# Supercomputing Frontiers and Innovations

2024, Vol. 11, No. 2

## 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

### Editorial Director

### Associate Editors

# Contents

DOI: 10.14529/jsfi240201

# Validation Test of Parallelized Codes in the Study of Flow and Heat Transfer Anomalous Enhancement in a Single Inclined Groove on a Plate

*Sergey A. Isaev*[1,2], *Alina A. Klyus*[2], *Alexandr G. Sudakov*[2], *Dmitry V. Nikushchenko*[1], *Alexandr E. Usachov*[3], *Vladimir V. Seroshtanov*[4], *Alexei Yu. Chulyunin*[5]

A validation test is presented based on the study of the phenomenon of separated turbulent air flow and anomalous enhancement of heat transfer in an inclined groove on a heated plate. The groove is made up of two halves of a spherical dimple with a spot diameter of 0.25, and it is connected by a trench insert that is 5 long. The generation of tornadoes in grooves associated with extraordinary static pressure differences contributes to the formation of fields of ultra-high velocities, high gradients of relative friction, and heat transfer coefficients inside the inclined groove. Databases of heat flux measurements in a groove on the isothermal section of the plate when varying the inclination angle from 0° to 90° were obtained on the SPbPU thermophysical setup. Parametric numerical and physical studies of heat transfer on a plate with a single groove were performed at $Re = 3\times10^4$. Using the parallelized package VP2/3 has resulted in a satisfactory agreement between experimental data and numerical predictions made within the RANS-SST framework. Abnormal heat transfer enhancement in grooves occurs at angles of inclination from 30° to 75°, which correlates with conclusions on analysis of pressure distributions in grooves on the plate. The vortex structure in the groove on the plate at an inclination angle of 45° is illustrated. It is shown that the focused pressure difference between the adjacent stagnation zone on the windward slope and the reduced pressure region at the entrance to the groove initiates a tornado-like vortex and the developing intense swirling flow.

*Keywords: heat transfer enhancement, tornado-like vortex, inclined groove, plate, turbulence, parallelized VP2/3 package.*

## Introduction

Validation tests, which are considered as analogues of experimental problems, serve to configure and verify mathematical models of control processes, in particular, turbulent transport [1], as well as assess the accuracy and efficiency of developed program codes [3]. Recently [7], original validation tests have been proposed based on a study of separated flow enhancement in inclined grooves on structured surfaces of a plate and channel wall. The grooves are two halves of a spherical dimple connected by a long trench cylindrical insert [5]. The entrance spherical segments that are oriented towards the free-stream flow create tornado-like structures that developed into high-intensity flows in separation zones inside the groove and create zones of decreased pressure. Ultra-high velocities of reverse-direction and secondary flow in the grooves turn out to be on the order of the characteristic velocity of the free-stream flow and sometimes exceed it. As a consequence, the generated intense vortex flows inside the grooves form zones of parameter gradients that are anomalously high for separated flows. These zones are characterized by multiple superiorities of the absolute values of negative friction and heat transfer coefficient compared

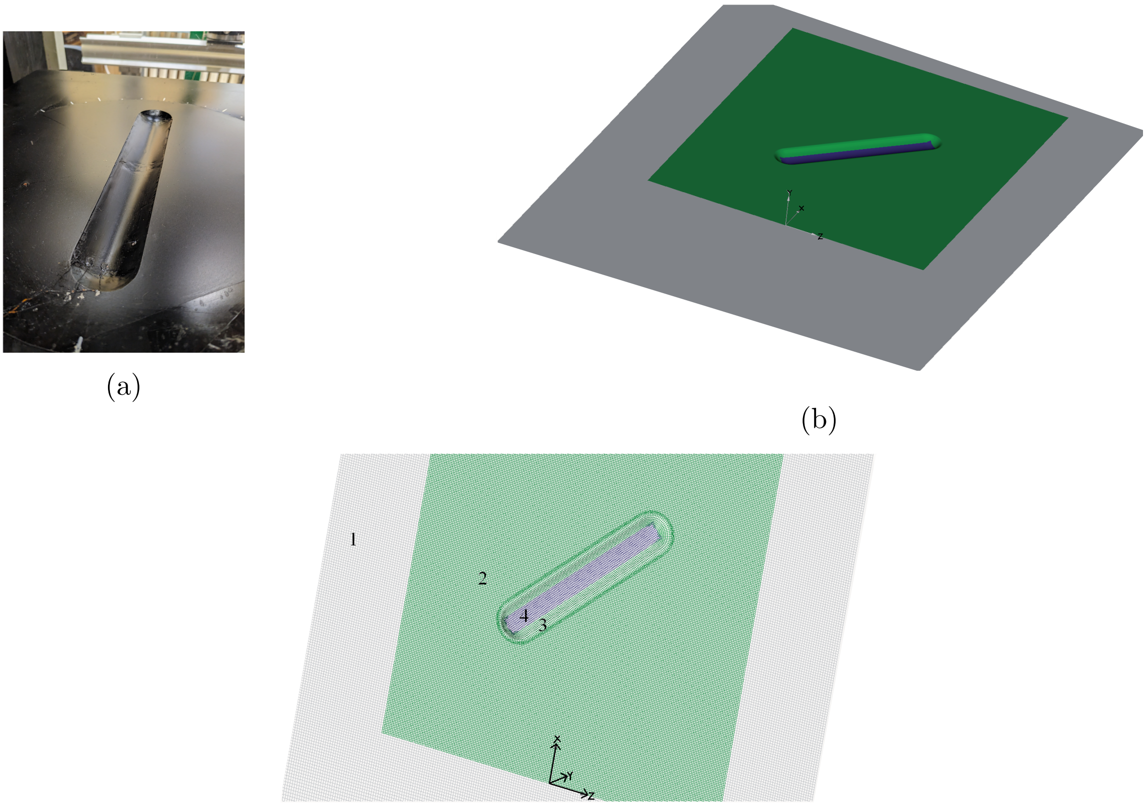
[1] Saint Petersburg State Marine Technical University, Saint Petersburg, Russian Federation
[2] A. Novikov Saint-Petersburg State University of Civil Aviation, Saint Petersburg, Russian Federation
[3] Moscow Research Complex Central Aerohydrodynamic Institute, Moscow, Russian Federation
[4] Peter the Great St.Petersburg Polytechnic University, Saint Petersburg, Russian Federation
[5] Institute of Mechanics, Lomonosov Moscow State University, Moscow, Russian Federation

to the flat walls. A connection has been established between the separated flow and anomalous enhancement of the heat transfer in inclined grooves with extraordinary pressure drops between the regions of deceleration of flow entering the grooves and rarefaction in places where tornado-like vortices are generated [6]. This paper examines tests concerning the anomalous enhancement of heat transfer in the separation zone in the inlet part of an inclined groove on an isothermal heated section in a thermally insulated plate. The experiments were carried out using gradient heatmetry at the thermophysical setup of Peter the Great St. Petersburg Polytechnic University. [14]. Parametric calculations are carried out by multi-block computing technology parallelized code VP2/3 (Velocity-Pressure, 2D/3D) [8] using 24-core Xeon Gold 6562 processor.

The article is organized as follows. Section 1 is devoted to methods of numerical simulation and physical study. The research results and their detailed description are presented in Section 2. Conclusion summarizes the study.

## 1. Methodology. Calculations and Experimental Justification

In a thermophysical setup, a thermally insulated plate with an inclined groove in a heated isothermal region is placed in the subsonic wind tunnel test section (Fig. 1a). The closed-circuit wind tunnel is equipped with a cooling system for the air-flow connected to a cold water supply system, which makes it possible to maintain the temperature of air almost constant (±0.2 K) during long-term experiments.



(a)

(b)

(c)

**Figure 1.** Thermophysical setup with a heated plate and an inclined groove (a), a setup digital twin (b) and a multi-block wall with a Cartesian $x, y, z$ coordinate system (c)

To align the flow, guide vanes are installed in the turns of the tunnel, and a honeycomb is installed in front of the confuser. The experimental models are made in the form of a rotating round plate with a diameter of 0.3 m with a groove under study. The groove consists of two halves of a spherical dimple connected by a trench cylindrical insert. Its width is $D = 0.04$ m, depth is $h = 0.25D$ and the length of the trench insert is $L = 5D$.

The groove is made of a copper sheet 0.002 m thick and soldered into a steel plate 0.001 m thick. From below, a cylindrical box is attached to the plate with groove. The box is equipped with connecting legs for supplying steam and draining condensate. The model is heated using saturated steam supplied from an electric steam generator. The excess pressure in the steam generator does not exceed 200 mm of water and the temperature at the heat transfer surface is kept constant and close to 100 °C. The cylindrical box is installed in a heated rectangular part (with dimensions of $0.475 \times 0.415$ m), located on a thermal insulated flat plate with dimensions of $0.71 \times 0.6$ m. The temperature of the heated surface $T_w$ was monitored using a testo-875 thermal imager. The heated isothermal section is 0.25 m away from the leading edge of the plate, and the center of the groove is located in its middle longitudinal section at a distance of 0.5075 m from its beginning. This approach enabled the stabilization of the hydrodynamic and thermal boundary layers in front of the groove.

A cylindrical box-shaped structure with a groove can be rotated around an axis passing through the center of the groove, which made it possible to change the inclination angle $\theta$ from 0 to 90° to the free-stream velocity vector $U$. The experiments were carried out for the Reynolds number $Re = UD/\nu = 3 \times 10^4$, where $\nu$ is coefficient of kinematic viscosity of air. In studies [6, 7], the Reynolds number is twice as high, but in a separate series of PIV-experiments it was shown that a turbulent flow develops in front of the groove. To determine the relative local heat transfer coefficient in characteristic cross-sections of the groove gradient heat flux sensors (GHFS) were installed [14]. In this study, we used battery-type GHFSs based on ATE made of single-crystal bismuth with a purity of 0.9999. In the experiments, the dimensions of the sensors were $(2.5 \times 2.5 \times 0.2) \times 10^{-3}$ m, and the volt-watt sensitivity was 5 mV/W. GHFSs signals were recorded and archived using an NI 9216 ADC by National Instruments. The GHFSs are installed uniformly in the middle longitudinal cross-section and in the character section of the transition from the input spherical segment to the cylindrical trench. The measurement uncertainty was examined and it was determined that the total standard relative measurement uncertainty of the local heat transfer coefficient is less than 10%.

For the digital twin of the experimental setup, convective heat transfer is considered during stationary turbulent air-flow around a single groove at an inclination angle $\theta$ from 0° to 90° on a heated section of the plate (shown in solid color in Fig. 1b). The heated isothermal section is mounted in a thermally insulated plate, at the entrance to which a uniform flow is set. The turbulence number is set close to the experimental one (1%), and the turbulence scale is taken to be of the order of the characteristic dimensionless size $D$. The relative radius of the edge rounding is 0.025. The Prandtl number $Pr$ is taken to be 0.71.

The computer solution of the stationary Reynolds-averaged Navier–Stokes (RANS) equations for describing turbulent flow around a plate, as well as the energy equation for predicting heat transfer characteristics, is based on the concept of splitting into physical processes [8] using the consistent pressure correction procedure SIMPLEC [15] and multi-block structured grids with their partial overlap. The semi-empirical shear stress transport (SST) model proposed by Menter [10] is utilized to solve the equations of motion. The generalized transport equation is written in increments of dependent variables. In the explicit part of the equation, the discretization of the convective terms of the equations is carried out using a hybrid scheme of variable order of approximation for the equations of momentum and the TVD scheme [9] for the equations of turbulence characteristics. The central finite-difference scheme is used to represent the diffusion terms. The implicit part of the generalized equation uses an upwind scheme with one-sided differences. The hybrid HS scheme combines a second-order approximation upstream scheme with Leonard quadratic interpolation [16] with a blending factor $R$ and a first-order approximation upwind scheme with one-way differences with a weight number (1-$R$).

By using centered computational grids, where dependent variables are determined at the cell centers, the pressure field needs to be monotonized by introducing the Rhee-Chow correction [11, 12]. The algebraic equations solution is carried out by the preconditioned BiCGSTAB method [13] with an algebraic multigrid accelerator from the Demidov library (amgcl) [2] for pressure correction and ILU0 factorization algorithm for other variables. The original VP2/3 pack (Velocity-Pressure, 2D/3D) was developed on the basis of multi-block computing technologies (MCT) described in [8]. MCT is based on a set of multi-scale, multi-tiered, and intersecting structured grids, consistent with the multi-scale structural elements of a physical problem [4]. In two rows of boundary cells of each of the intersecting or overlapping grids, the parameters are found using linear interpolation [8].

A rectangular section of a flat plate with a length of 19.625 and a width of 15.675 is being used to evaluate the flow and heat transfer. The center of the groove is located at a distance of 12.688 from the inlet section, in which a uniform flow is formed. The computational domain is built on a rectangular section of the plate and has a vertical size of 8.8. The wall flow region is displayed on the external Cartesian grid MG, condensing towards the wall (Fig. 1c; the numbers indicate: 1 – Cartesian grid for describing the flow along the plate; 2 – small-scale Cartesian grid in the near-wall layer of the heated section of the plate; 3 – curvilinear O-type grid, consistent with the groove surface; 4 – curvilinear grid “patch” in the groove central area). Inside the computational region, a rectangular zone adjacent to the wall is distinguished in the vicinity of an inclined groove with a length of 11.875, a width of 10.375 and a height of 0.7. The temperature of the wall of the selected zone is fixed at 373 K, as in the experiment. The remaining surface of the plate is assumed to be thermally insulated. In the selected rectangular area, an oblique MR mesh is constructed. In the vicinity of the groove for the basic version of mesh A, the cells of the MG and MR meshes are specified as square with a pitch of 0.07 and 0.05 in the longitudinal and transverse directions. In the direction of the inlet, outlet and lateral boundaries, the sizes of the MG grid cells increase. Curvilinear elliptical meshes Ring and Rec are introduced to represent the spatially separated flow inside the groove. The central zone is served by Rec as a patch for Ring, which is an O-type mesh. Thus, the calculation of flow and heat transfer in the computational domain is carried out on a multi-block grid of four fragmentary multi-scale grids with partial overlap. The size of the wall-mesh step is $10^{-5}$.

