Cray XC40-LC, while the gathering time itself was reduced by a factor of  $10^2$ – $10^3$ . The results highlight the importance of optimizing the output performance in Earth system models.

*Keywords:* Earth system model, INMCM6, MPI, data gathering, derived types, manual packing, Cray.

## Introduction

The horizontal resolution in Earth system models (ESMs) has been increasing for the past two decades. In Coupled Model Intercomparison Project Phase 3 (CMIP3), the typical horizontal resolution was 250 km in the atmosphere and 150 km in the ocean models, while in CMIP6 this increased to 125 km and 75 km respectively [7]. In HighResMIP horizontal resolution is 50 km in the atmosphere model and 25 km in the ocean model, reaching eddy-permitting scale [11]. Between CMIP5 and CMIP6, including HighResMIP, the number of vertical levels in the atmosphere and ocean models nearly doubled [7]. Global models with finer grids capture more aspects of the circulation of the atmosphere with upper stratosphere and ocean, resolving more physical processes explicitly instead of parameterizing them.

Not only the spatial resolution of ESMs increases, but also does the complexity and range of described processes. For the last 4 decades aerosols, carbon cycle, dynamical vegetation, atmospheric chemistry and land ice have extended [16] the typical components of ESMs, which previously were only atmosphere, land, ocean and sea ice. Each ESM component produces more diagnostic information that can be used for applications.

Another modern trend is seamless weather-climate prediction approach [12], which means using one ESM to forecast on different timescales. This seamless approach forces the ESMs to support different frequency of output ranging from daily and monthly (climate scale) to hourly, allowing to capture diurnal cycle and predict extreme precipitation events well.

An increase in grid resolution and complexity of ESMs together with modern application ESMs for weather prediction leads to increase in amount of output data, including model results,

<sup>1</sup>Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russian Federation

<sup>2</sup>Hydrometeorological Research Center of Russian Federation, Moscow, Russian Federation

<sup>3</sup>Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation

<sup>4</sup>Yandex.Technologies, Moscow, Russian Federation

diagnostics, and intermediate variables, which need to be stored. Writing and saving this massive amount of data can degrade the overall performance and scalability of the model [1, 2].

INM RAS Earth system model (INMCM) is a global climate model. Two versions of the INM RAS Earth system model – INMCM5 [27, 28] and INMCM48 [29] – participate in the CMIP6 [9] and show good results [6]. INMCM5 is capable of simulating the present-day climate [27, 28], as well as its changes in 1850–2014 [23]. Moreover, INMCM5 simulates extreme climate and weather phenomena well [14, 19, 30]. Following the seamless approach, new systems of long-range [8, 15, 20, 33–35] and decadal [13, 31, 32] weather forecast based on the INMCM5 have been developed and have successfully completed operational testing at the Hydrometcenter of Russia.

The INM RAS Earth system model is constantly evolving, and the first version of the new model generation called INMCM6 has recently been released [22]. Following the world trends in climate modelling mentioned above, during the further INMCM6 development [18] we intend to supplement the model with new components, such as atmosphere chemistry, land nitrogen cycle, dynamical vegetation, methane cycle, ocean biochemistry. We also consider the version with spatial resolution of  $1^\circ$  in the atmosphere and  $0.25^\circ$  in the ocean as the primary candidate to participate in CMIP7 and a basis for the next version of long range weather forecast system. That means that output operations can become a bottleneck for INMCM too.

In this paper we evaluate the INMCM data output performance on two HPC systems and determine the weak points. We optimize the weak points without redesigning the output system from scratch. After that we give some insights on future evolution of INMCM output subsystem.

The organization of this paper is as follows. Section 1 provides an overview of the INM RAS climate model and the HPC systems used for numerical experiments. Section 2 describes the methodology of measurements and presents the output performance and its scalability. Section 3 explains the changes we made to optimize the output. Section 4 presents the effect of optimizations applied to the INMCM output. Finally, the conclusions summarize the results and highlight the future steps.

## 1. Materials and Methods

### 1.1. INM RAS Earth System Model

There is a family of INM RAS Earth system model versions with different spatial and temporal resolutions and different sets of included modules. Basically, INMCM consists of two models: the atmosphere model with interactive aerosol [24] and land surface [25, 26] modules and the ocean model [21] with sea ice dynamics and thermodynamics module [36, 37]. The aerosol module describes the evolution of the 10 substances concentrations. The sea ice dynamics and thermodynamics module describes sea ice with the elastic-viscous-plastic rheology with a single gradation of thickness.

The atmosphere model is based on the system of the hydrothermodynamic equations with hydrostatic approximation in advective form. The atmosphere general circulation model uses a semi-implicit integration scheme that requires solving an auxiliary Helmholtz-type equation at each dynamical step. The current version uses a fast Fourier transform based algorithm and requires global data transposing for its parallel implementation [10].

The ocean model solves a set of large-scale hydrothermodynamic equations with hydrostatic and Boussinesq approximations. The ocean model step consists of several stages. Two most com-

putationally demanding – an isopycnal diffusion and dissipation of the horizontal components of velocity – have recently been optimized [4, 5]. The other one is the barotropic adaptation because it requires solving a system of three implicitly discretized equations for the velocity components and the sea level. This system is solved [21] iteratively using GMRES with the block ILU(0) preconditioner from the PETSc [3] package.

The atmosphere and the ocean general circulation models and the aerosol module are implemented as coupled distributed applications that exchange data using MPI library. The aerosol module works on the same grid as the atmosphere model and uses the same size of MPI communicator.

In the study we use two versions from the up-to-date INM RAS Earth system model generation INMCM6 [22] – INMCM6LM and INMCM6M.

The horizontal resolution of the INMCM6LM atmosphere model is  $2^\circ \times 1.5^\circ$  in longitude and latitude. The time step in the atmosphere dynamic core is 3 minutes for this spatial resolution. INMCM6M atmosphere model has the horizontal resolution of  $1.25^\circ \times 1^\circ$  in longitude and latitude, and its atmosphere dynamic core does 32 steps per hour resulting in 1.88 minute time step.

The vertical resolution of the atmosphere model is the same for INMCM6LM and INMCM6M. There are 73 vertical  $\sigma$ -levels with the resolution in the stratosphere about 500 m. The ocean general circulation model is also the same and has a horizontal resolution of  $0.5^\circ \times 0.25^\circ$  in longitude and latitude and 40 vertical  $\sigma$ -levels. The time step in the ocean model is 12 minutes.

In the current INMCM implementation the most output is produced by the atmosphere model and is written by the atmosphere root process sequentially. In this paper we focus on atmosphere model output because ocean model output is limited only to monthly averaged fields. The set of atmosphere output fields includes the prognostic variables, the land surface parameters such as soil water and temperature, snow water equivalent, surface and radiative flux components. The output is grouped by both type (2D or 3D) and periodicity of writing. The atmosphere component output properties are summarized in Tab. 1. For INMCM6LM  $nlat = 120$ ,  $nlon = 180$ , for INMCM6M  $nlat = 180$ ,  $nlon = 288$ . For both model versions  $nplevs = 26$ ,  $nslevs = 73$ .

**Table 1.** Output of INMCM atmosphere component

Group name	Number of fields	Periodicity	Size of one field	Type
dyz	17	1 per 24h	$nplevs \times nlat$	2D
dxy	71	1 per 24h	$nlat \times nlon$	2D
dxyz	7	1 per 24h	$nplevs \times nlat \times nlon$	3D
dxys	10	1 per 24h	$nslevs \times nlat \times nlon$	3D
6h	44	4 per 24h	$nlat \times nlon$	2D
3h	37	8 per 24h	$nlat \times nlon$	2D
1h	2	24 per 24h	$nlat \times nlon$	2D

## 1.2. HPC Systems

The INMCM output performance was studied using two following high performance computing systems.



The first is a Cray XC40-LC massively parallel supercomputer installed at the Main Computer Center of Federal Service for Hydrometeorology and Environmental Monitoring. The Cray XC40-LC consists of 976 compute nodes interconnected via the Cray’s proprietary Aries network. Each node has 128 Gb of RAM and two Intel Xeon E5-2697v4 processors with 18 CPU cores and 45 Mb of Intel Smart Cache per processor. The total number of computational cores is 35136. This supercomputer uses Lustre 3.2 distributed file system.

The second is the INM RAS cluster installed at the Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences. This HPC system has 36 compute nodes connected by a single Mellanox EDR MSB7800 InfiniBand with up to 100 Gb/s bandwidth. Each node has two 20-core Intel Xeon Gold 6230v2 processors with 27.5 Mb of Intel Smart Cache per processor. The total number of available CPU cores is 1440. Each compute node has 256 Gb of RAM. The main storage is a RAID6 drive shared with computing nodes using NFS4 via 1Gb/s network.

The main differences between the HPC systems used are the number and performance of CPU cores (newer cores on the INM RAS cluster), the interconnect used and the file system type. We use slightly different versions of Intel compilers to compile and build INMCM: 19.1.2.254 on the Cray XC40-LC and 19.1.3.304 on the INM RAS cluster. The compilation optimization flags are the same on both HPC systems, `-O3` for the atmosphere model and aerosol module and `-O2` for the ocean model. MPI library implementations were different across the systems: Intel MPI 2019.9.304 was used on the INM RAS cluster, and Cray MPICH 7.7.16 was used on the Cray XC40-LC.