To examine mesh independence numerical predictions of the integral characteristics of flow and heat transfer on grids with different number of computational cells are compared. Basic grid A contains 8.8 million cells, grid B – 6.25 million cells, grid C – 4.18 million cells, grid D – 3 million cells.

The center of the Cartesian coordinate system $(x, y, z)$ is located in the middle cross-section of the channel at a distance of 6.25 from the inlet. The lateral and upper boundaries of the computational domain have sliding conditions set. In the outlet section, “soft” boundary conditions are specified – conditions for the continuation of the solution to the boundary, and no-slip conditions are set on the plates wetted surface. In the groove, a coordinated system of $s, y, t$ is put in place that is aligned in the middle longitudinal section, vertical direction, and transverse characteristic section of the transition from the input spherical segment to the cylindrical trench.

Temperature $T_{ref}$ represents the nondimensionalization scale (293 K) for the isothermal upper boundary. At the inlet the temperature T is set constant and equal to $T_{ref}$, and at the outlet from the computational domain, soft boundary conditions are set for temperature $T$. The dimensionless temperature at the isothermal section of the wall is taken to be 1.273.

The solving a problem is iterative. At each iteration step, the pressure correction equation is solved and the fields of the Cartesian components of velocity, pressure, and turbulence characteristics are calculated. The computational process ends when the maximum errors of the dependent variables reach the level of $10^{-5}$ and stabilize the extreme local and integral parameters, including the drag force and the total heat transfer coefficient in the test area with an inclined groove. In this paper, the local flow and heat transfer characteristics of a grooved plate are presented. These include longitudinal and transverse distributions of the static pressure drop $P - P_{pl}$, relative heat transfer coefficient $Nu/Nu_{pl}$ in characteristic cross-sections of a flowing around plate with a groove. The index *pl* refers to the parameters on the flat (smooth) plate.

**Table 1.** Justification of mesh independence

| $N$, mln | $Nu_{mm}$ | $10^2 \times C_x$ | $Nu_{mmd}$ | $10^2 \times C_{xd}$ |
|---|---|---|---|---|
| 8.8 | 28.32 | 0.6978 | 27.64 | 1.596 |
| 6.25 | 28.35 | 0.6980 | 27.71 | 1.597 |
| 4.18 | 28.34 | 0.6985 | 27.74 | 1.603 |
| 3 | 28.36 | 0.6961 | 27.81 | 1.606 |

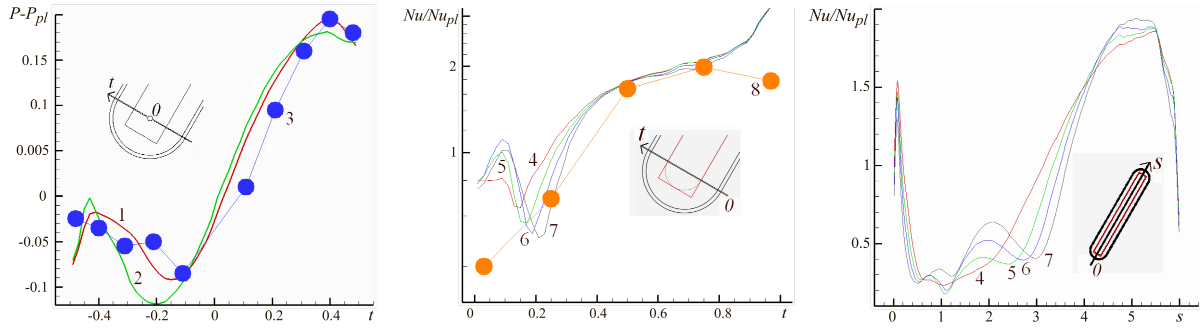
Table 1 compares numerical predictions of heat transfer coefficient $Nu_{mm}$ and drag coefficient $C_x$ for a test area with an inclined groove of size 7 × 7, and $Nu_{mmd}$ and $C_{xd}$ for an area limiting the groove contour with a size of $6 \times 1$, on multi-block meshes of the same topology and different number of the computational cells $N$ at $Re = 10^4$.

The mesh independence of the results obtained when using the convective terms of the Leonard scheme ($R = 1$) for discretization is demonstrated, and the integral indicators change little within the range of varying the number of cells from 3 to 9 million.

## 2. Results Analysis

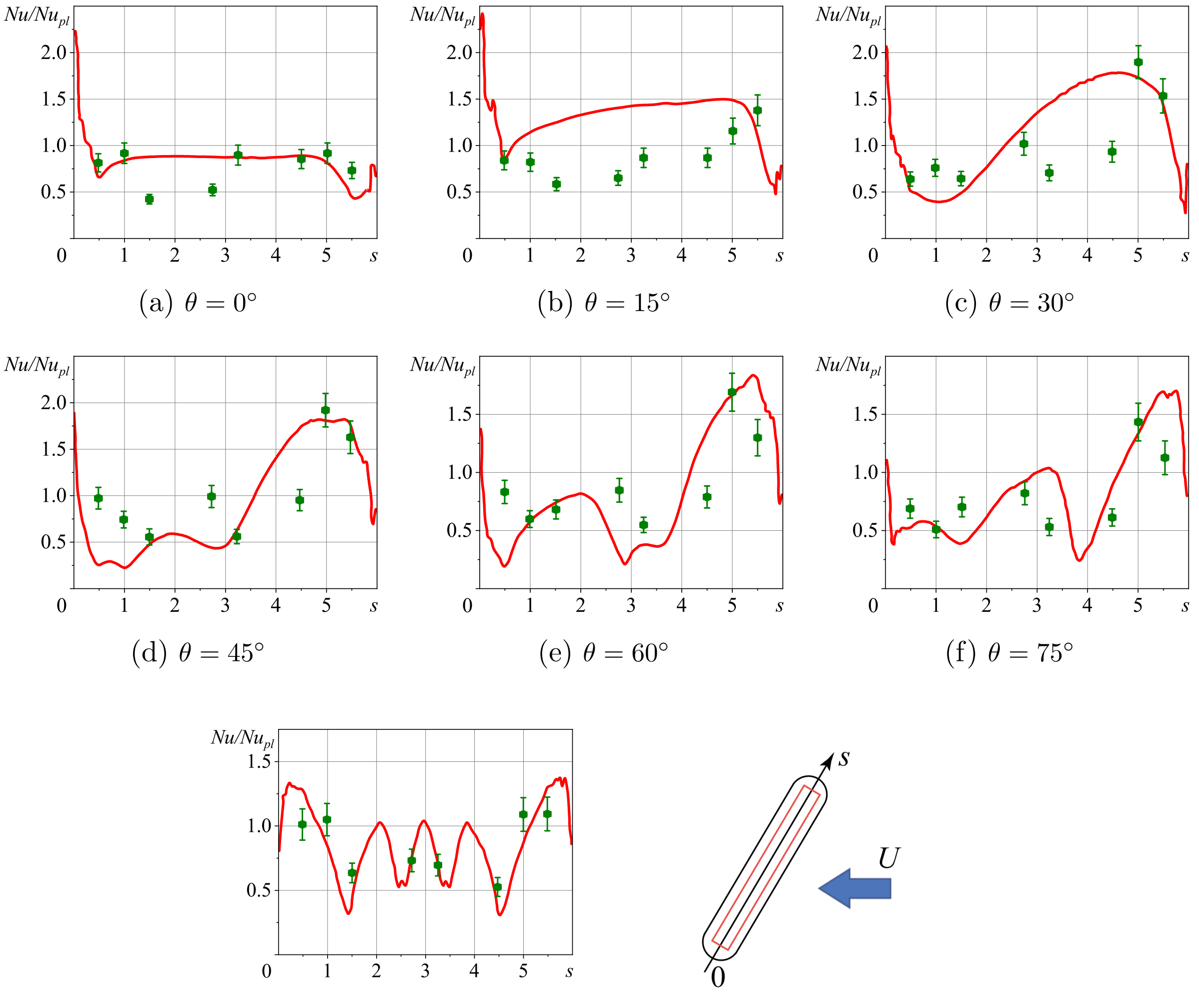
Figures 2, 3 and 4, show comparisons between numerical predictions and experimental data [7] and this study. Parametric calculations of heat transfer on a plate with a single groove are made at $Re = 3 \times 10^4$. The turbulent boundary layer on the plate before the groove was analyzed by PIV.

The extraordinary pressure drops measured on the setup of the Institute of Mechanics of Lomonosov Moscow State University [7] in the characteristic cross-section of the inlet part of a groove on the plate with a similar configuration, obtained at $Re = 6.7 \times 10^4$, are in good agreement with the RANS numerical solutions closed using the SST turbulence model (Fig. 2a; numbers indicate: 2 – forecasts based on the Fluent and 1, 4. . .7 – forecasts obtained using the VP2/3 package; 4 – $R = 0.5$; 5 – $R = 0.7$; 6 – $R = 0.8$; 7 – $R = 0.9$. 1, 3 – [7]; 2, 8 – this study).



(a) (b) (c)

**Figure 2.** Comparison of calculated (1, 2, 4–7) and experimental (3, 8) distributions of $P - P_{pl}(t)$ (a) and $Nu/Nu_{pl}(t)$ (b) in the transverse inlet section of a groove inclined at an angle of 45° passing through the center of the transition of the spherical segment and the trench, and the distributions of $Nu/Nu_{pl}(s)$ in the longitudinal middle section of the groove (c)



(a) $\theta = 0°$ (b) $\theta = 15°$ (c) $\theta = 30°$

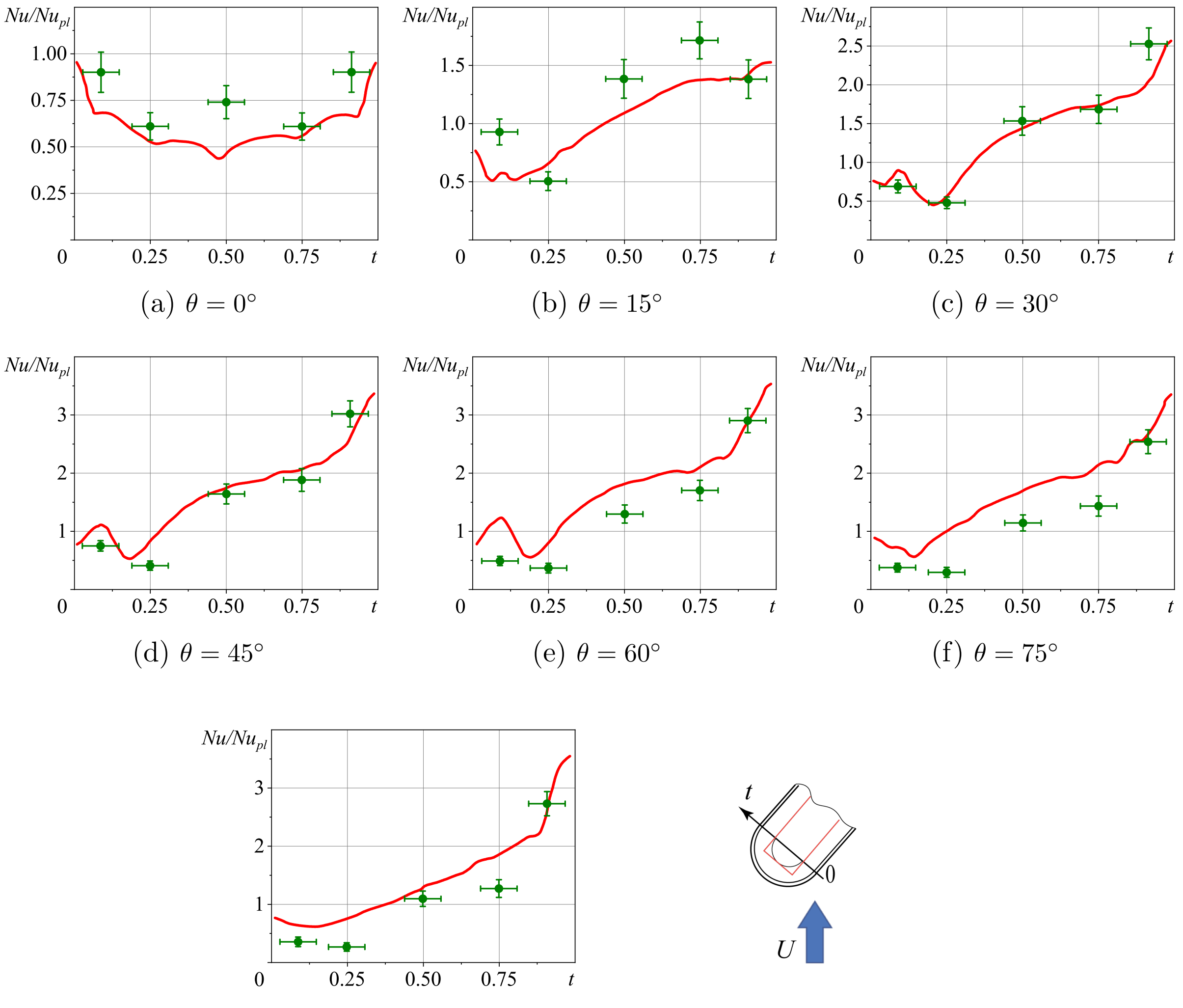
(d) $\theta = 45°$ (e) $\theta = 60°$ (f) $\theta = 75°$

(g) $\theta = 90°$

**Figure 3.** Comparison of calculated curves and experimental $Nu/Nu_{pl}$ points in the middle longitudinal cross-section of a groove on a heated plate with varying inclination angle $\theta$

On the windward slope, in the braking zone of the external flow entering the groove inclined at an angle $\theta$ of 45°, a pressure level of about 0.2 is reached, related to the double dynamic pressure ($\rho U^2$). An extraordinary pressure difference between the stagnation zone and the rarefaction region at the bottom of the groove is created by the development of a tornado-like structure generated in the spherical segment of the groove. Without the formation of a tornado and a swirling high-intensity flow, a zone of reduced pressure would not have arisen at the bottom of the groove with a minimum pressure of the order of $-0.1$. The velocities of return flows and secondary swirling flow in the inlet part of the groove are close to the free-stream flow velocity [6].

The influence of the coefficient $R$ in the hybrid scheme for approximating the convective terms of the momentum equations on the distributions $Nu/Nu_{pl}$ in the characteristic transverse and longitudinal cross-sections of the groove is considered in Fig. 2 b, c, and a comparison is made with GHFS-measurement data.



(a) $\theta = 0°$

(b) $\theta = 15°$

(c) $\theta = 30°$

(d) $\theta = 45°$

(e) $\theta = 60°$

(f) $\theta = 75°$

(g) $\theta = 90°$

**Figure 4.** Comparison of calculated curves and experimental $Nu/Nu_{pl}$ points in the in grooves cross-section passing through the center of the transition of the inlet spherical segment and the trench insert with varying inclination angle $\theta$

It is obvious that an increase in numerical diffusion with a decrease in $R$ "smooths" out the $Nu/Nu_{pl}$ distributions and has a slight effect on the maximum relative thermal loads. The $R$ coefficient in the parametric study is chosen to be 0.9. A good agreement has been achieved between the calculated $Nu/Nu_{pl}$ distributions and the experimental data, and a twofold increase in the relative heat transfer at the bottom of the groove in the zone of formation of intense return flow has been established.

Figures 3 and 4 present comparisons of the $Nu/Nu_{pl}$ databases obtained during experiments in the longitudinal and transverse characteristic sections of the groove when varying the inclination angle $\theta$ from 0° to 90° and numerical predictions based on RANS solutions using parallelized VP2/3 codes.

Overall, there is a satisfactory agreement between numerical simulation and experiment taking into account the turbulence modeling approach. However, the results obtained for $\theta = 15°$ require further study. It should be noted that an anomalous heat transfer enhancement in the grooves occurs at angles $\theta$ from 30° to 75°, which correlates with the conclusions from the analysis of pressure distributions in the grooves on the plate [6].

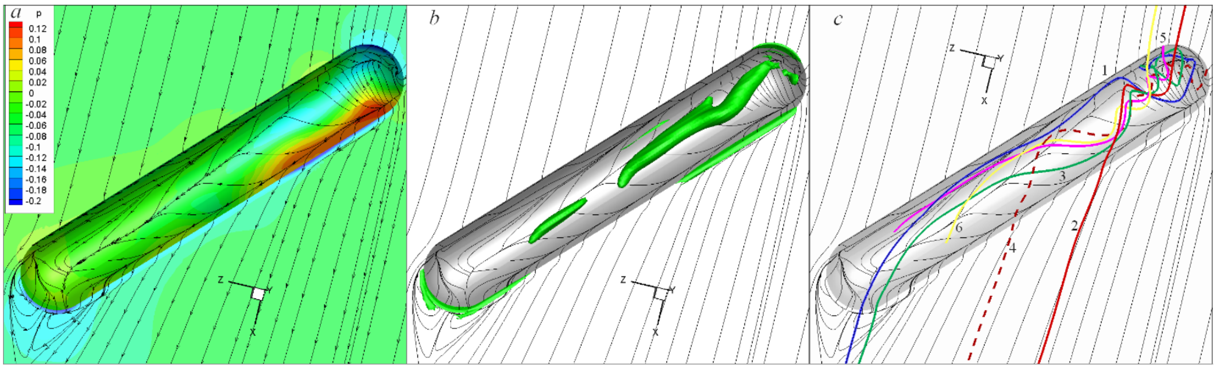


**Figure 5.** Pressure field (a), isosurface Q2 = $-1.12$ (b) and trajectories of liquid particles, passing through a point with coordinates $x = 4.4$; $z = -1.5$ (c), with printed patterns air spreading over the surface of the groove on the plate at an inclination angle of 45°. $1 - y = -0.24$; $2 - y = -0.2$; $3 - y = -0.15$; $4 - y = -0.1$; $5 - y = -0.05$; $6 - y = 0$

Figure 5 illustrates the vortex structure in a groove on the plate at an angle $\theta = 45°$. Air spreading over the wall is shown superimposed on the pictures distribution of surface pressure related to twice the velocity pressure, isosurface of criterion Q2 equal to $-1.12$, and trajectories of liquid particles passing through a fixed point in the center of the joining section of the input spherical segment and a cylindrical trench. Focused pressure difference between adjacent zones braking on a windward slope and an area of reduced pressure at the entrance to the groove is initiated by a tornado-like vortex and a developing intense swirling flow, shown by isosurfaces of Q2. The flow in the groove carries complex nature, demonstrated in Fig. 5.

## Conclusion

To test software for parallel computing systems, a validation test is presented, based on a study of the phenomenon of separated turbulent air flow and heat transfer anomalous enhancement in an inclined groove on a heated plate. Its peculiarity is the generation of tornadoes in grooves, associated with extraordinary differences in static pressure, the formation of fields of ultra-high velocities, high gradients of relative friction and heat transfer coefficient inside the inclined groove. Databases of heat flux measurements in a groove on the isothermal section of the plate were created when the inclination angle varied from 0° to 90°, obtained at the SPbPU thermophysical stetup using gradient heatmetry. There is quite a satisfactory agreement between experimental data and numerical predictions made within the framework of RANS-SST using the parallelized package VP2/3.

## Acknowledgements

This research was sponsored by the Russian Science Foundation (grant number 22-19-00056 (experiment) and grant number 23-19-00083 (calculations)).

## References

1. Chen, C.J., Jaw, S.Y.: Fundamentals of turbulence modeling (1998), `https://api.semanticscholar.org/CorpusID:118248285`

2. Demidov, D.: AMGCL – A C++ library for efficient solution of large sparse linear systems. Software Impacts 6, 100037 (2020). `https://doi.org/10.1016/j.simpa.2020.100037`

3. Deryugin, Y., Zhuchkov, R., Zelenskiy, D., *et al.*: Validation results for the LOGOS multifunction software package in solving problems of aerodynamics and gas dynamics for the lift-off and injection of launch vehicles. Mathematical Models and Computer Simulations 7, 144–153 (03 2015). `https://doi.org/10.1134/S2070048215020052`

4. Isaev, S., Chulyunin, A., Nikushchenko, D., *et al.*: Analysis of the anomalous intensification of a separate flow and heat transfer in a stabilized section of a narrow channel with single-row, inclined, oval-trench dimples with the use of various grids and turbulence models. High Temperature 59, 106–114 (01 2021). `https://doi.org/10.1134/S0018151X21010041`

5. Isaev, S.A.: Genesis of Anomalous Intensification of Separation Flow and Heat Transfer in Inclined Grooves on Structured Surfaces. Fluid Dynamics 57(5), 558–570 (Oct 2022). `https://doi.org/10.1134/S0015462822050081`

6. Isaev, S.A., Guvernyuk, S.V., Nikushchenko, D.V., *et al.*: Correlation between the abnormal enhancement of the separated flow and extraordinary pressure drops in the groove on the plate when the angle of inclination changes from 0 to 90°. Technical Physics Letters 49(8), 33–36 (2023). `https://doi.org/10.61011/TPL.2023.08.56684.19560`

7. Isaev, S.A., Sudakov, A.G., Nikushchenko, D.V., *et al.*: Tests for Validation Problems of Anomalous Intensification of Separation Flow and Heat Transfer on Structured Surfaces with Extraordinary Pressure Differences. Fluid Dynamics 58(5), 894–905 (Oct 2023). `https://doi.org/10.1134/S001546282360133X`

8. Isaev, S., Baranov, P., Popov, I., *et al.*: Numerical simulation and experiments on turbulent air flow around the semi-circular profile at zero angle of attack and moderate Reynolds

number. Computers & Fluids 188, 1–17 (2019). https://doi.org/10.1016/j.compfluid.2019.03.013

9. Leonard, B.P.: A stable and accurate convective modelling procedure based on quadratic upstream interpolation. Computer Methods in Applied Mechanics and Engineering 19, 59–98 (Jun 1979). https://doi.org/10.1016/0045-7825(79)90034-3

10. Menter, F.: Zonal Two Equation k-$\omega$ Turbulence Models for Aerodynamic Flows. https://doi.org/10.2514/6.1993-2906

11. Pascau, A., Garcia-Polanco, N.: Consistency of SIMPLEC scheme in collocated grids (06 2010)

12. Rhie, C.M., Chow, W.L.: Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA Journal 21(11), 1525–1532 (Nov 1983). https://doi.org/10.2514/3.8284

13. Saad, Y.: Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, second edn. (2003). https://doi.org/10.1137/1.9780898718003

14. Seroshtanov, V., Gusakov, A.: Gradient Heatmetry and PIV Investigation of Heat Transfer and Flow near Circular Cylinders. Inventions 7(3) (2022). https://doi.org/10.3390/inventions7030080

15. van Doormaal, J.P., Raithby, G.D.: Enhancements of the simple method for predicting incompressible fluid flows. Numerical Heat Transfer 7, 147–163 (Jun 1984)

16. van Leer, B.: Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. Journal of Computational Physics 32(1), 101–136 (1979). https://doi.org/10.1016/0021-9991(79)90145-1

DOI: 10.14529/jsfi240202

# Assessment of Several Advanced Numerical Algorithms Implemented in the CFD Code SINF/Flag-S for Supercomputer Simulations

*Evgueni M. Smirnov*[1], *Dmitri K. Zaitsev*[1], *Alexander A. Smirnovsky*[1], *Elizaveta V. Kolesnik*[1], *Aleksei A. Pozhilov*[1]

Computational Fluid Dynamics demands substantial computational resources and advanced numerical algorithms for accurate simulation of fundamental and industrial problems. This paper presents an experience in assessing several numerical algorithms implemented recently into the in-house finite-volume code SINF/Flag-S developed at the Peter the Great St. Petersburg Polytechnic University for supercomputer simulation. Three topics are covered: (i) implementation and testing of an original geometric multigrid method for solving linear algebraic equations; (ii) application of a fractional step method for solving unsteady incompressible fluid motion equations; and (iii) description and testing of a density-based solver for compressible gas viscous flow simulation across a wide Mach number range. For each of the topics considered, the results of the calculations of some testing problems are presented, namely: a model problem of heat transfer in a cubic domain, turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container, free convective flow around a subsea cooler model, high-speed gas flow with strong effects of viscous-inviscid interaction. The parallel efficiency of the implemented algorithms is demonstrated, and their significance for large-scale simulations on supercomputers is highlighted.

*Keywords: CFD, multigrid method, incompressible fluid, fractional step method, compressible gas, density-based solver.*

## Introduction

Computational Fluid Dynamics (CFD) is one of the most resource-consuming domains in the scientific field, which requires a large amount of computational power to perform accurate numerical simulations for various physical problems and tasks. In particular, huge computing power is necessary to perform high-precision numerical simulation of turbulent flows and complex 3D high-speed flows of viscous gas. Therefore, it is of great interest to both improve the technical characteristics of supercomputers and introduce increasingly efficient and reliable numerical methods into codes.

In this paper, we represent our recent experience in assessing some advanced numerical techniques implemented in the in-house finite-volume code SINF/Flag-S, developed at the SPbPU (Peter the Great St. Petersburg Polytechnic University). A brief description of the code is given in Section 1. In the following sections, three topics are considered: (i) implementation and testing of the original geometric multigrid method for solving systems of linear algebraic equations, (ii) approbation of the original fractional step method for solving unsteady equations of the incompressible fluid motion and (iii) description and testing of an effective density-based solver for the compressible gas viscous flows in a wide range of Mach number (sub-, trans- and supersonic). Some data are also provided to illustrate the parallel efficiency of the implemented algorithms.

Multigrid methods are very effective for numerically solving large-size systems of equations. Although the basic idea was proposed back in 1961 [12], the development of effective multigrid methods is still the subject of up-to-date scientific studies [13, 14, 29]. In particular, multigrid methods demonstrate high efficiency at numerically solving systems of linear algebraic equations

[1] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation

obtained by discretizing a Poisson-like equation. Typical problems, in which the Poisson-like equation arises, are as follows: steady heat transfer in solids, a pressure equation in SIMPLE-like algorithms for pressure-velocity coupling [36] and in fractional-step methods. It is known that the time required to solve the pressure equation in a SIMPLE-like algorithm can reach 90% of the total calculation. It is also known that the complexity of the multigrid algorithm increases as $O(N)$, where $N$ is the number of grid cells, while the complexity of the Krylov sub-space methods (like Conjugated Gradients method, see [41]) increases at best as $O(N^{3/2})$. Thus, the necessity of implementing multigrid method in the numerical codes is essential when dealing with large grids and calculations on supercomputers. In Section 2 of the paper, we present our experience in developing and approbating an original parallelized Geometric Multigrid Method for solving systems of linear algebraic equations that arise when solving elliptical-type governing equations using the Finite Volume Method.