## 2. Analysis of INM RAS Earth System Model Output Performance

In [17] we evaluated the INMCM6M performance on the same two HPC systems and found its scalability as satisfying. However, these measurements were carried out with only monthly data output enabled. When the model was operated on Cray XC40-LC in production mode we observed huge slowdown unless the output is disabled. This fact drew our attention to the INM RAS Earth system model output performance.

**Table 2.** List of used INMCM configurations

MPI processes	INMCM6M		MPI processes	INMCM6LM
ATM	Cray XC40-LC	INM RAS	ATM	Cray XC40-LC
84	✓	✓	36	✓
120	✓	✓	60	✓
144	✓	✓	72	✓
180	✓	✓	84	✓
240	✓	✓	120	✓
288	✓	✓	240	✓
312		✓	270	✓
360	✓	✓		
432	✓			
540	✓			
720	✓			

We carried out a range of time measuring experiments for INMCM6M and INMCM6LM with full output of atmosphere component enabled. All INMCM simulations are performed under the conditions of the CMIP6 piControl scenario [9]. All forcings are fixed at conditions of the year of 1850. All simulations last for 1 model month. The configurations of the experiments are summarized in Tab. 2. All configurations are chosen so that the running time is determined by the atmosphere model [17].

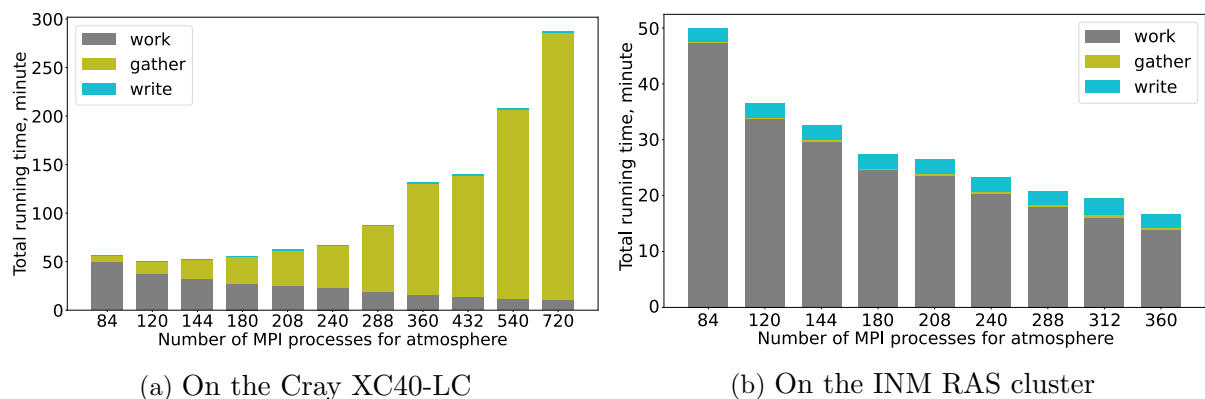
To analyze the output performance we added special tracing calls to all output subroutines. Each call appends current `MPI_Wtime` along with a small text message to an in-memory log. Since all output is done on the root MPI process, all time measurements were also done only on the root process.

At the end of the program the tracing log is flushed to a file, minimizing runtime tracing overhead. In any case, the overhead is not significant since we add tracing calls manually and only for those subroutines that are involved in producing output. When the tracing log file is produced by the program, an extra postprocessing step is necessary to extract all timings from the log and aggregate them by specific code regions.

After the initial code analysis we have divided the whole atmosphere code flow into three major parts: **work** – the computation process itself; **gather** – collecting the output data on the root process; **write** – writing the collected data to the file system.

## 2.1. INMCM6M

The results shown in Fig. 1 for INM RAS cluster and Cray XC40-LC are quite different. On the INM RAS cluster the running time is determined by the **work** and **write** part with the **gather** part being negligible. The **work** time reduces when the number of used MPI processes increases. The **write** part remains almost constant, which agrees with the fact that all output is done by a single process. The **gather** time increases, but remains under 3% of the total running time.

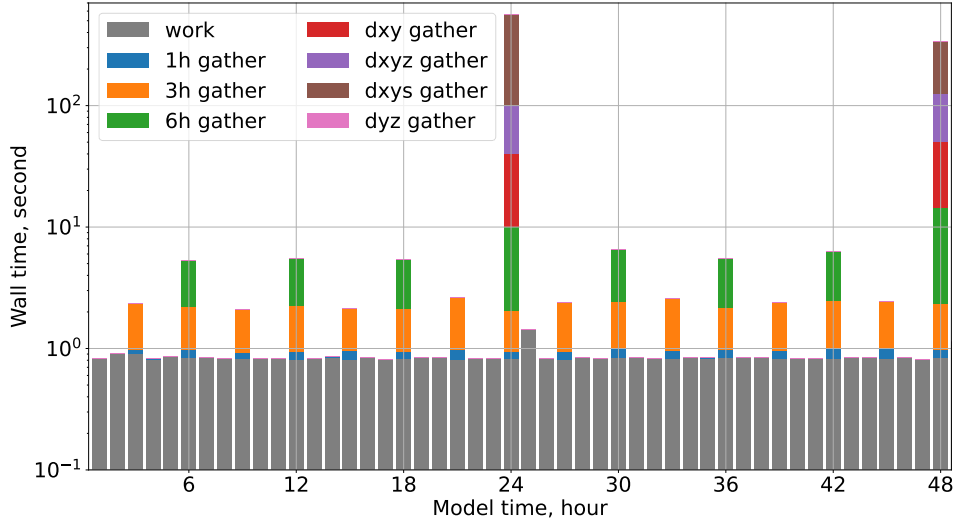


**Figure 1.** INMCM6M running wall time against the number of MPI processes for the atmosphere component

In contrast, on the Cray XC40-LC the **gather** time grows very quickly, preventing any speedup when the number of processes increases. Starting from 180 atmosphere processes configuration, the model spends most of its running time gathering output data.

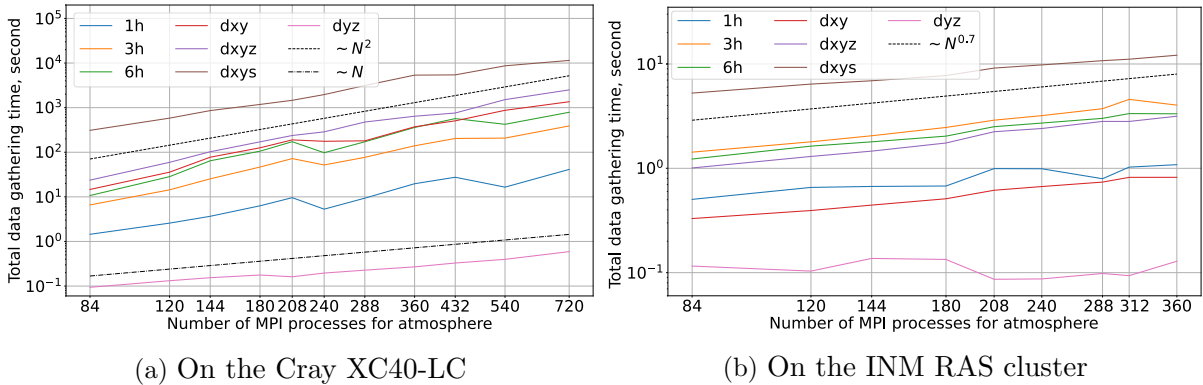
The computational times (**work**) on both systems are in good agreement. The output writing on the INM RAS cluster is slower by 2.5–3 times, which can be explained by using a high performance distributed file system on Cray XC40-LC.

Figure 2 shows the distribution of time spent in gathering different output field types on a timeline. Most of the time is consumed by gathering daily 3D output on  $\sigma$ -levels. All gathering operations except for the `dyz` and 1h output data are taking considerable amount of time compared to the actual computation time.



**Figure 2.** INMCM6M wall time distribution on the Cray XC40-LC for each model hour using 720 MPI processes for the atmosphere component. The wall time axis is in logarithmic scale

Figure 3 demonstrates that for the Cray XC40-LC the gathering time scales as the square of the number of involved MPI processes, while on the INM RAS cluster it remains sublinear. For all output except for the `dyz` the gathering is done from all of the processes, but for `dyz` output it is done only along one meridian. Gathering of `dyz` is fast enough, so we will not consider it further.