For faster simulation of unsteady incompressible fluid flows, especially in case of turbulent motion, effective and robust numerical methods are needed. Traditional SIMPLE-like algorithms require a number of iterations at each current time layer while advancing in physical time. In the so called non-iterative methods, intermediate iterations between time-layers are almost absent (except the iterations due to the grid "non-orthogonality" issues). In the class of the non-iterative methods, fractional-step methods are very widespread [3, 10, 21]. Usually, these methods are partially or fully implicit; in the latter case, relatively large time steps can be achieved. Section 3 covers a brief description of the implicit fractional-step method implemented in the SINF/Flag-S code for solving unsteady equations of incompressible fluid motion and two examples of its application for performing Direct Numerical Simulation of free convective flows.

In industry and aero-engineering, problems often arise, where compressible flows are present in a wide range of velocities. Up-to-date numerical methods for calculating compressible flows provide the ability to accurately resolve different gas-dynamic discontinuities. There are many different schemes for calculating convective fluxes in case of trans- and supersonic motion, which are currently being actively developed [31, 45]. On the other hand, there are many applications where the role of viscous motion in subsonic regions are also very significant, for example, the flow of gas or vapor in a turbine vane cascade [30] or the flow around a helicopter blade [1]. For these applications, simulation should be performed with a reliable and robust numerical method that allows a smooth transition from subsonic to supersonic flow, taking into account viscous effects. Section 4 briefly describes our experience in implementing robust numerical techniques for calculating viscous compressible gas flows in a wide range of the Mach number followed by an example of using these techniques for accurate test computations of a high-speed 3D problem with complex effects of viscous-inviscid interaction.

## 1. General Description of the SINF/Flag-S Code

The finite-volume "unstructured" code SINF/Flag-S operates on unstructured meshes with cells of arbitrary topology (tetrahedrons, hexahedrons, prisms, polyhedra). The code is written primarily in the Fortran 90 programming language, with some functions written in C programming language. The SINF/Flag-S program is a console application; all information is read from control text files. The ANSYS Fluent format is used for mesh files and results files; to allow restarting from saved fields, a special file format is used.

The SINF/Flag-S code is intended for simulation the following classes of problems:

- steady and unsteady flows of viscous fluid;

- compressible viscous gas flows in a wide range of the Mach number;
- conjugate or non-conjugate heat transfer in liquids, gases and solids;
- simulation of turbulent flows using different approaches, such as: (U)RANS, LES, IDDES, DNS;
- heat buoyancy in the mass force fields of different kinds: gravitational, centrifugal, Coriolis;
- flows in the porous medium with accounting for phase interchange (evaporation and condensation);
- multiphase flow simulation using the Volume of Fluid approach;
- two phase filtration flow simulation using the "black oil" model;
- calculation of flows in domains with moving boundaries.

In the case of incompressible or low-Mach flows, different methods of solving governing equations can be used, in particular: the SIMPLEC method [48], the Rogers–Kwak scheme [40], the coupled method based on the Rhie–Chow interpolation [9]. To calculate unsteady flows, iterative algorithms based on the above-mentioned methods or the "nominally" non-iteration fractional step method (see Section 3.1) can be used. For simulation of the compressible viscous gas flows in a wide range of the Mach number (from sub- to supersonic), the coupled density-based solver is used (see Section 4.1).

Discretization of spatial operators is performed using the Finite Volume Method on collocated grids. For the approximation of viscous (diffusion) terms, the hybrid central scheme of 2nd order with non-orthogonality correction is used. Different interpolation schemes can be used for the approximation of the convective terms in governing equations. For the incompressible or low-Mach flows there are following interpolation schemes: central (linear, 2nd or 4th order using gradients), second order upwind, QUICK, weigthed schemes, bounded schemes with Jameson or scalar limiters. In the density-based solver, different schemes for inviscid fluxes calculation have been implemented, which are briefly described in Section 4.

For solving systems of linear algebraic equations, in the SINF/Flag-S code there are different linear solvers, in particular: CG, BiCG, GMRES, Geometrical Multigrid (see description in Section 2.1), which can be preconditioned, in particular, by Symmetric Gauss–Seidel preconditioner, Incomplete Cholesky or LU factorization.

To ensure the ability to solve the problem in parallel mode, the domain decomposition approach based on task parallelism using the MPI (Message Passing Interface) library of the MPI 2.0 standard is used [15]. The division of the problem into separate subtasks is carried out based on blocks of a multi-block computational mesh, for which the code contains an auxiliary utility Par4Flag; the latter uses the ParMetis library [27] to decompose the computational mesh. To exchange information between processes, an additional row of "halo" cells is used, geometrically duplicating the border cells of the adjacent block.

## 2. Geometric Multigrid Method and Evaluation of Its Performance

### 2.1. Description of Implemented Geometric Multigrid Method

As it was shown by the practice of hydrodynamic calculations, the most resource-intensive part of the SIMPLE-like algorithms is solving the Poisson equation for pressure correction. The Poisson equation also describes thermal conductivity in solid structural elements and the filtration movement of the working fluid in a porous wick. Thus, using a specialized algorithm in a linear solver that speeds up obtaining a Poisson equation solution can significantly reduce the time spent on solving the problem and, accordingly, significantly increase the efficiency of using available computing resources. When solving linear systems of large size, multigrid methods demonstrate the greatest efficiency, since their use allows achieving a linear increase in the time required to solve a problem as its size increases. The main difficulty in applying multigrid methods on unstructured meshes is in constructing grids for coarse levels. One of the most promising is the agglomeration method [34], in which the cells of coarse computational grids are obtained by agglomeration (combining) the cells of the original grid. The question of the most effective method of cell agglomeration still remains open and is the subject of a number of modern studies, for example [32, 35].

An original agglomeration multigrid method was implemented in the SINF/Flag-S code. The agglomeration method is based on the algorithm proposed in [37]. At the same time, it includes several original modifications, which significantly improve the quality of the resulting agglomerates and, accordingly, increase the stability and speed of convergence of the multigrid method. The modified agglomeration algorithm consists of the following steps, repeated cyclically after the initial creation of a queue of border cells:

1. Construction of the basis. A cell from the queue is merged with two (2D case) or three (3D case) neighboring cells that have a common node, and the distance to which does not exceed the distance to the nearest cell by more than 4 times. If there are several options for merging, then the one that contains the largest number of cells already in the queue is selected (if there are still several options, then the first one is taken). If a suitable set of cells is not found and the distance to at least one of the neighboring cells exceeds the distance to the nearest cell by more than 4 times, then the first step of the algorithm is repeated, with an attempt to construct a basis containing one less cell. If it is possible to form a basis in the end, then proceed to step 2; otherwise – to consider the next cell from the queue.
2. Addition of cells. The cells adjacent to the resulting agglomerate are examined. If among them there is a cell adjacent to at least two cells from the agglomerate, and the addition of this cell reduces the value of the parameter $Ar$ (aspect ratio, defined as the ratio of linear dimensions characterizing the surface area and volume of the agglomerate), then this cell is added to the agglomerate; otherwise, the transition to the next step is carried out. If there are several suitable cells, then the one whose addition leads to a smaller value of the $Ar$ parameter is selected. If the number of cells included in the agglomerate is less than 4 (2D case) or 8 (3D case), then the step is repeated.
3. Queue replenishment. Cells that are adjacent to the resulting agglomerate and have not yet been combined with other cells are added to the queue (if they have not already been added to it). Then the next cell from the queue is processed (step 1).
4. Holes patching. All unmerged cells are marked. The selected unmerged cell is merged with one of the neighboring agglomerates. It is forbidden to add an unmerged cell to the agglomerate if all the neighbors of this cell included in the agglomerate were marked (i.e., they were also unmerged cells). Among the agglomerates allowed for combining, the one for which the addition of this cell will most strongly decrease (or least increase) the value of the $Ar$ parameter is selected. Then proceed to the next unmerged cell.

The number of grid levels is selected automatically, and the construction of new grids continues until one cell remains in each calculation block. The difference operator on coarse grids is formed by summing the equations with scaling coefficients [26], however, instead of the global scaling proposed in [26] (by the number of grid cells), local scaling is used (based on the ratio of the distances between the centers of the cells of the fine and coarse grids), which increases efficiency of the method in the case of grids with highly elongated cells. The Symmetric Gauss–Seidel method (SGS) is used as a smoother. Various types of multigrid cycles (V-, W-, F-cycle) with a fixed number of iterations have been implemented. The developed multigrid method can be used both as an independent solver and as a preconditioner for the GMRES method.

## 2.2. Evaluation of Method Performance and Parallelization Efficiency

The effectiveness of the developed multigrid method was evaluated on a model problem of heat transfer in a cubic domain with constant thermal conductivity (see Fig. 1). Almost all the boundaries of the region were considered adiabatic, except two small regions adjacent to the opposite vertices, in one of which the temperature value was specified, and in the second the heat flux was specified. The dimensionless formulation of the problem is illustrated in Fig. 1.

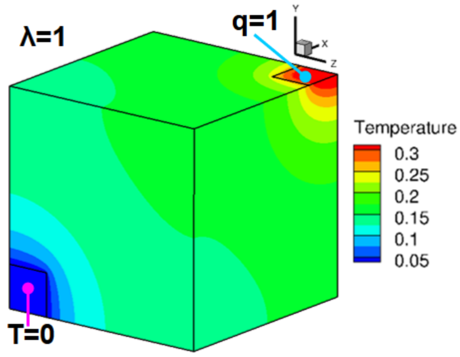


**Figure 1.** Model problem of conductivity heat transfer in a cube

Comparative calculations were performed using the SINF/Flag-S code and the ANSYS Fluent 23R1 code on uniform (isotropic) hexahedral and tetrahedral computational grids of different spatial resolutions, as well as on polyhedral grids obtained from the tetrahedral one using the ANSYS Fluent package. When using the SINF/Flag-S code, the two different methods were used as linear solvers: the developed multigrid method (with W-cycle) and the GMRES method (with a SGS preconditioner). Of the options available in the Fluent code, the BCGSTAB linear solver method with the multigrid preconditioner using a W-cycle has been selected as the most effective one for solving the formulated task (based on the results of preliminary tests). Calculations were performed in single-threaded mode. The time required to solve the problem with a prescribed accuracy was being determined (the problem was considered solved when the value of the relative integral imbalance of the heat flux became less than $10^{-4}$). Figure 2 shows the dependencies of the problem computational time on the number of computational grid cells obtained with various codes and numerical methods.

Here, the data obtained in calculations with polyhedral grids are presented; calculations with hexahedral and tetrahedral grids gave similar results. It can be seen that when using the GMRES method (code SINF/Flag-S), the time to solve the problem increases proportionally to the number of cells to the power of 1.5. The use of both multigrid algorithms provides, as expected, a linear increase in computational time as the number of grid cells increases. The multigrid method developed and implemented in the SINF/FlagS code made it possible to solve this problem about 2 times faster than using the Fluent code.

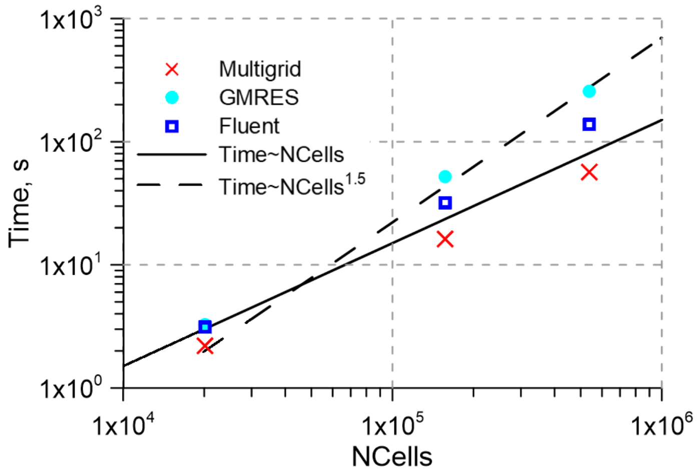
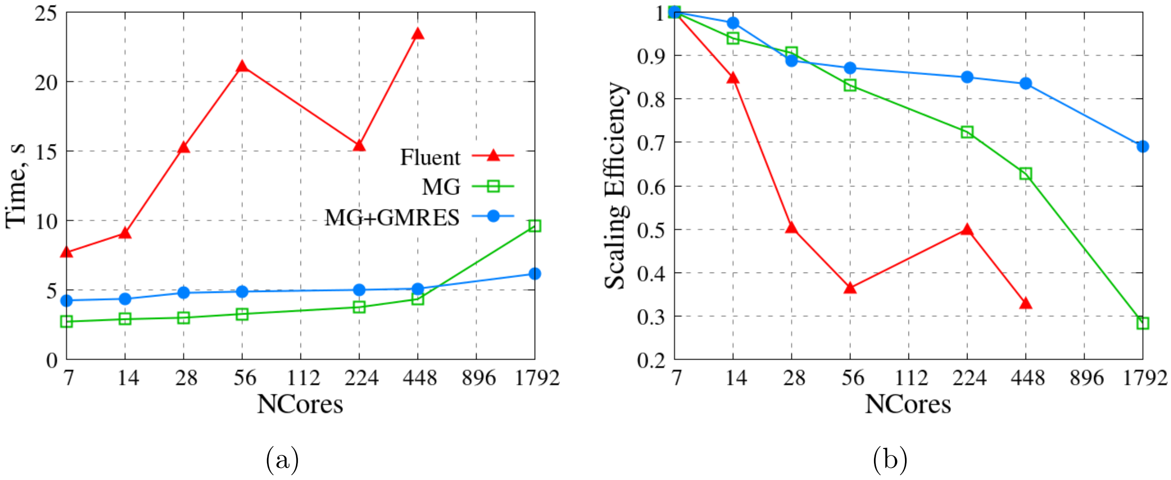


**Figure 2.** Efficiency of different solvers with increasing the grid size

The scalability of the developed multigrid method was evaluated as well. The considered model problem was solved using 7 to 1792 cores while maintaining the number of cells per computing core; accordingly, the grid size varied from four million to a one billion cells. The calculations were carried out on the SPbPU Polytechnic RSC Tornado supercomputer. Figure 3 shows the dependencies of the problem-solving time and the scaling efficiency on the number of cores involved when using the SINF/Flag-S and ANSYS Fluent 23R1 codes. In calculations with the SINF/Flag-S code, the developed multigrid method was used both independently (MG) and as a preconditioner for the GMRES method (in the latter case, a more economical V-cycle was used instead of the W-cycle). When using the Fluent code, as before, the BCGSTAB method with a multigrid preconditioner turned out to be the most effective. From Fig. 3 one can see that when using a relatively small number of cores (up to 500), the MG+GMRES combination in the SINF/Flag-S code works up to one and a half times slower than the pure multigrid method, but the scalability of such combination is better. One can conclude also that for the considered model problem, the Fluent code is significantly inferior to the SINF/Flag-S code both in terms of problem solving time and scalability.