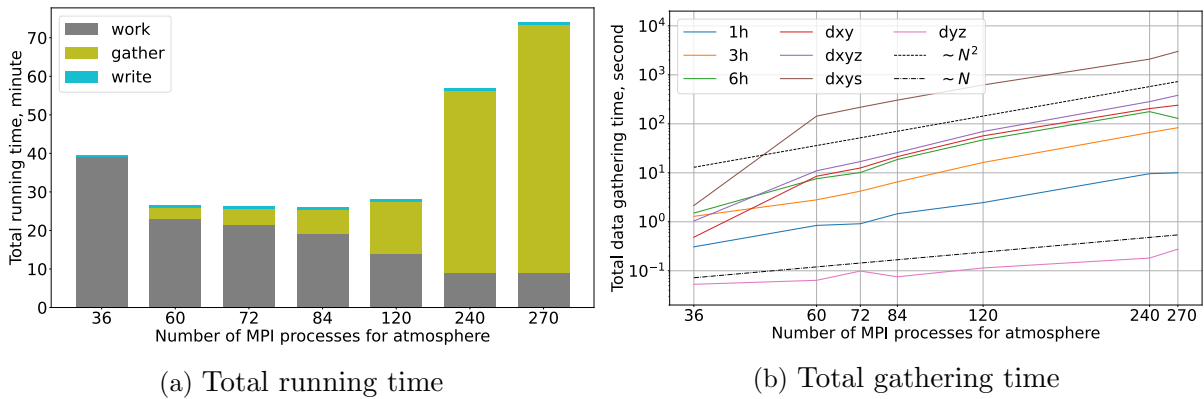
(a) On the Cray XC40-LC

(b) On the INM RAS cluster

**Figure 3.** INMCM6M gathering time for different types of fields against the number of MPI processes for the atmosphere component

## 2.2. INMCM6LM

We have also studied how output affects the running time of the INMCM6LM, which has coarser horizontal resolution than INMCM6M. Its performance on Cray XC40-LC is similar to INMCM6M and is presented in Fig. 4. The gathering time is negligible when all processes share the same computational node (the first configuration) and grows quickly when the atmosphere component uses several nodes.



**Figure 4.** INMCM6LM running wall time and gathering time against the number of MPI processes for the atmosphere component

### 2.3. Details of Model Output

INM RAS Earth system model uses arrays that are distributed along the longitude and the latitude directions. Each process allocates a part of the array including halo zone for exchanges with neighbors. The root process, which has rank 0, allocates the array as full, but uses only a part of it, except for the output. When an output of a distributed array is needed, the array is gathered on the root process.

All output routines consist of gathering data and writing it to the file system. The outline of their implementation is given in Listing 1. For 3D arrays the subroutine Gather2D is called inside a loop over vertical coordinate.

---

#### Algorithm 1 Output a distributed field

---

<pre> <b>procedure</b> OUTPUT2D(<i>A</i>, <i>record</i>)      GATHER2D(<i>A</i>)      <b>if</b> <i>myrank</i> = 0 <b>then</b>         <i>f</i> ← OPENFILE(...)         WRITEFILE(<i>f</i>, <i>record</i>, <i>A</i>)         CLOSEFILE(<i>f</i>)     <b>end if</b> <b>end procedure</b>                 </pre>	<pre> <b>procedure</b> OUTPUT3D(<i>A</i>, <i>record</i>)     <b>for</b> <i>k</i> = 1, ..., <i>n<sub>k</sub></i> <b>do</b>         GATHER2D(<i>A</i>(:, :, <i>k</i>))     <b>end for</b>     <b>if</b> <i>myrank</i> = 0 <b>then</b>         <i>f</i> ← OPENFILE(...)         WRITEFILE(<i>f</i>, <i>record</i>, <i>A</i>)         CLOSEFILE(<i>f</i>)     <b>end if</b> <b>end procedure</b>                 </pre>
---	---

---

Gathering of a 2D distributed array is done in a straightforward linear algorithm given in Listing 2. For clarity the MPI\_TYPE\_COMMIT and MPI\_TYPE\_FREE calls are omitted.

There is no clear understanding why such implementation of GATHER2D leads to  $O(nproc^2)$  time complexity observed on Cray XC40-LC. One possible explanation is: the 2D arrays are quite small, even for high resolution version the whole 2D array is only 200 kB and each process owns and sends less than 1 kB. All these messages are sent without blocking the sending part of the communication and are quickly delivered to the root process receiving queue, flooding it with  $O(nproc)$  messages. After that the root process needs to repeatedly scan the queue looking for a message from a certain rank, resulting in  $O(nproc^2)$  complexity.

**Algorithm 2** Gathering of a distributed 2D array

---

```

procedure GATHER2D( $A$ ) ▷  $A$  – a distributed 2D array
Require:  $A(i_b : i_e, j_b : j_e)$  – elements owned by the current process
  if  $myrank = 0$  then
    for  $rank = 1, \dots, nprocs - 1$  do
       $i'_b, i'_e, j'_b, j'_e \leftarrow \text{DOMAIN}(rank)$  ▷ elements owned by  $rank$ 
       $type \leftarrow \text{MPI\_TYPE\_VECTOR}(A(i'_b : i'_e, j'_b : j'_e))$ 
       $\text{MPI\_RECV}(A(i'_b, j'_b), 1, type, rank, tag, comm)$ 
    end for
  else
     $type \leftarrow \text{MPI\_TYPE\_VECTOR}(A(i_b : i_e, j_b : j_e))$ 
     $\text{MPI\_SEND}(A(i_b, j_b), 1, type, 0, tag, comm)$ 
  end if
end procedure

```

---

### 3. Output Optimization

Previously we have shown that output time is dominated by distributed arrays gathering time. Therefore it is sufficient to optimize the gathering routines without any additional rework of the existing output code.

#### 3.1. Gathering 3D Fields

The original implementation of 3D output is inefficient: it repeatedly gathers rather small 2D arrays. So we made a special subroutine for gathering 3D arrays that transfers whole 3D parts instead of slicing them. We introduced an extra 3D derived MPI type that combines all 2D slices into a single datatype. The implementation is given in Listing 3.

**Algorithm 3** Gathering of a distributed 3D array

---

```

procedure GATHER3D( $A$ ) ▷  $A$  – a distributed 3D array
Require:  $A(i_b : i_e, j_b : j_e, 1 : n_k)$  – elements owned by the current process
  if  $myrank = 0$  then
    for  $rank = 1, \dots, nprocs - 1$  do
       $i'_b, i'_e, j'_b, j'_e \leftarrow \text{DOMAIN}(rank)$  ▷ elements owned by  $rank$ 
       $slice \leftarrow \text{MPI\_TYPE\_VECTOR}(A(i'_b : i'_e, j'_b : j'_e))$ 
       $type \leftarrow \text{MPI\_TYPE\_HVECTOR}(n_k, 1, \text{sizeof}(A(:, :, 1)), slice)$ 
       $\text{MPI\_RECV}(A(i'_b, j'_b, 1), 1, type, 0, tag, comm)$ 
    end for
  else
     $slice \leftarrow \text{MPI\_TYPE\_VECTOR}(A(i_b : i_e, j_b : j_e))$ 
     $type \leftarrow \text{MPI\_TYPE\_HVECTOR}(n_k, 1, \text{sizeof}(A(:, :, 1)), slice)$ 
     $\text{MPI\_SEND}(A(i_b, j_b), 1, type, 0, tag, comm)$ 
  end if
end procedure

```

---

### 3.2. Gathering 2D Fields

The MPI library has builtin collective call `MPI_GATHERV` that is capable of gathering a 1D array from blocks of unequal size. However, one cannot directly use it with partitioned 2D array, since its blocks are non-contiguous in memory.

To make use of `MPI_GATHERV`, we need to reshape 2D subarrays into a contiguous memory block, gather them into an auxiliary buffer and then unpack it into the 2D array on the root process. This buffer and extra arguments for `MPI_GATHERV` can be initialized once and reused on invocations. The implementation is outlined in Listing 4.

---

**Algorithm 4** Optimized gathering of a distributed 2D array

---

```

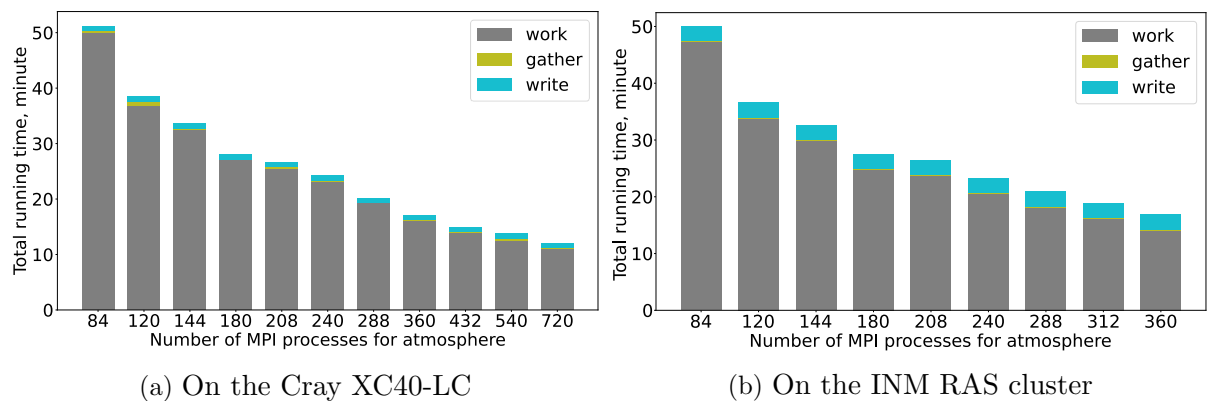
procedure GATHER2D_OPT( $A$ ) ▷  $A$  — a distributed 2D array
Require:  $A(i_b : i_e, j_b : j_e)$  — elements owned by the current process
Require: allocated  $recvbuf, sendbuf$ , filled  $recvcnts, displs$ 
    if  $myrank = 0$  then
        MPI_GATHERV(MPI_IN_PLACE, 0, recvbuf, recvcnts, displs, tag, comm)
        for  $rank = 1, \dots, nprocs - 1$  do
             $i'_b, i'_e, j'_b, j'_e \leftarrow \text{DOMAIN}(rank)$  ▷ elements owned by  $rank$ 
             $A(i'_b : i'_e, j'_b : j'_e) \leftarrow \text{UNPACK2D}(rank, recvbuf)$ 
        end for
    else
         $sendbuf \leftarrow \text{PACK2D}(myrank, A(i_b : i_e, j_b : j_e))$ 
        MPI_GATHERV(sendbuf, sendcount, \dots, tag, comm)
    end if
end procedure
    
```

---

The `PACK2D` and `UNPACK2D` are auxiliary routines that convert between 2D and 1D in native Fortran column-major order.

## 4. Results

### 4.1. INMCM6M



**Figure 5.** INMCM6M running wall time against the number of MPI processes for the atmosphere component after output data gathering was optimized

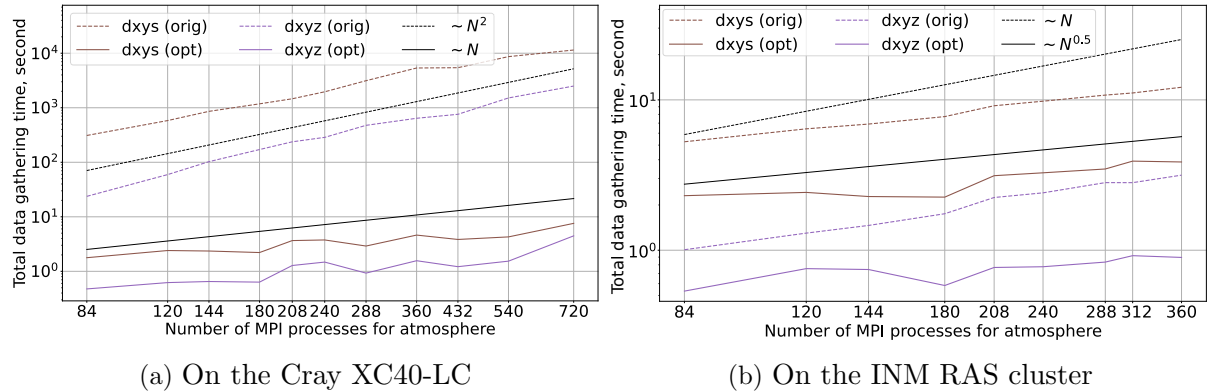
After optimizing gathering of both 3D and 2D arrays the total model running time on Cray XC40-LC reduced significantly, see Fig. 5. Gathering data is not a bottleneck anymore and output is bound by writing time now.

Let us discuss how different optimizations affect the gathering time for each type of the output.

#### 4.1.1. Optimization of 3D output gathering

Optimization of 3D output was done by replacing looped GATHER2D by GATHER3D version, as shown in Listing 3. After this optimization the total gathering time of 3D fields reduced. Figure 6 demonstrates that on Cray XC40-LC the reduction was nearly  $10^3$  times, while on the INM RAS cluster it was only about 3–4 times. Total 3D gathering time was reduced below 10 seconds on both systems and does not exceed 2 % of the total running time.

It is worth noting that not only the gathering time was reduced, but also its asymptotic behavior changed: for Cray XC40-LC it dropped from  $O(nproc^2)$  to sublinear and for INM RAS cluster it dropped from approximately linear to approximately  $O(\sqrt{nproc})$ .



**Figure 6.** INMCM6M optimized 3D output gathering time against the number of MPI processes for the atmosphere component

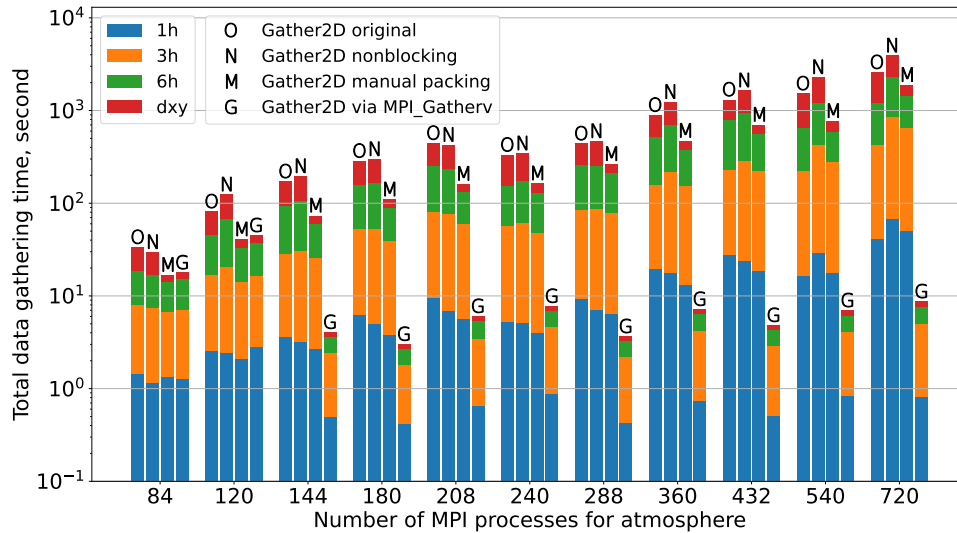
#### 4.1.2. Optimization of 2D output gathering

The timings for 2D output gathering are presented in Fig. 7 and Fig. 8. For Cray XC40-LC the GATHER2D\_OPT which is based on MPI\_GATHERV demonstrates outstanding gathering time drop for more than two orders of magnitude, reducing total 2D gathering time below 10 seconds. For INM RAS cluster there also is some improvement, but not so remarkable.

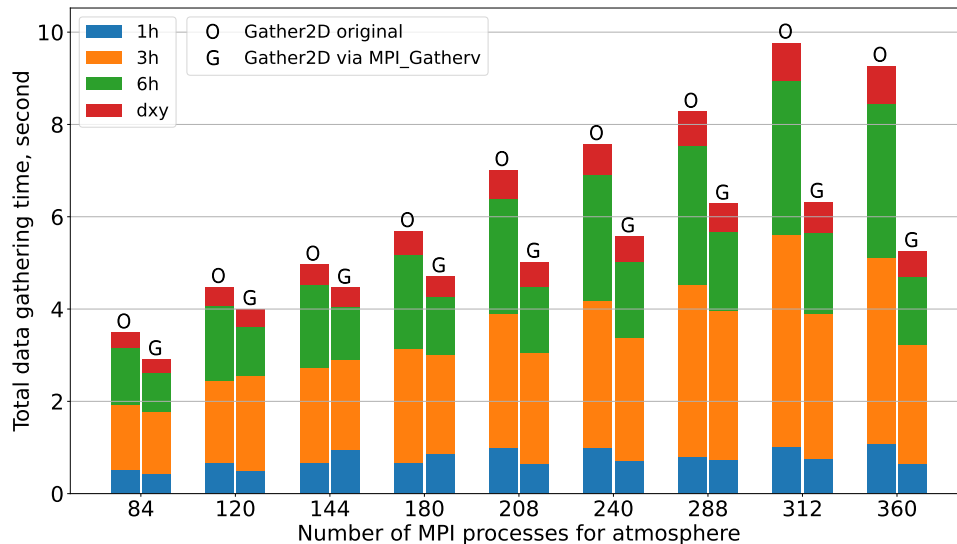
The optimized version of 2D gathering code has approximately equal performance both on Cray XC40-LC and INM RAS cluster, with the latter being a little bit faster, probably due to its smaller size and simpler interconnect topology.

The MPI\_GATHERV based optimization of GATHER2D was the first we have tried. Since MPI\_GATHERV cannot be used with heterogeneous derived types, this implementation internally relies on manual packing. To study which part of the implementation is responsible for the speedup, we considered two additional implementations:

- GATHER2D\_NONBLOCKING – similar to GATHER2D, but using MPI\_IRecv and MPI\_Waitall instead of sequential MPI\_Recv;



**Figure 7.** INMCM6M 2D output gathering time against the number of MPI processes for the atmosphere component on Cray XC40-LC. Note, that total time axis is in logarithmic scale



**Figure 8.** INMCM6M 2D output gathering time against the number of MPI processes for the atmosphere component on INM RAS cluster

- `GATHER2D MANUAL PACKING` – similar to `GATHER2D_OPT` except that it uses `MPISend`, `MPI_Irecv` and `MPI_Waitall` instead of `MPI_Gatherv` and uses the same manual packing and unpacking routines `PACK2D`, `UNPACK2D` instead of derived MPI types.