(a) (b)

**Figure 3.** “Weak” parallel scalability: computational time (a) and scaling efficiency (b)

# 3. Fractional Step Method and Its Testing

## 3.1. Description of Method

As it was mentioned in Introduction, in some cases, when solving unsteady equations of incompressible fluid dynamics, the use of iterative time advancement methods (the SIMPLE method and similar) can be very resource-consuming. In particular, simulation of turbulent flows using eddy-resolving approaches such as LES, DES, etc., requires small time steps to adequately resolve the dynamics of turbulent eddy structures. To speed up the solution, it is reasonable to use faster numerical approaches based on a non-iterative transition from one time layer to another. One such approach is the so-called projection method or fractional step method [8, 10, 21, 33].

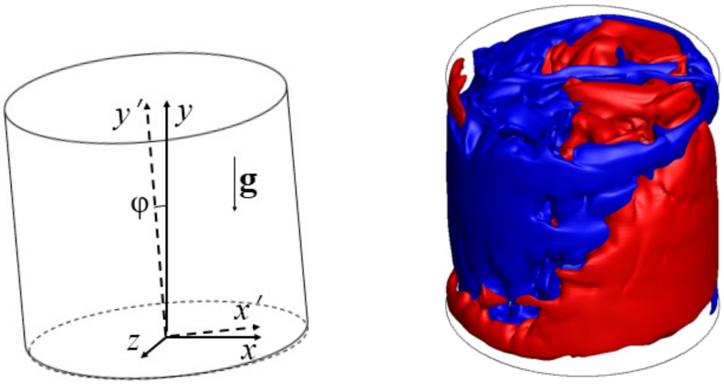
The fundamental principle of the fractional step method (FSM) lies in the separation of the spatial operators in the momentum equation. The pressure gradient is interpreted as a projection operator that transforms an arbitrary velocity field into a solenoidal one. The momentum balance equation is thus split into two equations. The first one is solved either without the pressure gradient at all or with the pressure gradient taken from the previous time layer and serves to determine the predictor value of the velocity. This predictor velocity, of course, does not necessarily satisfy the continuity equation. The second equation, which is written for the corrector velocity vector, relates the pressure gradient at the new time layer to the velocity vector at the new time layer. The combination of the second equation and the continuity equation leads to the Poisson equation for the pressure at the new time layer. It is important to emphasize that, unlike SIMPLE-like algorithms, where a Poisson-like equation also arises (it is written for pressure correction), the resulting equation for pressure at the new time layer is physically justified and should be solved with good accuracy to ensure the mass balance equation at the new time layer is satisfied. So, the use of advanced numerical techniques like multigrid methods is rather essential for the FSM.

Another important aspect that affects the effectiveness of the FSM is the evaluation of the convective terms in the predictor velocity equation. The convective term can be calculated by different methods. The most common approach includes the extrapolation of the convective term from the two preceding time levels using the Adams–Bashforth scheme. For the better stability of the numerical algorithm, implicit schemes for calculating the convective term are preferable; they can be introduced in different ways, for example, as in [6, 19]. The SINF/Flag-S code uses an original scheme for computing the convective terms, which combines extrapolation from two preceding time levels with introducing implicitness. Namely, the mass flux through the face of the finite volume is extrapolated from previous time layers using the Adams–Bashforth scheme (at the intermediate time layer in the case of the Crank-Nicolson scheme or at a new time layer in the case of the backward 2nd order scheme for time derivative approximation), while the transported quantity (velocity vector) is written implicitly at the desired time layer (intermediate or new). Our experience in using the proposed algorithm in the SINF/Flag-S code shows that the introduced implicitness can essentially increase the FSM stability in the case of highly skewed and non-orthogonal cells. For more details concerning the realized FSM see [44].

Below, the performance of the fractional step method implemented in the SINF/Flag-S code is illustrated by two examples: turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container and free convective transitional flow near a subsea cooler model.

### 3.2. Turbulent Rayleigh–Bénard Convection in a Slightly Tilted Cylindrical Container

Rayleigh–Bénard convection of fluid in a vertical cylindrical container is one of the canonical model problems in the investigations of natural convection [2]. Experimental and numerical studies show that large-scale circulation (e.g. in the form of large-scale vortex covering the entire region of convective flow) is a characteristic feature of such flow. Using the SINF/Flag-S code, Direct Numerical Simulation of water turbulent convection in a slightly tilted cylindrical container heated from below was performed recently [43] at the Rayleigh number $Ra = 10^8$ and the Prandtl number $Pr = 6.4$ (the container height was equal to its diameter). As shown in [42], tilting of a cylinder leads to "locking" of the large-scale vortex in a certain azimuthal position. In Fig. 4, the computational domain as well as large-scale vortex vizualization are presented. The simulation was carried out based on the Navier–Stokes equations written with the Boussinesq approximation for including the buoyancy effects.

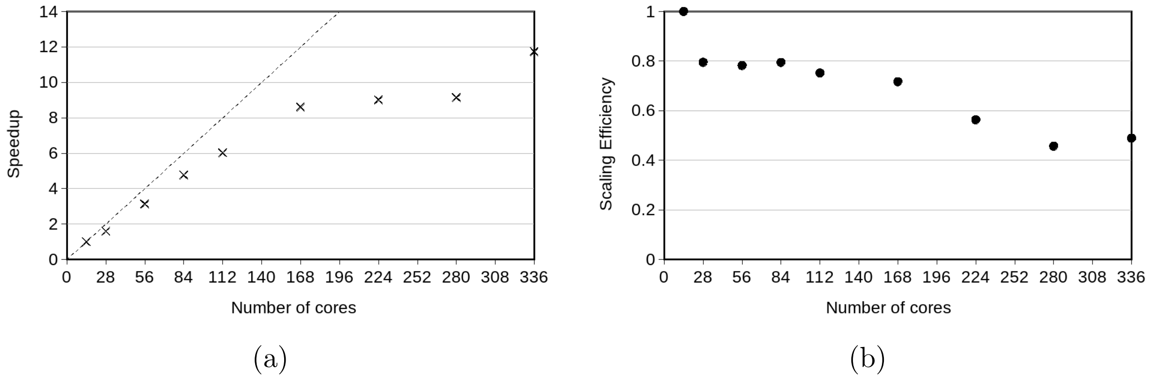


(a) (b)

**Figure 4.** Geometry of computational domain for tilted container (a) and visualization of large-scale vortex by isosurfaces of positive (red) and negative (blue) instantaneous vertical velocity component (b)

The fractional step method with the Crank–Nicolson scheme was used for advancing in time. A central difference scheme was used to approximate the convection terms in the governing equations. The computational grid consisted of approximately $1.5 \times 10^7$ cells. The time step was chosen so that the local values of the Courant number were less than unity; characteristic time of the buoyancy process (buoyancy time) was estimated as about 25000 time steps. The computed fields were averaged over time starting after a transient process; the sample for averaging was about 150 buoyancy times.

The calculations were performed using resources of the Polytechnic Supercomputer Center, real time of the calculation of 150 buoyancy time (about $4 \times 10^6$ time steps) using 12 nodes (28 cores on each node) of the supercomputer took about two months. More details about the study and results can be found elsewhere [43]. Testing of the parallelization efficiency was performed as well. The results on the calculated speedup and scaling efficiency are presented in Fig. 5. It can be seen that a strong degradation in scalability is observed in the interval from 168 to 224 cores. Nevertheless, the scaling efficiency in case of a large number of cores remains about 0.5 that is a relatively good level.



(a) (b)

**Figure 5.** Parallel scalability: speed-up (a) and scaling efficiency (b)

### 3.3. Free Convective Flow Around a Subsea Cooler Model

The problem of free convective flow around a subsea cooler arises from the practical need to have reliable data on the heat transfer efficiency of a heat exchanger located on the seabed and intended to cool natural gas produced from the shelf [16]. Such passive heat exchangers consist of a bank of pipes that use the surrounding cold water as a coolant (see Fig. 6). The heating of the surrounding water induces a buoyancy-driven draft flow, resulting in mixed convection conditions for the inner-rows tubes; this convection is characterized by relatively low Reynolds numbers, regarding the flow around the pipes. When the heat exchanger design relies primarily on the data for natural convection, the heat transfer coefficient at the external surface of the pipe becomes a critical uncertainty. Up-to-date CFD methods can provide valuable data on external heat transfer for the low-Reynolds-number regimes commonly encountered in buoyancy-dominated coolers. More details about the conditions typical for subsea cooler simulation can be found elsewhere [17, 18].

Here we present some results of the 3D simulation of water flow near a subsea cooler consisting of ten and twenty-two pipes with inlet and outlet manifolds (Fig. 6). No-slip conditions and a constant temperature were set on the surface of heated pipes and collectors. It was assumed that the exchanger was located in a boundless water basin, the outer boundaries of the computational region were moved away from the cooler to a large distance, and conditions for the free flow of water were imposed on them.

The simulation was carried out based on the direct solution of the unsteady Navier–Stokes equations; buoyancy effects were taken into account in the Boussinesq approximation; typical values of the Grashof number evaluated for a pipe was about $10^5$. The calculations were performed using the SINF/Flag-S code with the fractional step method. Resources of the Polytechnic Supercomputer Center were used. For the bank of ten pipes, the grid consisted of more than 75 million cells, and to obtain a representative sample, the calculations using 448 cores were being carried out in two weeks. For the bundle of 22 pipes, the grid size was about 150 mln cells. Using 896 cores, a representative sample was also obtained in two weeks.

Visualization of the flow is shown in Fig. 7, where streamlines colored by temperature as well as temperature distributions in cross sections are presented. It can be seen that in the center of the tube bank, the flow is nearly stationary. Contrary to that, the flow in the upper pipes region is highly unsteady, with almost turbulent behavior.

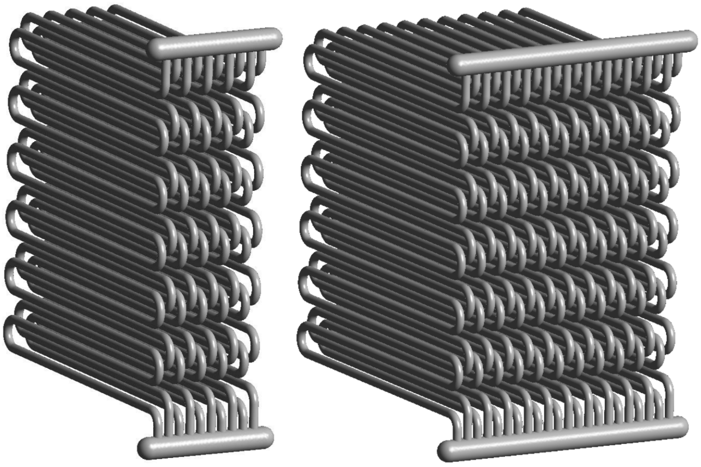
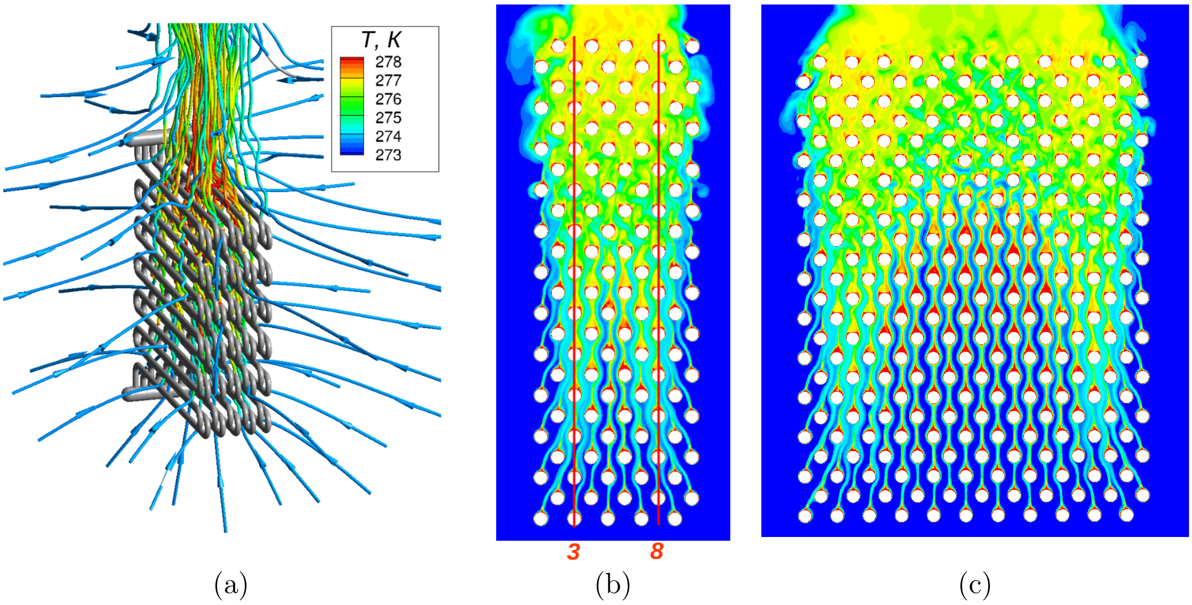


**Figure 6.** The computational geometry of subsea cooler consisting of 10 (left) and 22 (right) pipes



(a) (b) (c)

**Figure 7.** Flow visualization: a – streamlines colored by temperature for the bank of 10 pipes; b and c – temperature in cross section for two geometries (red lines show the position of 3rd and 8th row)

# 4. Coupled Density-based Solver for Viscous Compressible Gas Flows in a Wide Range of Mach Number

## 4.1. General Remarks for Developed Density-based Solver

Compressible gas flows are characterized by the presence of discontinuities of various kinds, which especially manifest themselves at high flow velocities (when Mach number greater than unity) in the form of shock waves. In numerical simulation of such flows, the method of approximating convective flux is especially important, and the scheme used must in one way or another take into account the characteristic properties of the systems of equations. In the SINF/Flag-S code, the following schemes are available for convective flux evaluation: Godunov, Roe, HLL, HLLC, and schemes of the AUSM family.