The idea behind `GATHER2D NONBLOCKING` is that messages could be processed in arriving order and not in rank order as in `GATHER2D`. However, this implementation showed itself even worse when the number of processes was large and gave only minuscule improvement for small number of processes.

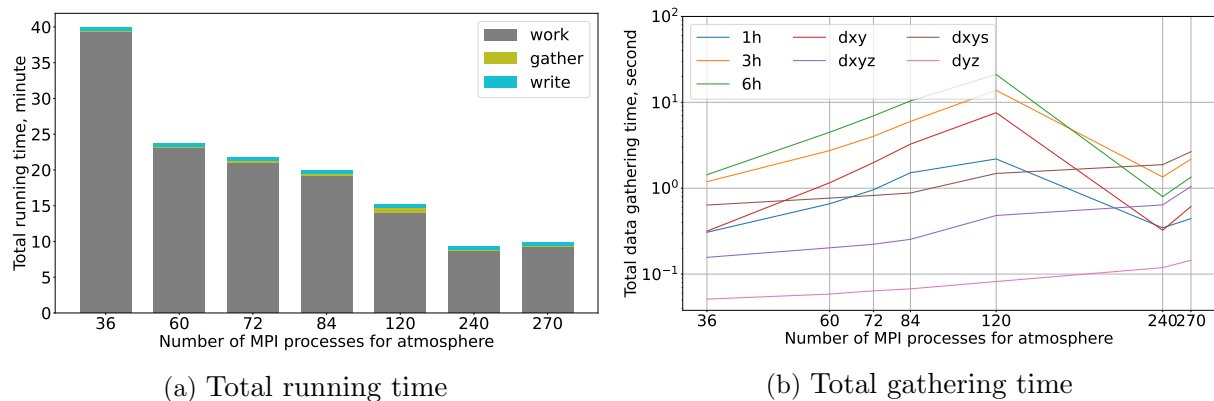
The `GATHER2D MANUAL PACKING` allows to eliminate any possible overhead associated with derived MPI types, for example, committing and freeing them each time we send or receive a message, or reallocating additional buffers in MPI library implementation. This version indeed showed some speedup, especially for `dxy` output, but still was two orders of magnitude slower than `MPI_GATHERV`.



We believe that outstanding MPI\_GATHERV performance is explained by using topology-aware gathering strategy. While in INM RAS cluster all nodes are connected via a single Infini-Band commutator, the Cray XC40-LC has complex interconnect architecture arising from its size. Hence, topology-aware algorithm does not significantly improve gathering performance on INM RAS cluster, but is crucial for large HPC systems like Cray XC40-LC.

## 4.2. INMCM6LM

The results for INMCM6LM are consistent with INMCM6M except for the relative fractions of time spent for `work`, `gather` and `write` parts. Figure 9 shows the effect of output gathering optimization in INMCM6LM.



**Figure 9.** INMCM6LM running wall time and gathering time against the number of MPI processes for the atmosphere component after optimization

For the INMCM6LM version the fraction of time spent in `write` is greater than that for the INMCM6M version. For example, when using 240 MPI processes in atmosphere with optimized gathers on Cray XC40-LC, INMCM6M spends 3.8 % of wall time in `write` while INMCM6LM spends 5.2 %. And for INM RAS cluster `write` takes three times longer due to less efficient file system. Therefore optimizing `write` performance is more important for INMCM6LM than for INMCM6M and for INM RAS cluster than for Cray XC40-LC.

## Conclusion

Investigating the reasons behind INMCM6M slowdown on Cray XC40-LC with enabled daily data output we found a problem in distributed array gathering implementation. The naive algorithm that worked almost flawlessly for years on different HPC systems became a serious bottleneck when the model was ported to Cray XC40-LC.

For 3D data it was sufficient to replace repeated 2D gatherings with an implementation that sends all vertical levels at once. For 2D data the only way to overcome the bottleneck was to adapt an appropriate collective routine MPI\_GATHERV from Cray's MPI library. We believe the huge performance difference is explained by a topology-aware algorithm used in the MPI library implementation. This leads to a conclusion that existing library collective operations should be preferred over the custom ones. The library versions guarantee some uniform performance across different HPC systems, while custom implementations might suddenly break after porting to a new system.

After the problem with data gathering is fixed, the next output bottleneck is serial data writing itself. Our future work is focused on switching to parallel data input and output. And though this is a reasonable thing to do on a distributed file system like Lustre on Cray XC40-LC, it is hard to tell in advance whether it would speed up output on HPC systems with a shared network drive, like INM RAS cluster.

## Acknowledgements

The research was carried out at the Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences. The analysis of INMCM6M output performance and output optimization presented in Sections 2.1, 3 and 4.1 was supported by the Russian Federation research and technical development program in ecological strategy and climate change through grant FFMG-2023-0001 “Development of an extended version of the INM RAS Earth system model within a new computational framework”. All results for the INMCM6LM described in Sections 2.2 and 4.2 were obtained with the support of the Russian Science Foundation, Project 20-17-00190. All computations were performed using the HPC system of the Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences and Cray XC40-LC HPC system at the MCC of Roshydromet. Preliminary analysis of INMCM output performance was carried out using MVS1Q1 at the Joint SuperComputer Center of the Russian Academy of Sciences.

*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. Acosta, M.C., Palomas, S., Paronuzzi Ticco, S.V., *et al.*: The computational and energy cost of simulation and storage for climate science: lessons from CMIP6. *Geoscientific Model Development Discussions* 2023, 1–21 (2023). <https://doi.org/10.5194/gmd-2023-188>
2. Balaji, V., Maisonnave, E., Zadeh, N., *et al.*: CPMIP: measurements of real computational performance of Earth system models in CMIP6. *Geoscientific Model Development* 10(1), 19–34 (2017). <https://doi.org/10.5194/gmd-10-19-2017>
3. Balay, S., Gropp, W.D., McInnes, L.C., *et al.*: Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In: Arge, E., Bruaset, A.M., Langtangen, H.P. (eds.) *Modern Software Tools for Scientific Computing*, pp. 163–202. Birkhäuser Boston, Boston, MA (1997). [https://doi.org/10.1007/978-1-4612-1986-6\\_8](https://doi.org/10.1007/978-1-4612-1986-6_8)
4. Blagodatskikh, D.V.: Comparison of computational efficiency of two versions of a terrain-following ocean climate model. *Numerical Methods and Programming* 24, 440–449 (2023). <https://doi.org/10.26089/NumMet.v24r430>
5. Blagodatskikh, D.V., Iakovlev, N.G., Volodin, E.M., *et al.*: Non-local discretization of the isoneutral diffusion operator in a terrain-following climate ocean model. *Russian Journal of Numerical Analysis and Mathematical Modelling* 38(6), 1–8 (2023). <https://doi.org/10.1515/rnam-2023-0026>





6. Bock, L., Lauer, A., Schlund, M., *et al.*: Quantifying Progress Across Different CMIP Phases With the ESMValTool. *Journal of Geophysical Research: Atmospheres* 125(21), e2019JD032321 (2020). <https://doi.org/10.1029/2019JD032321>
7. Chen, D., Rojas, M., Samset, B.H., *et al.*: Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. In: Masson-Delmotte, V., Zhai, P., Pirani, A., *et al.* (eds.) *Climate Change 2021: The Physical Science Basis*, pp. 147–286. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA (2021). <https://doi.org/10.1017/9781009157896.003>
8. Emelina, S.V., Khan, V.M., Semenov, V.A., *et al.*: Seasonal Hydrodynamic Forecasts Using the INM-CM5 Model for Estimating the Beginning of Birch Pollen Dispersion. *Izv. Atmos. Ocean. Phys.* 59, 351359 (2023). <https://doi.org/10.1134/S0001433823040059>
9. Eyring, V., Bony, S., Meehl, G.A., *et al.*: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development* 9(5), 1937–1958 (2016). <https://doi.org/10.5194/gmd-9-1937-2016>
10. Gloukhov, V.: Parallel implementation of the INM atmospheric general circulation model on distributed memory multiprocessors. In: Sloot, P.M.A., Hoekstra, A.G., Tan, C.J.K., Dongarra, J.J. (eds.) *Computational Science – ICCS 2002. Lecture Notes in Computer Science*, vol. 2329, pp. 753–762. Springer Berlin, Heidelberg (2002). <https://doi.org/10.1007/3-540-46043-8>
11. Haarsma, R.J., Roberts, M.J., Vidale, P.L., *et al.*: High Resolution Model Intercomparison Project (HighResMIP v1.0) for CMIP6. *Geoscientific Model Development* 9(11), 4185–4208 (2016). <https://doi.org/10.5194/gmd-9-4185-2016>
12. Hoskins, B.: The potential for skill across the range of the seamless weather-climate prediction problem: a stimulus for our science. *Quarterly Journal of the Royal Meteorological Society* 139(672), 573–584 (2013). <https://doi.org/10.1002/qj.1991>
13. Khan, V.M., Vilfand, R., Tishchenko, V., *et al.*: Assessment of Changes in the Temperature Regime of Northern Eurasia for the Next Five Years According to the INM RAS Earth System Model Forecasts and Their Possible Consequences for Agriculture. *Russ. Meteorol. Hydrol.* 48, 745754 (2023). <https://doi.org/10.3103/S1068373923090029>
14. Kim, Y.H., Min, S.K., Zhang, X., *et al.*: Evaluation of the CMIP6 multi-model ensemble for climate extreme indices. *Weather and Climate Extremes* 29, 100269 (2020). <https://doi.org/10.1016/j.wace.2020.100269>
15. Kulikova, I.A., Nabokova, E.V., Khan, V.M., *et al.*: Madden-Julian Oscillation in the Context of Subseasonal Variability, Teleconnections, and Predictability. *Russ. Meteorol. Hydrol.* 48, 645657 (2023). <https://doi.org/10.3103/S1068373923080010>
16. Stocker, T.F., Qin, D., Plattner, G.K., *et al.* (eds.): IPCC, 2013: *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, p. 1535. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA (2013), [https://www.ipcc.ch/site/assets/uploads/2018/02/WG1AR5\\_all\\_final.pdf](https://www.ipcc.ch/site/assets/uploads/2018/02/WG1AR5_all_final.pdf)

17. Tarasevich, M., Sakhno, A., Blagodatskikh, D., *et al.*: Scalability of the INM RAS Earth System Model. In: Voevodin, V., Sobolev, S., Yakobovskiy, M., *et al.* (eds.) Supercomputing. pp. 202–216. Lecture Notes in Computer Science, Springer Nature Switzerland (2023). [https://doi.org/10.1007/978-3-031-49432-1\\_16](https://doi.org/10.1007/978-3-031-49432-1_16)
18. Tarasevich, M.A., Tsybulin, I.V., Onoprienko, V.A., *et al.*: Ensemble-based statistical verification of INM RAS Earth system model. Russian Journal of Numerical Analysis and Mathematical Modelling 38(3), 173–186 (2023). <https://doi.org/10.1515/rnam-2023-0014>
19. Tarasevich, M.A., Volodin, E.M.: Influence of various parameters of INM RAS climate model on the results of extreme precipitation simulation. In: International Young Scientists School and Conference on Computational Information Technologies for Environmental Sciences, May 27 – June 6, 2019. IOP Conference Series: Earth and Environmental Science, vol. 386, p. 012012. IOP Publishing (2019). <https://doi.org/10.1088/1755-1315/386/1/012012>
20. Tarasevich, M.A., Volodin, E.M.: The Influence of Autumn Eurasian Snow Cover on the Atmospheric Dynamics Anomalies during the Next Winter in INMCM5 Model Data. Supercomputing Frontiers and Innovations 8(4), 24–39 (2021). <https://doi.org/10.14529/jsfi210403>
21. Terekhov, K.M., Volodin, E.M., Gusev, A.V.: Methods and efficiency estimation of parallel implementation of the  $\sigma$ -model of general ocean circulation. Russ. J. Numer. Anal. Math. Modelling 26(2), 189–208 (2011). <https://doi.org/10.1515/rjnamm.2011.011>
22. Volodin, E.M.: Simulation of Present-Day Climate with the INMCM60 Model. Izvestiya, Atmospheric and Oceanic Physics 59(1), 16–22 (2023). <https://doi.org/10.1134/S0001433823010139>
23. Volodin, E.M., Gritsun, A.S.: Simulation of observed climate changes in 1850–2014 with climate model INM-CM5. Earth System Dynamics 9(4), 1235–1242 (2018). <https://doi.org/10.5194/esd-9-1235-2018>
24. Volodin, E.M., Kostykin, S.V.: The aerosol module in the INM RAS climate model. Russian Meteorology and Hydrology 41(8), 519–528 (2016). <https://doi.org/10.3103/S106837391608001X>
25. Volodin, E.M., Lykosov, V.N.: Parametrization of Heat and Moisture Transfer in the Soil-Vegetation System for Use in Atmospheric General Circulation Models: 1. Formulation and Simulations Based on Local Observational Data. Izvestiya, Atmospheric and Oceanic Physics 34(4), 405–416 (1998), [https://www.researchgate.net/publication/270586916\\_Parameterization\\_of\\_Heat\\_and\\_Moisture\\_Transfer\\_in\\_the\\_Soil-Vegetation\\_System\\_for\\_Use\\_in\\_Atmospheric\\_General\\_Circulation\\_Models\\_1\\_Formulation\\_and\\_Simulations\\_Based\\_on\\_Local\\_Observational\\_Data](https://www.researchgate.net/publication/270586916_Parameterization_of_Heat_and_Moisture_Transfer_in_the_Soil-Vegetation_System_for_Use_in_Atmospheric_General_Circulation_Models_1_Formulation_and_Simulations_Based_on_Local_Observational_Data)
26. Volodin, E.M., Lykosov, V.N.: Parametrization of Heat and Moisture Transfer in the Soil-Vegetation System for Use in Atmospheric General Circulation Models: 2. Numerical Experiments in Climate Modeling. Izvestiya, Atmospheric and Oceanic Physics 34(5), 559–569 (1998), [https://www.researchgate.net/publication/270586932\\_Parameterization\\_of\\_Heat\\_and\\_Moisture\\_Transfer\\_in\\_the\\_Soil-Vegetation\\_System\\_for\\_Use\\_in](https://www.researchgate.net/publication/270586932_Parameterization_of_Heat_and_Moisture_Transfer_in_the_Soil-Vegetation_System_for_Use_in)

Atmospheric\_General\_Circulation\_Models\_2\_Numerical\_Experiments\_in\_Climate\_Modeling

27. Volodin, E.M., Mortikov, E.V., Kostykin, S.V., *et al.*: Simulation of modern climate with the new version of the INM RAS climate model. *Izvestiya, Atmospheric and Oceanic Physics* 53(2), 142–155 (2017). <https://doi.org/10.1134/S0001433817020128>
28. Volodin, E.M., Mortikov, E.V., Kostykin, S.V., *et al.*: Simulation of the present-day climate with the climate model INMCM5. *Climate Dynamics* 49(11), 3715–3734 (2017). <https://doi.org/10.1007/s00382-017-3539-7>
29. Volodin, E.M., Mortikov, E.V., Kostykin, S.V., *et al.*: Simulation of the modern climate using the INM-CM48 climate model. *Russian Journal of Numerical Analysis and Mathematical Modelling* 33(6), 367–374 (2018). <https://doi.org/10.1515/rnam-2018-0032>
30. Volodin, E.M., Tarasevich, M.A.: Simulation of Climate and Weather Extreme Indices with the INM-CM5 Climate Model. *Russian Meteorology and Hydrology* 43(11), 756–762 (2018). <https://doi.org/10.3103/S1068373918110067>
31. Volodin, E.M., Vorobyeva, V.V.: On the multi-annual potential predictability of the Arctic Ocean climate state in the INM RAS climate model. *Russian Journal of Numerical Analysis and Mathematical Modelling* 37(2), 119–129 (2022). <https://doi.org/10.1515/rnam-2022-0010>
32. Vorobeva, V.V., Volodin, E.M., Gritsun, A.S., *et al.*: Analysis of the Atmosphere and the Ocean Upper Layer State Predictability for up to 5 Years Ahead Using the INMCM5 Climate Model Hindcasts. *Russian Meteorology and Hydrology* 48(7), 581–589 (2023). <https://doi.org/10.3103/S106837392307004X>
33. Vorobyeva, V., Volodin, E.: Analysis of the predictability of stratospheric variability and climate indices based on seasonal retrospective forecasts of the INM RAS climate model. *Russian Journal of Numerical Analysis and Mathematical Modelling* 36(2), 117–126 (2021). <https://doi.org/10.1515/rnam-2021-0010>
34. Vorobyeva, V., Volodin, E.: Evaluation of the INM RAS climate model skill in climate indices and stratospheric anomalies on seasonal timescale. *Tellus A: Dynamic Meteorology and Oceanography* 73(1), 1–12 (2021). <https://doi.org/10.1080/16000870.2021.1892435>
35. Vorobyeva, V.V., Volodin, E.M.: Experimental Studies of Seasonal Weather Predictability Based on the INM RAS Climate Model. *Mathematical Models and Computer Simulations* 13(4), 571–578 (2021). <https://doi.org/10.1134/S2070048221040232>
36. Yakovlev, N.G.: Reproduction of the large-scale state of water and sea ice in the Arctic Ocean from 1948 to 2002: Part II. The state of ice and snow cover. *Izvestiya, Atmospheric and Oceanic Physics* 45(4), 478–494 (2009). <https://doi.org/10.1134/S0001433809040082>
37. Yakovlev, N.G.: Reproduction of the large-scale state of water and sea ice in the Arctic Ocean in 1948–2002: Part I. Numerical model. *Izvestiya, Atmospheric and Oceanic Physics* 45(3), 357–371 (2009). <https://doi.org/10.1134/S0001433809030098>

# High-Performance Computing of Microtubule Protofilament Dynamics by Means of All-Atom Molecular Modeling

Vladimir A. Fedorov<sup>1,2</sup> , Ekaterina G. Kholina<sup>1</sup> ,  
Nikita B. Gudimchuk<sup>1</sup> , Ilya B. Kovalenko<sup>1</sup> 

© The Authors 2023. This paper is published with open access at SuperFri.org

Molecular dynamics (MD) simulation is a useful tool for understanding biological systems at the level of individual molecules and atoms. However, studying such massive biological systems as microtubules and even their constituent components (tubulin protofilaments) takes an enormous amount of processing power. In this paper, using MD calculations of individual microtubule protofilaments, we demonstrate how computational architecture and calculation options affect computing performance. When using the “GPU-resident” option in the GROMACS MD package, you may gain a fantastic computation acceleration by using the newest high-end graphics processing unit (GPU), even in conjunction with a rather outdated central processing unit (CPU). For instance, MD of the biomolecular system containing a tubulin protofilament in an explicitly specified solvent consisting of more than 300 thousand atoms can be investigated with performance of 171 ns/day at time step 2 fs when using a single-node computer with the latest CPU and GPU generation architecture (Intel Core i9-13900K and Nvidia RTX4090 respectively). Nevertheless, high performance computing platforms (e.g., the volta2 partition of “Lomonosov-2” supercomputer) can be very suitable for simulation experiments with a large number of independent calculations, such as the umbrella sampling technique. Obtained results allow one to choose the best price-performance solution to study molecular dynamics of biological systems.