High order of accuracy is achieved using the MUSCL slope-limiting approach, according to which piecewise-linear reconstruction of the solution is built for each control volume. In the SINF/Flag-s code, there are two approaches for constructing a high-order accuracy scheme for the case of unstructured grids: a scalar approach with second-order limiters [5], and a quasi-one-dimensional approach, consisting in applying quasi-one-dimensional computations for locally selected directions (i.e., for each face). The proposed in [4, 28] version of the quasi-one-dimensional approach, where additional virtual points are reconstructed, is implemented in the SINF/Flag-S code. To preserve monotonicity of the solution near discontinuities, second-order TVD schemes are used. Thanks to the reconstruction, refined values of the variables to the left and right of the control volume face are calculated, which are then used to calculate the inviscid fluxes when selecting one of the above-mentioned schemes.

In the case of very high Mach numbers, the so-called “carbuncle” instability may appear in the numerical solution [38], which leads to a strong distortion of the shock wave front in the numerical solution. In the SINF/Flag-S code, additional artificial viscosity, introduced according to [39], is used to suppress this instability.

On the other hand, the main difficulty in modeling low-speed compressible flows lies in the “acoustic stiffness” (a strong difference in the magnitude of the eigenvalues) of the system of equations for the dynamics of a compressible gas, which arises at substantially subsonic flow velocities [7, 11, 20, 46]. This is due to the very different characteristic times of convective transport and propagation of acoustic disturbances and leads to the fact that in the system of equations the matrix for time derivatives becomes degenerate as the Mach number tends to zero. Under these conditions, the stability and speed of convergence of traditional methods of numerical integration of the full Navier–Stokes equations, many of which were originally developed to calculate supersonic flows, sharply deteriorate, and in the limiting case of the Mach number approaching zero, these methods generally lose their functionality. To overcome this difficulty, regularization based on the Turkel preconditioning is implemented into the SINF/Flag-S code, according to [49].

The implemented in the code SINF/Flag-S density-based solver was successfully approved on the series of different problems, including the problem of the sub- or supersonic flow past the blunt-fin body mounted on a plate. The detailed description of the problem and numerical results both for incompressible fluid flows with different Prandtl numbers and the compressible gas flow with the Mach number ranging from 0.01 to 0.5 are presented in [24], with the focus on the effect of compressibility on heat transfer intensification in the region occupied by horseshoe-shaped vortex structures. In the next section, we present the results of numerical simulations and parallel efficiency testing for a representative three-dimensional problem of high-speed gas flow with strong effects of viscous-inviscid interaction.

## 4.2. Test Problem

The problem formulation is based on the data of the computational and experimental study [47], which investigated the structure of laminar supersonic air flow (Mach number $M_\infty = 6.7$, Prandtl number $Pr = 0.71$, and ratio of specific heats $\gamma = 1.4$) past a fin mounted on a plate (Fig. 8). Following [47], the flow is assumed to be laminar throughout the domain. The Reynolds number, calculated from the bluntness diameter ($D = 2.5$ mm), corresponds to the minimum value considered in [47]: $Re_D = 1.25 \times 10^4$. The free-stream total temperature is $T_0 = 630$ K. The computational domain is shown in Fig. 8. The fin is positioned at a distance of $L_{plate} = 145$ mm from the plate's leading edge. The other dimensions of the computational domain are taken from [47] as follows: $R = 76.5$ mm, $L_f = 60$ mm, and $H = 25$ mm.

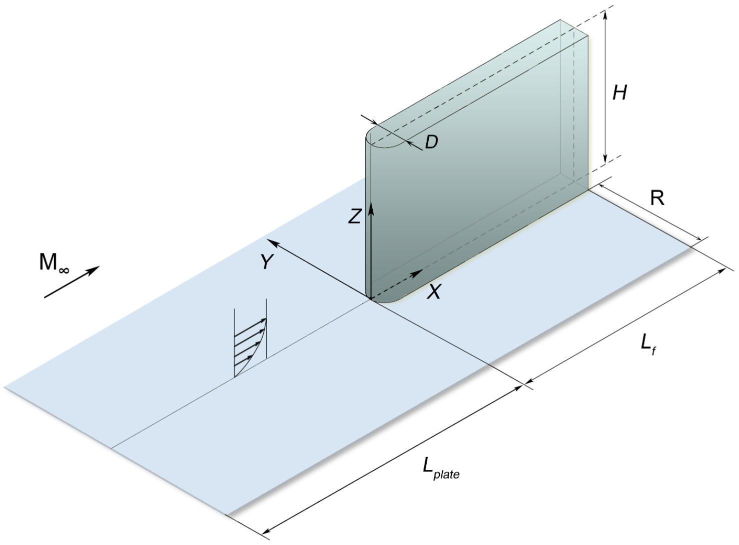


**Figure 8.** Test problem definition

The full three-dimensional Navier–Stokes equations were solved treating air as a perfect gas. The temperature dependence of the viscosity coefficient was evaluated using Sutherland's law. An uniform flow was specified at the inlet boundary of the computational domain, and the no-slip condition was applied on the body surface and the plate. Non-reflecting boundary conditions were specified on the lateral and upper boundaries, and the zero-gradient condition was specified at the outlet. The body surface and the plate were maintained at a constant temperature, $T_w = 300\ K$, corresponding to a temperature ratio of $T_w/T_\infty = 4.76$.

The convective fluxes on the faces of control volumes were evaluated using the second-order AUSM scheme with the van Albada TVD limiter. A nominally second-order scheme was used for evaluating viscous fluxes. The simulation was performed based on the time-dependent formulation, using an implicit time-advancing scheme of second-order accuracy (the backward three-layer scheme). Under considered conditions, the transient process converged to a steady-state solution. The basic simulation was performed using a quasi-structured grid with 20 million cells. The grid had approximately 70 cells across the incoming boundary layer, which had a thickness of about $1.2D$ before separation. The first near-wall cell is approximately 0.01 mm in the normal direction. For testing grid sensitivity, the simulation was also performed on a finer grid, which consisted of 50 mln cells.

Figure 9 provides a general overview of the flow structure. The figure shows the volume streamlines and the relative heat flux distribution over the body and plate surfaces ($q_{w0}$ is the heat flux calculated for a plate without an obstacle). The flow's primary characteristics include the formation of a separation region with a system of horseshoe vortices that extend around the fin's side, and a pronounced non-monotonicity in the heat flux distribution within the junction region.

Figure 10 presents the calculated distributions of the plate-surface nondimensional heat flux (normalized by the heat flux local values $q_{wp}$, which were calculated for the smooth plate without an obstacle at the corresponding points) along the symmetry line (a) and along a cross line $x/D = 1$ (b). Additionally, experimental and numerical data from [47] are included. The

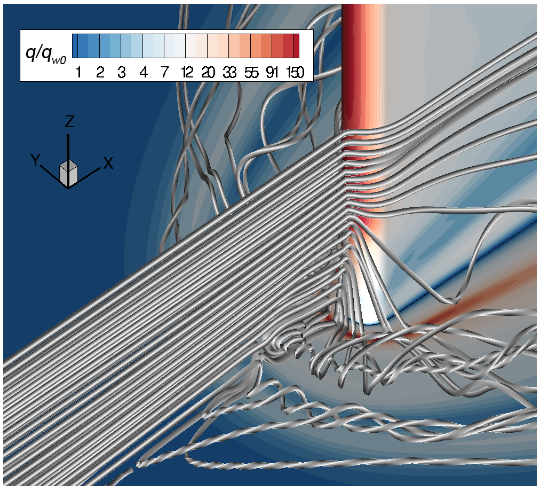


**Figure 9.** Visualization of the three-dimensional flow structure

primary specificity in the calculated heat flux distributions along the symmetry line is that the distributions exhibit two local maxima, in addition to the global one (it is outside the field of the Fig. 10). Our computational results align well with the computational data reported in [47]. Let us note that the cited paper authors believed that good agreement of their numerical data with the experimental results had been achieved. Also, there is negligible difference between the results obtained using the base grid and the refined grid for both solutions, confirming the adequacy of the grid resolution.

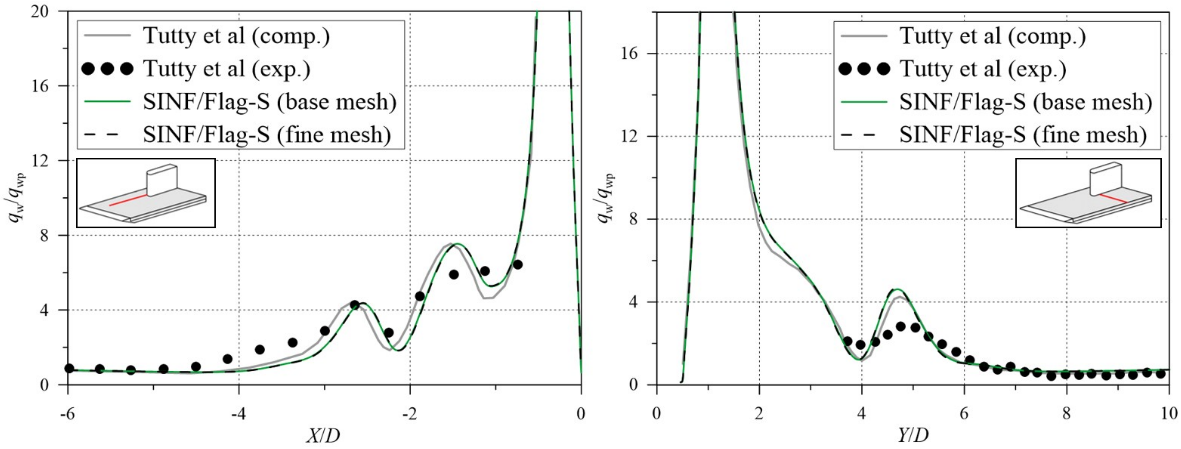


**Figure 10.** Relative heat flux distribution: (left) along the line of symmetry on the plate $Y/D = 0$; (right) along line $X/D = 1$

Testing the efficiency of code parallelization was carried out by running simulation using a different number of cores of the Polytechnic Supercomputer Center cluster. The time for 100 iterations was calculated. The results are presented in Fig. 11.

Concluding this section, one should note that the problems of non-uniqueness of the solution can occur when studying this or a related class of gas flow (see [22, 23, 25]). Since, additionally to geometry, there are many other parameters determining such tasks (Reynolds number, Mach number, temperature factor, relative thickness of incoming boundary layer), getting knowledge on the challenging problem of the non-uniqueness of the solution assumes running a huge amount of parametric refined calculations on power supercomputers.

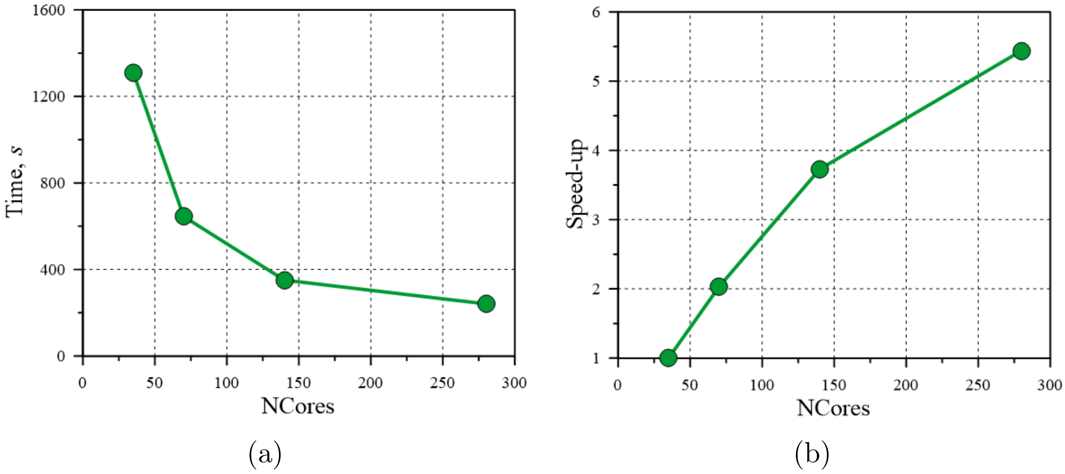


**Figure 11.** Parallel scalability: calculation time (a) and speed-up (b)

## Conclusion

In the paper, we have presented our recent experience in assessing several advanced numerical techniques implemented in the in-house SINF/Flag-S code.

To improve the efficiency for solving systems of linear algebraic equations, which appears as a result of the Poisson equation discretization, an original parallelized geometric multigrid method was implemented in the SINF/Flag-S code. Dealing with a model problem of conduction heat transfer in a cubic domain, it was shown that the implemented multigrid method, used both as a separate linear solver and as a preconditioner to the GMRES method, demonstrates very good performance and scalability both in single-threaded and in parallel mode. It was also established that the multigrid method implemented in the last versions of the ANSYS Fluent code is significantly inferior to the one implemented in the SINF/Flag-S, both in terms of problem-solving time and scalability.

To improve the efficiency of solving essentially non-stationary problems (in particular, in turbulent flow simulation using eddy-resolved approaches), an original fully-implicit fractional step method was implemented in the SINF/Flag-S. The algorithm approbation was illustrated by two examples of unsteady flows. The first one is the DNS of turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container. Here, the computational domain has a relatively simple geometry, but the flow is characterized by intense turbulent motion with strongly different turbulent scales: a large-scale circulation has the form of a vortex covering the entire volume, while the size of the vortices responsible for dissipation is several orders of magnitude smaller. Accurate simulation of all turbulent scales requires very small time steps, so the non-iterative solver is preferable, especially when a large enough sample is needed for reliable statistics. Results of testing of the implemented fraction step method in parallel mode showed relatively good scaling efficiency, even in the case of numerous cores: although at first it degrades with increasing numbers of cores, it later goes to a level of about 0.5, which is quite acceptable. The second example is related to simulation of unsteady free convective flow around a subsea cooler model. This industrial problem is characterized by a numerous heated elements (pipes) interacting with the surrounding water, so grids of very large size should be used for simulation. On the other hand, the flow in such configurations may be simultaneously laminar in some regions of the whole flow and transient or even turbulent in other regions. The use of the fractional step method also gives a significant increase in computational performance here.

A reliable tool for numerical simulation of compressible gas flows in a wide range of the Mach numbers is important for various practical applications, including turbomachinery and aeronautics. With considering a complex 3D model problem of the viscous-inviscid interaction of supersonic flow past a blunt fin mounted on a plate, we have demonstrated the ability of the coupled density-based solver implemented in the SINF/Flag-S code to accurately resolve both the shock wave structure and the regions of substantially subsonic flow, where viscous effects play a significant role (in the boundary layer and within horseshoe vortices). The performed investigation of parallel scalability showed fairly good speed-up of the calculations when using up to several hundred computational cores.

## Acknowledgments

The results of the work were obtained using computational resources of Peter the Great Saint-Petersburg Polytechnic University Supercomputing Center (www.spbstu.ru).

## References

1. Abalakin, I.V., Bobkov, V.G., Kozubskaya, T.K., *et al.*: Numerical simulation of flow around rigid rotor in forward flight. Fluid Dyn. 55(4), 534–544 (Jul 2020). `https://doi.org/10.1134/S0015462820040011`

2. Ahlers, G., Grossmann, S., Lohse, D.: Heat transfer and large scale dynamics in turbulent Rayleigh–Bénard convection. Rev. Mod. Phys. 81, 503–537 (Apr 2009). `https://doi.org/10.1103/RevModPhys.81.503`

3. Anunciação, M., Augusto Villela Pinto, M., Neundorf, R.: Solution of the Navier–Stokes equations using projection method and preconditioned conjugated gradient with multigrid and ILU solver. Rev. Int. Métodos Numér. Cálc. Diseño Ing. 36(1), 17 (2020). `https://doi.org/10.23967/j.rimni.2020.01.003`

4. Bakhvalov, P.A., Kozubskaya, T.K.: Cell-centered quasi-one-dimensional reconstruction scheme on 3D hybrid meshes. Math. Models Comput. Simul. 8(6), 625–637 (Nov 2016). `https://doi.org/10.1134/S2070048216060053`

5. Barth, T., Linton, S.: An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation. In: 33rd Aerospace Sciences Meeting and Exhibit. `https://doi.org/10.2514/6.1995-221`

6. Bevan, R.L.T., Boileau, E., van Loon, R., *et al.*: A comparative study of fractional step method in its quasi-implicit, semi-implicit and fully-explicit forms for incompressible flows. Int. J. Numer. Methods Heat Fluid Flow 26(3/4), 595–623 (May 2016). `https://doi.org/10.1108/HFF-06-2015-0233`

7. Carrión, M., Woodgate, M., Steijl, R., Barakos, G.: Implementation of all-Mach Roe-type schemes in fully implicit CFD solvers – demonstration for wind turbine flows. International

Journal for Numerical Methods in Fluids 73(8), 693–728 (2013). https://doi.org/10.1002/fld.3818

8. Chorin, A.J.: Numerical solution of the Navier–Stokes equations. Math. Comput. 22(104), 745 (Oct 1968)

9. Darwish, M., Moukalled, F.: A Fully Coupled Navier–Stokes Solver for Fluid Flow at All Speeds. Numerical Heat Transfer, Part B: Fundamentals 65(5), 410–444 (2014). https://doi.org/10.1080/10407790.2013.869102

10. De Michele, C., Capuano, F., Coppola, G.: Fast-Projection Methods for the Incompressible Navier–Stokes Equations. Fluids 5(4) (2020). https://doi.org/10.3390/fluids5040222

11. Dellacherie, S., Jung, J., Omnes, P., Raviart, P.A.: Construction of modified Godunov-type schemes accurate at any Mach number for the compressible Euler system. Mathematical Models and Methods in Applied Sciences 26(13), 2525–2615 (2016). https://doi.org/10.1142/S0218202516500603

12. Fedorenko, R.P.: A relaxation method for solving elliptic difference equations. USSR Computational Mathematics and Mathematical Physics 1(4), 1092–1096 (1962). https://doi.org/10.1016/0041-5553(62)90031-9

13. Gorobets, A.V.: An approach to the implementation of the multigrid method with full approximation for CFD problems. Computational Mathematics and Mathematical Physics 63(11), 2150–2161 (2023). https://doi.org/10.1134/S0965542523110106

14. Gorobets, A.V., Soukov, S.A., Magomedov, A.R.: Heterogeneous parallel implementation of a multigrid method with full approximation in the NOISETTE code. Matem. Mod. 36(2), 129–146 (2024). https://doi.org/10.20948/mm-2024-02-08

15. Gropp, W., Lusk, E., Skjellum, A.: Using MPI (2nd ed.): portable parallel programming with the message-passing interface. MIT Press, Cambridge, MA, USA (1999)

16. Gyles, B.R., Haegland, B., Dahl, T.B., *et al.*: Natural Convection - Subsea Cooling: Theory, Simulations, Experiments and Design. International Conference on Offshore Mechanics and Arctic Engineering, vol. Volume 1: Offshore Technology; Polar and Arctic Sciences and Technology, pp. 11–20 (Jun 2011). https://doi.org/10.1115/OMAE2011-49030

17. Ivanov, N., Ris, V., Tschur, N., Zasimova, M.: Numerical simulation of conjugate heat transfer in a tube bank of a subsea cooler based on buoyancy effects. J. Phys. Conf. Ser. 745, 032058 (Sep 2016). https://doi.org/10.1088/1742-6596/745/3/032058

18. Ivanov, N., Ris, V.V., Tschur, N.A., Zasimova, M.: Effect of gas flow direction on passive subsea cooler effectiveness. Comput. Therm. Sci. Int. J. 11(1-2), 1–16 (2019). https://doi.org/10.1615/COMPUTTHERMALSCIEN.2018024704

19. Jan, Y.J., Sheu, T.W.H.: A quasi-implicit time advancing scheme for unsteady incompressible flow. Part I: Validation. Computer Methods in Applied Mechanics and Engineering 196(45), 4755–4770 (2007). https://doi.org/10.1016/j.cma.2007.06.008

20. Jung, J., Perrier, V.: Steady low Mach number flows: Identification of the spurious mode and filtering method. Journal of Computational Physics 468, 111462 (2022). https://doi.org/10.1016/j.jcp.2022.111462

21. Kim, J., Moin, P.: Application of a fractional-step method to incompressible Navier–Stokes equations. Journal of Computational Physics 59(2), 308–323 (1985). https://doi.org/10.1016/0021-9991(85)90148-2

22. Kolesnik, E.V., Smirnov, E.M.: Supersonic laminar flow past a blunt fin: Duality of the numerical solution. Tech. Phys. 66(6), 741–748 (Jun 2021). https://doi.org/10.1134/S1063784221050133

23. Kolesnik, E.V., Smirnov, E.M.: Duality of the stream pattern of supersonic viscous gas flow past a blunt-fin junction: The effect of a low sweep angle. Fluid Dyn. 58(1), 1–8 (Feb 2023). https://doi.org/10.1134/S0015462822601887

24. Kolesnik, E.V., Smirnovsky, A.A., Smirnov, E.M.: Compressibility effect on heat transfer intensified by horseshoe vortex structures in turbulent flow past a blunt-body and plate junction. Journal of Physics: Conference Series 1565(1), 012104 (jun 2020). https://doi.org/10.1088/1742-6596/1565/1/012104

25. Kolesnik, E., Smirnov, E., Babich, E.: Dual numerical solution for 3D supersonic laminar flow past a blunt-fin junction: Change in temperature ratio as a method of flow control. Fluids 8(5), 149 (May 2023)

26. Koobus, B., Lallemand, M.H., Dervieux, A.: Unstructured volume-agglomeration MG: Solution of the Poisson equation. International Journal for Numerical Methods in Fluids 18(1), 27–42 (1994). https://doi.org/10.1002/fld.1650180103

27. Lasalle, D., Karypis, G.: Multi-threaded graph partitioning. In: 2013 IEEE 27th International Symposium on Parallel and Distributed Processing. pp. 225–236 (2013). https://doi.org/10.1109/IPDPS.2013.50

28. Le Touze, C., Murrone, A., Guillard, H.: Multislope MUSCL method for general unstructured meshes. Journal of Computational Physics 284, 389–418 (2015). https://doi.org/10.1016/j.jcp.2014.12.032

29. Lien, F.S., Ji, H., Ryan, S.D., *et al.*: A coupled multigrid solver with wall functions for three-dimensional turbulent flows over urban-like obstacles. International Journal for Numerical Methods in Fluids 95(9), 1349–1371 (2023). https://doi.org/10.1002/fld.5189

30. Liu, Y.: An efficient numerical method for highly loaded transonic cascade flow. J. Eng. Math. 60(1), 115–124 (Jan 2008)

31. Mani, M., Dorgan, A.J.: A perspective on the state of aerospace computational fluid dynamics technology. Annual Review of Fluid Mechanics 55, 431–457 (2023). https://doi.org/10.1146/annurev-fluid-120720-124800

32. Marmignon, C., Cantaloube, B., Le Pape, M.C., *et al.*: Development of an agglomeration multigrid technique in the hybrid solver elsA-H. In: Seventh International Conference on Computational Fluid Dynamics (ICCFD7). p. 15 (2012)

33. Matsui, K.: A projection method for Navier–Stokes equations with a boundary condition including the total pressure. Numerische Mathematik 152(3), 663–699 (2022). https://doi.org/10.1007/s00211-022-01323-x

34. Mavriplis, D.: Multigrid techniques for unstructured meshes. ICASE Report (1995)

35. Nishikawa, H., Diskin, B., Thomas, J.L.: Critical study of agglomerated multigrid methods for diffusion. AIAA J. 48(4), 839–847 (Apr 2010). https://doi.org/10.2514/1.J050055

36. Patankar, S.V., Spalding, D.B.: A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. International Journal of Heat and Mass Transfer 15(10), 1787–1806 (1972). https://doi.org/10.1016/0017-9310(72)90054-3

37. Patel, A.: Development of an adaptive RANS solver for unstructured hexahedral meshes. PhD thesis, Université libre de Bruxelles, Brussels, Belgium (2003)

38. Quirk, J.J.: A contribution to the great Riemann solver debate. Int. J. Numer. Methods Fluids 18(6), 555–574 (Mar 1994). https://doi.org/10.1002/fld.1650180603

39. Rodionov, A.V.: Artificial viscosity to cure the carbuncle phenomenon: The three-dimensional case. J. Comput. Phys. 361, 50–55 (May 2018). https://doi.org/10.1016/j.jcp.2018.02.001

40. Rogers, S.E., Kwak, D.: Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations. AIAA Journal 28(2), 253–262 (1990). https://doi.org/10.2514/3.10382

41. Saad, Y.: Iterative methods for sparse linear systems. SIAM, Philadelphia, MS, 2 edn. (2003)

42. Smirnov, S.I., Abramov, A.G., Smirnov, E.M.: Numerical simulation of turbulent Rayleigh–Bénard mercury convection in a circular cylinder with introducing small deviations from the axisymmetric formulation. J. Phys. Conf. Ser. 1359(1), 012077 (Nov 2019). https://doi.org/10.1088/1742-6596/1359/1/012077

43. Smirnov, S., Smirnov, E.: Direct numerical simulation of the turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container. St. Petersburg State Polytechnical University Journal. Physics and Mathematics 13, 13–23 (2020). https://doi.org/10.18721/JPM.13102

44. Smirnov, S.I., Smirnov, E.M., Smirnovsky, A.A.: Endwall heat transfer effects on the turbulent mercury convection in a rotating cylinder. St Petersbg. Polytech. Univ. J. Phys. Math. 3(2), 83–94 (Jun 2017). https://doi.org/10.1016/j.spjpm.2017.05.009

45. Toro, E.F.: Riemann solvers and numerical methods for fluid dynamics. Springer, Berlin, Germany, 3 edn. (Apr 2009)

46. Turkel, E.: Preconditioned methods for solving the incompressible and low speed compressible equations. Journal of Computational Physics 72(2), 277–298 (1987). https://doi.org/10.1016/0021-9991(87)90084-2