*Keywords: molecular dynamics, tubulin, microtubule, CPU, GPU, computing performance.*

## Introduction

Microtubules are polymers of the tubulin protein that are involved in many critical functions in living cells. One of these functions is the separation of genetic material during cell division (the chromosomes segregation). Search and attachment to chromosomes becomes possible due to an amazing property of microtubules, namely, dynamic instability, their ability to spontaneously polymerize and depolymerize [4]. Despite decades of research, the detailed molecular mechanisms underlying this process remain unknown. The experiments and classical theoretical approaches provide particular facts without explaining the behavior of the entire microtubule at the molecular level.

A powerful tool for studying of biological systems functioning at the level of individual molecules and atoms is molecular dynamics (MD) simulation. However, the study of such large systems as a microtubule and even its individual components (tubulin protofilaments) requires extremely high computational resources. The question arises about choosing the optimal computational architecture for studying such systems using the MD method. Computational capabilities are improved every year with new generations of CPUs and GPUs, so it is important to know what computational capabilities each computer configuration provides. In addition, software packages for MD calculations are being improved by adding options increasing the performance of MD calculations. In this article, using MD calculations of individual microtubule protofilaments, we demonstrate how the choice of computational architecture and calculation options affects their performance.

<sup>1</sup>Lomonosov Moscow State University, Moscow, Russia

<sup>2</sup>Center for Theoretical Problems of Physico-Chemical Pharmacology, Russian Academy of Sciences, Moscow, Russia

The article is organized as follows. Section 1 is devoted to the computing method of all atom molecular dynamics. In Section 2, we described the results comparing the performance of various computing systems and various parameters for molecular dynamics calculations. Conclusion summarizes the study and allows one to make the best choice of MD parameters and computational architecture.

## 1. Methods

All tests were conducted using the all-atom explicit solvent MD. The calculations were executed using the GROMACS 2022.4 software package [2] which facilitates parallel computing on hybrid architectures and incorporates the CHARMM27 force field [6]. Each benchmark simulation was run for a duration of 30 minutes, employing the TIP3P water model. The initial structure of the taxol-bound tubulin tetramer was retrieved from the Protein Data Bank (PDB id 5SYF [5]). The dimensions of the simulation volume were selected to ensure that the distance from the protein’s surface to the nearest boundary of the simulation box was never less than two nanometers. Long-range electrostatic interactions were accounted for using the particle mesh Ewald method [3]. Both Coulomb and Lennard-Jones cutoffs were configured to 1.25 nm. For the water box systems, the time step was chosen to be 1 fs and no restraints were used. For the tubulin protofilament, the time step was chosen to be 2 or 4 fs.

In this work, we used two types of integrators: stochastic dynamics (sd) and molecular dynamics (md). Sd is an accurate and efficient leap-frog stochastic dynamics integrator with constraints; coordinates need to be constrained twice per integration step which can take a significant part of the simulation time. Md is a leap-frog algorithm for integrating Newtons equations of motion. When calculating with an integration step of 2 fs, the constraints were imposed on the bonds of atoms with hydrogens; when calculating with an integration step of 4 fs, they were applied to all bonds. Also, for all systems, when using the md integrator with a step of 1 or 2 fs, the “GPU-resident” option was tested, which updates coordinates on the GPU. This parallelization mode is referred to as “GPU-resident” as all force and coordinate data remain resident on the GPU for a number of steps. The “GPU-resident” scheme however is still able to carry out part of the computation on the CPU concurrently with GPU calculation which helps to support the broad range of GROMACS features not all of which are ported to GPU and also allows improving performance by making use of the otherwise mostly idle CPU [1]. Specifications of MD systems used for benchmarking are summarized in Tab. 1.

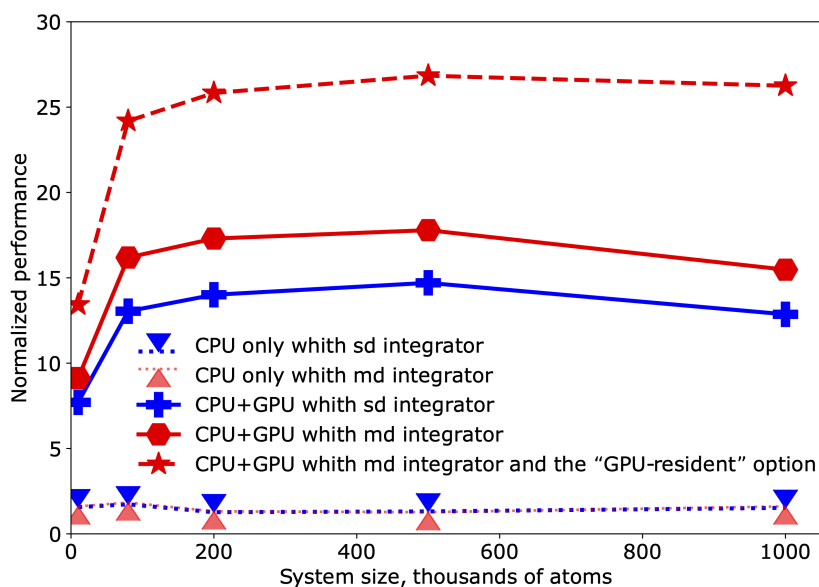
**Table 1.** Specification of molecular dynamics systems used in the benchmarks

MD systems	MD system name	Number of atoms	System size, nm
Water box (WB)	WB-10	10,206	$4.7 \times 4.7 \times 4.7$
	WB-80	80,232	$9.3 \times 9.3 \times 9.3$
	WB-200	203,415	$12.7 \times 12.7 \times 12.7$
	WB-500	500,076	$17.2 \times 17.2 \times 17.2$
	WB-1000	1,000,005	$21.7 \times 21.7 \times 21.7$
Tubulin tetramer	Tub-4	307,453	$11.3 \times 12.5 \times 22.2$

## 2. Results and Discussion

### 2.1. System Size Has Very Little Impact on Performance

To find out how calculation performance depends on the system size, we have tested the performance of all-atom MD calculations for water box systems of different sizes: 10, 80, 200, 500, and 1000 thousands of atoms. The calculations were carried out for CPU only and CPU+GPU computer architectures both with sd or md integrator type. The table with all performance tests is available via this link. The results for Intel Core i9-13900K CPU and RTX4090 GPU with “GPU-resident” option are presented in Fig. 1. We normalized the performance to the value for the largest WB1000 system in order to compare MD simulation results across systems of various sizes. We can see that, with the exception of a system of ten thousand atoms on CPU+GPU computer architectures, normalized performance practically never changes with system size. The ineffective use of GPU computing resources can be the cause of the system’s relatively poor normalized performance for a small system (less than 80,000 atoms). However, biological systems of interest typically contain larger numbers of atoms. On the other hand, small systems are already computationally fast, the computing performance of a system of 10,000 atoms reaches  $1.3 \mu\text{s}/\text{day}$ . So, the actual performance depends linearly on system size, and there is no saturation, at least for molecular systems up to 1 million of atoms.



**Figure 1.** Normalized performance of MD calculation for systems of different sizes both for CPU only and CPU+GPU computer systems. Sd integrator type is colored blue, md – red

### 2.2. MD Performance is Always Slightly Greater with Md Integrator than with Sd Integrator

We compared MD calculation performance with two integration algorithms, sd and md (Fig. 1). For all our MD calculations, md algorithm gave slightly better performance than sd algorithm. For CPU calculations, the performance of the md algorithm compared to sd was better by no more than 6% (Tab. 2, WB-80 system). For GPU calculations, md algorithm shows



up to 24% better performance than sd (Tab. 2, WB-80 system). Note that the “GPU-resident” option cannot be used with the sd algorithm.

**Table 2.** Single-node performance (ns/day) for systems of different size depending on various combinations of CPUs and GPUs and protocol of MD calculation

system	Intel Core CPU model	sd CPU	sd CPU+GPU	md CPU	md CPU+GPU	md (CPU+GPU) with the GPU-resident option
WB-10	i9-13900K	156	770	161	913	1343
WB-80	i9-13900K	21.6	163	23	202	302
WB-200	i9-13900K	6.35	70	6.58	86.5	129
WB-500	i9-13900K	2.63	29.4	2.54	35.6	53.7
WB-1000	i9-13900K	1.52	12.9	1.61	15.5	26.3
WB-1000	i9-7900X	0.85	6.53	0.89	9.23	21.6

### 2.3. “GPU-resident” Option Gives More than Two Times Better Performance

We applied the “GPU-resident” option and it significantly improved MD calculation performance. The larger the molecular system, the greater acceleration was given by this option. For an MD system consisting of a million atoms, the option gave up to 2.3 times better calculation performance (Tab. 2, WB-1000 system, Intel Core i9-7900X processor). Note that the option is not available with sd integrator.

### 2.4. Latest GPU Gives High Performance Even with Old CPU if the “GPU-resident” Option is Applicable

Surprisingly, newer and more powerful video cards in combination with a central processor from one of the previous generations provided the greatest performance gain. Indeed, the “GPU-resident” option on a computer with 5 years old Intel Core i9-7900X CPU and the newest RTX4090 GPU for an MD system consisting of a million atoms, gave a 2.3 times performance increase, while on the latest Intel Core i9-13900K CPU + RTX4090 GPU it gave only 1.7 times acceleration (Tab. 2, WB-1000 system). This means that there is no need to upgrade the whole computer, it is enough to upgrade only the GPU and this will be the best solution in terms of price-performance ratio, especially if we take into account that the top-end GPU does not exceed 40% of the cost of a new computer.

### 2.5. Difference in GPUs is Much More Important than Difference in CPUs

CPU upgrade itself from 5 years old Intel Core i9-7900X to the newest Intel Core i9-13900K with RTX2080ti gave only 1.02 times acceleration in the case of tubulin tetramer MD simulation with “GPU-resident” option, while GPU upgrade from 5 years old RTX2080ti to the latest RTX4090 gives up to 2.7 times performance increase (Tab. 2). So the difference in GPUs is

much more important than difference in CPUs when the “GPU-resident” option is available, and there is no sense in putting an old GPU in the newest computer.

**Table 3.** Single-node performance (ns/day) for tubulin tetramer depending on various combinations of CPUs and GPUs and protocol of MD calculation

time step, fs	integrator	GPU-resident	CPU	GPU	Performance, ns/day
4	sd	No	i9-7900X	RTX2080ti	56.7
4	sd	No	i9-13900K	RTX2080ti	77.0
4	sd	No	i9-7900X	RTX4090	77.1
4	sd	No	i9-13900K	RTX4090	171
4	md	No	i9-7900X	RTX2080ti	69.8
4	md	No	i9-13900K	RTX2080ti	82.4
4	md	No	i9-7900X	RTX4090	116
4	md	No	i9-13900K	RTX4090	211
2	md	No	i9-7900X	no GPU	6.8
2	md	No	i9-13900K	no GPU	12.1
2	md	Yes	i9-7900X	RTX2080ti	46.6
2	md	Yes	i9-13900K	RTX2080ti	47.5
2	md	Yes	i9-7900X	RTX4090	127
2	md	Yes	i9-13900K	RTX4090	171

## 2.6. Later GPU and CPU Generations Provide Improved Performance

Obviously, a computer with the latest CPU and GPU outperforms any other configuration, so if you need to perform MD calculations at peak performance, it is recommended that you upgrade the entire computer. Compared to the five-year-old i9-7900X CPU + RTX2080ti GPU configuration, the latest i9-13900K CPU + RTX4090 GPU configuration with “GPU-resident” option enabled delivers  $3.7\times$  faster tubulin tetramer MD simulation (Tab. 3). At the same time, if you upgrade only the GPU, then the calculations will accelerate only 2.7 times. However, if you must use the sd integrator type, the choice of computing architecture with the best price-performance ratio is to use the latest CPU and GPU. In this case, upgrading only the GPU gives a performance increase of 1.4 times, and if the entire computer is upgraded, the performance will increase by 3 times. Further increase in the productivity of molecular dynamics calculations is possible by increasing the integration step to 4 fs. This, in turn, does not allow the use of the “GPU-resident” option, so in this situation the performance will not increase in proportion to the integration step, that is, by 2 times. Productivity in this case will increase only 1.2 times. Interestingly, when using the sd integrator, GPU upgrade from RTX2080ti to RTX4090 is equivalent to CPU upgrade from i9-7900X to i9-13900K (Tab. 3). When using a computer with the modern architecture (latest GPU and CPU), the sd integrator is two times slower than the md integrator with “GPU-resident” option.

## 2.7. Lomonosov-2 Supercomputer Architecture Test

We also tested “Volta2” section of Lomonosov-2 supercomputer launched in 2018. Using just the CPU on a single node, we were able to achieve 12.8 ns/day performance (Tab. 4), which is comparable to the latest Intel i9-13900K CPU and more than double the performance of our i9-7900X test processor released in 2018 (Tab. 3). However, if four nodes are used in parallel in CPU-only mode, then performance increases by 2.3 times (Tab. 4).

When using one supercomputer node in CPU+GPU mode, the performance of MD calculations became an impressive 62 ns/day, that is, using GPU speeds up calculations by almost 5 times. However, the modern combination of i9-13900K CPU + RTX4090 GPU provides performance of 114 ns/day, which is almost 2 times faster (Tab. 3). Moreover, enabling the “GPU-resident” option increases the performance of the latter system to the stunning 171 ns/day, whereas the supercomputer node gave only 34.1 ns/day when this option was enabled. Thus, the GPU-resident option provides the greatest performance benefit when using a relatively old CPU and a newer GPU. However, when combined with powerful CPUs and relatively outdated GPUs, this option causes an imbalance of calculations owing to higher data interchange within the system, resulting in a drop in system performance.

**Table 4.** Performance of “Lomonosov-2” supercomputer (ns/day), depending on the number of computing nodes for MD simulations of tubulin tetramer with md integrator and 2 fs time step with “GPU-resident” option

Number of nodes	GPU	GPU-resident	Performance, ns/day
1	no GPU	No	12.8
4	no GPU	No	29.7
1	2×NVIDIA Tesla V100	No	62.1
4	2×NVIDIA Tesla V100	No	58.7
5	2×NVIDIA Tesla V100	No	96.4
1	2×NVIDIA Tesla V100	Yes	34.1
4	2×NVIDIA Tesla V100	Yes	57.1
5	2×NVIDIA Tesla V100	Yes	81.8

## Conclusion

Our comparative performance analysis of GPU-based architectures for all-atom MD simulations of realistic biomolecular systems provides the guidelines for selection of the best computing solution in terms of price-performance ratio. When using the “GPU-resident” option in the GRO-MACS MD package, you may gain fantastic computation acceleration by employing a current GPU, even in conjunction with a rather outdated CPU. Namely, changing only the GPU allows you to gain nearly three times quicker performance. However, if you also update the CPU, this will speed up the calculation by almost 4 times. MD of the biomolecular system containing a tubulin protofilament in an explicitly specified solvent consisting of more than 300 thousand atoms can be studied with performance of 171 ns/day at time step 2 fs when using single-node computer with the latest CPU and GPU generation architecture (Intel Core i9-13900K and Nvidia RTX4090 respectively). Nevertheless, HPC platforms (e.g., the volta2 partition of

“Lomonosov-2” supercomputer) can be very suitable for simulation experiments with a large number of independent calculations, such as the umbrella sampling technique. For calculations where one task is computed on one node, it would give effective performance in linear proportion with the number of nodes, i.e., it is possible to achieve more than 1  $\mu$ s/day calculation performance for a tubulin tetramer using the entire volta2 partition (18 nodes), which exceeds the capabilities of even the most advanced one-node computer by an order of magnitude.

## Acknowledgments

The study was supported by the Russian Science Foundation, project No. 22-74-00119, <https://rscf.ru/en/project/22-74-00119/>. GPUs Nvidia RTX4090 were purchased under the Development Program of Moscow State University (order No. 99 of February 6, 2023, contract No. 0183-44-2023).

*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., Alekseenko, A., Bergh, C., *et al.*: Gromacs 2023.2 manual (Jul 2023). <https://doi.org/10.5281/zenodo.8134388>
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, 19–25 (2015). <https://doi.org/10.1016/j.softx.2015.06.001>
3. Ewald, P.P.: Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der physik* 369(3), 253–287 (1921). <https://doi.org/10.1002/andp.19213690304>
4. 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>
5. Kellogg, E.H., Hejab, N.M., Howes, S., *et al.*: Insights into the distinct mechanisms of action of taxane and non-taxane microtubule stabilizers from cryo-em structures. *Journal of molecular biology* 429(5), 633–646 (2017). <https://doi.org/10.1016/j.jmb.2017.01.001>
6. MacKerell Jr, A.D., Feig, M., Brooks, C.L.: Improved treatment of the protein backbone in empirical force fields. *Journal of the American Chemical Society* 126(3), 698–699 (2004). <https://doi.org/10.1021/ja036959e>