47. Tutty, O.R., Roberts, G.T., Schuricht, P.H.: High-speed laminar flow past a fin–body junction. Journal of Fluid Mechanics 737, 19–55 (2013). https://doi.org/10.1017/jfm.2013.541

48. Van Doormaal, J.P., Raithby, G.D.: Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numer. Heat Trans. 7(2), 147–163 (Apr 1984). https://doi.org/10.1080/01495728408961817

49. Weiss, J.M., Smith, W.A.: Preconditioning applied to variable and constant density flows. AIAA Journal 33(11), 2050–2057 (1995). https://doi.org/10.2514/3.12946

DOI: 10.14529/jsfi240203

# A Further Development of the Asynchronous GPU CABARET Method for Jet-Noise Modelling

*Igor A. Solntsev*[1], *Sergey A. Karabasov*[1], *Georgy A. Faranosov*[1], *Oleg P. Bychkov*[1]

A new asynchronous modification of the CABARET method is proposed for the solution of Navier–Stokes equations in the Large Eddy Simulation regime. The modification is based on improvement of the asynchronous extrapolation step both for Euler and Navier–Stokes CABARET solver. The algorithm is implemented as a parallel code for NVIDIA GPU using multiple CUDA-cores with MPI multi-CPU support. The algorithm accelerated on Graphics Processing Units (GPUs) is applied for the jet flow simulations in the Wall Model Large Eddy Simulation framework. The efficiency of code parallelization is discussed. The suggested asynchronous CABARET algorithm provides an almost 5000 times acceleration of calculations compared to a single CPU core, and allows us to calculate 300 convective times of jet development per day on a grid of 16 million cells. The flow solutions are analysed and compared with TsAGI anechoic chamber experimental data. It is shown that the structure of the jet flow is reproduced correctly, capturing low-amplitude instability waves in the jet potential core and fine-scale turbulent fluctuations in the near-field. Far-field noise predictions in the Ffowcs Williams–Hawking formulation with the azimuthal decomposition of the far-field radiation reproduce the nontrivial spectra and directivities of individual far-field acoustic modes.

*Keywords: CABARET, asynchronous, GPU, parallel computing, jet noise.*

## Introduction

Unsteady computational methods play an increasingly important role in jet noise modelling and design optimization studies of current propulsion system [2–4, 14, 17, 23, 24]. For simulations of noise generated by aerodynamic flows, high-resolution numerical methods are typically used. Specifically, these methods should not only have small numerical dissipation, but also accurately preserve dispersion properties of the governing gas dynamics equations. One example of such high-resolution methods is CABARET (Compact Accurately Boundary-Adjusting high-REsolution Technique) [10, 13] which was previously applied for modelling of high-speed jets [7, 11, 18, 21] and aerofoil flows [1, 19] using the Monotonically Integrated Large Eddy Simulation (MILES) framework for turbulence modelling. CABARET Navier–Stokes solver method was extended in previous work [27] to multi-zonal meshes including rotating meshes typical of propeller applications.

While the linear wave properties of the CABARET scheme are exceptionally good for Courant–Friedrichs–Lewy ($CFL$) numbers around 0.5, for small $CFL$ numbers its accuracy deteriorates [10]. However, small $CFL$ numbers are unavoidable is some cases, e.g. during time marching at a constant global time step on non-uniform meshes. The asynchronous time-stepping technique with an optimal cell-local $CFL$ to avoid accuracy deterioration was implemented for the CABARET Euler solver in [25] and extended to Navier–Stokes equations [20]. This technique is improved in the current work by modification of the asynchronous extrapolation step making it even more computationally efficient.

The article is organised as follows: In Section 1, the governing equations in the form of hyperbolic conservation laws with a heterogeneous right-hand-side are outlined. In Section 2,

[1] Central Aerohydrodynamic Institute, Zhukovsky, Russia

the baseline CABARET algorithm for gas dynamics is briefly introduced. In Section 3, the new asynchronous time-stepping technique is presented. In Section 4, the efficiency of parallel asynchronous GPU code is discussed. Numerical results and comparison with TsAGI anechoic chamber experimental data are presented in Section 5.

## 1. Hyperbolic Form of Governing Equations

Following [27], the governing Navier–Stokes equations can be formulated using hyperbolic equations with an inhomogeneous right-hand side, which in Cartesian coordinates have the form:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}+\frac{\partial \mathbf{H}}{\partial z}=\mathbf{Q}, \tag{1}$$

where the conservative variables $\mathbf{U}$ and the conservation fluxes $\mathbf{F}, \mathbf{G}$ and $\mathbf{H}$ are

$$\begin{aligned}
&\mathbf{U}=(\rho,\rho u,\rho v,\rho w,\rho e)^{\mathsf{T}},\\
&\mathbf{F}=(\rho u,\rho uu+p,\rho uv,\rho uw,\rho ue+up)^{\mathsf{T}},\\
&\mathbf{G}=(\rho v,\rho vu,\rho vv+p,\rho vw,\rho ve+vp)^{\mathsf{T}},\\
&\mathbf{H}=(\rho w,\rho wu,\rho wv,\rho ww+p,\rho we+wp)^{\mathsf{T}}.
\end{aligned}$$

Here $u$, $v$, $w$ are the velocity components in the $x$-, $y$-, and $z$-direction in the relative frame, $\rho$, $e$, $p$ are density, total energy, and pressure. The ideal gas equation of state is assumed, where pressure and density are related to internal energy via the specific heat ratio, $\gamma$.

The right-hand-side source in (1) includes contributions of viscous terms defined by components of the viscous stress tensor $\sigma_{ij}$, where $i, j = x, y, z$

$$\begin{aligned}
\mathbf{Q}=(0,\ &\tfrac{\partial}{\partial x}\sigma_{xx}+\tfrac{\partial}{\partial y}\sigma_{xy}+\tfrac{\partial}{\partial z}\sigma_{xz},\tfrac{\partial}{\partial x}\sigma_{yx}+\tfrac{\partial}{\partial y}\sigma_{yy}+\tfrac{\partial}{\partial z}\sigma_{yz},\tfrac{\partial}{\partial x}\sigma_{zx}+\tfrac{\partial}{\partial y}\sigma_{zy}+\tfrac{\partial}{\partial z}\sigma_{zz},\\
&\left[\tfrac{\partial}{\partial x}(u\sigma_{xx}+v\sigma_{xy}+w\sigma_{xz})+\tfrac{\partial}{\partial y}(u\sigma_{yx}+v\sigma_{yy}+w\sigma_{yz})+\tfrac{\partial}{\partial z}(u\sigma_{zx}+v\sigma_{zy}+w\sigma_{zz})\right])^{\mathsf{T}}.
\end{aligned} \tag{2}$$

The hyperbolic part of the governing equations (1) comprises 5 characteristics corresponding to the local Riemann invariants – one entropy wave, two pressure waves, and two contact waves in each spatial coordinate direction. The entropy and contact waves propagate at the characteristic speed $\lambda_1 = uk_x + vk_y + wk_z$, and the pressure waves propagate at $\lambda_{2,3} = uk_x + vk_y + wk_z \pm a$, where $a = \sqrt{\gamma p/\rho}$ is the local sound speed and $(k_x, k_y, k_z)^{\mathsf{T}}$ is a propagation direction.

## 2. 3-steps CABARET Algorithm

Following [7, 13], the Compact Accurately Boundary-Adjusting high-REsolution Technique (CABARET) algorithm for solving hyperbolic equations with an inhomogeneous right-hand-side (1) is presented as a three step procedure. Let us consider a hexagonal spatial grid, where each centre of the control volume (cell-centre point), $c$, is surrounded by three pairs of opposite grid faces (face-centre points), $c-$ and $c+$, corresponding to a certain grid direction within the cell. By applying the Gauss–Ostrogradski theorem to integrate the governing system of conservation laws (1) in space and time, the conservative predictor and corrector steps of the CABARET algorithm are given by

$$\frac{\mathbf{U}_c^{n+\frac{1}{2}}-\mathbf{U}_c^n}{\tau^n/2}+\frac{(s_x\mathbf{F}_{c+}^n-s_x\mathbf{F}_{c-}^n)+(s_y\mathbf{G}_{c+}^n-s_y\mathbf{G}_{c-}^n)+(s_z\mathbf{H}_{c+}^n-s_z\mathbf{H}_{c-}^n)}{\Omega_c}=\mathbf{Q}(\mathbf{U}_c^n), \tag{3}$$

$$\frac{\mathbf{U}_c^{n+1}-\mathbf{U}_c^{n+\frac{1}{2}}}{\tau^n/2}+\frac{(s_x\mathbf{F}_{c+}^{n+1}-s_x\mathbf{F}_{c-}^{n+1})+(s_y\mathbf{G}_{c+}^{n+1}-s_y\mathbf{G}_{c-}^{n+1})+(s_z\mathbf{H}_{c+}^{n+1}-s_z\mathbf{H}_{c-}^{n+1})}{\Omega_c}=\mathbf{Q}(\mathbf{U}_c^{n+\frac{1}{2}}), \tag{4}$$

where $\mathbf{U}_c$ is the vector of conservation variables referred to the cell centre, $\mathbf{F}_{c+}, \mathbf{F}_{c-}$, $\mathbf{G}_{c+}, \mathbf{G}_{c-}$ and $\mathbf{H}_{c+}, \mathbf{H}_{c-}$ denote the pairs of conservation fluxes in the $x$-, $y$- and $z$- directions computed at the face centres, $\Omega_c$ and $s_x, s_y, s_z$ are the cell volume and the face normal areas respectively, $\tau^n$ is a time step, and the sub-index and super-index $n, n+1/2, n+1$ denote the space and time discretisation. For both the predictor and corrector steps, the right-hand-side terms $\mathbf{Q}$ are computed by substituting the most recently updated conservation variable in the source equation (2).

It can be noted that the corrector step (4) involves the unknown flux functions from the new time step, $n+1$. These functions are updated at the intermediate characteristic decomposition and extrapolation step, which follows right after the predictor step (3). In the characteristic decomposition and extrapolation step, we determine flux variables in the centers of faces on a new layer $n+1$ so that the characteristic invariants are computed at the center of each cell $\mathbf{R}_c^{n+1/2}$ and at the centers of the faces

$$\mathbf{R}=\begin{pmatrix} r \\ q \end{pmatrix}, \quad q=\varpi-Gp^{\mu}, \quad r=\varpi+Gp^{\mu}, \quad s=\ln\left(p/\rho^{\gamma}\right),$$

where $\mu=\frac{\gamma-1}{2\gamma}, G=\frac{2\gamma}{\gamma-1}\exp(\frac{s}{2\gamma})$, and $\varpi$ corresponds to the speed $u$, $v$ or $w$ for each direction $x$, $y$ and $z$.

The computation of the local Riemann invariants at the current and mid-time level, $n$, $n+1/2$, is followed by extrapolation of the cell-face variables to the new time step $n+1$ in each grid cell, $c$,

$$\tilde{\mathbf{R}}_{c+}^{n+1}=2\mathbf{R}_c^{n+1/2}-\mathbf{R}_{c-}^{n} \quad \text{and} \quad \tilde{\mathbf{R}}_{c-}^{n+1}=2\mathbf{R}_c^{n+1/2}-\mathbf{R}_{c+}^{n}.$$

For each face that is an interface of two adjacent cells $L$ and $R$ the extrapolated values from each side of the contact $\tilde{\mathbf{R}}_{L+}^{n+1}$, $\tilde{\mathbf{R}}_{R-}^{n+1}$ are adjusted using the maximum principle. Two sets of corrected invariants on the cell contact surface are then used and the signs of the characteristic wave speeds are accounted for to select a unique set of invariants for each cell face. Having defined the complete set of characteristic variables, the $(5\times 5)$ linear system solved to obtain a solution of the Riemann problem, and the values of the variables and fluxes $\mathbf{F}_{c+}^{n+1}$, $\mathbf{F}_{c-}^{n+1}$, $\mathbf{G}_{c+}^{n+1}$, $\mathbf{G}_{c-}^{n+1}$, $\mathbf{H}_{c+}^{n+1}$, $\mathbf{H}_{c-}^{n+1}$ are computed at the new time level $n+1$ for each cell face $c+, c-$. The results are then substituted in the conservation flux functions for the conservative corrector step (4).

## 3. Asynchronous Time-stepping

Asynchronous time stepping, i.e., when the solution in cells of different sizes is updated at different rates and adjusted to the cell-local $CFL$ number rather than to a global one, is an effective way of improving the efficiency of explicit methods with highly non-uniform grids without any loss of accurate linear waves propagation properties. In particular, the asynchronous time-stepping technique was implemented for the CABARET Euler [25] and the Navier–Stokes solver [20] keeping the following important properties: (i) simplicity and compactness of the original computational stencil, (ii) strict conservation property and (iii) a built-in recipe for the treatment of inactive flow regions. This technique is improved in the current work by modification of the asynchronous extrapolation step both for the Euler and for the Navier–Stokes solver.

To illustrate the main concepts, let us consider a one-dimensional flow on a non-uniform grid along the $X$ axis with a variable cell size $h_c$ at time $t^n$. We will assume that the upper and lower indices correspond to evolving space and time. Let us determine the value of the optimal time step in each of the cells from the relation $\tau_c^n = CFL \cdot h_c/(|u_c^n| + a)$, here $u$ is the local flow speed, and $a$ is the local speed of sound, assuming $CFL = 0.5$. The next active time step of the asynchronous algorithm, $\tau^n$ is determined by the step from the current moment $t^n$ to the next state $t^{n+1}$, determined by the minimum value for all grid cells.

To increase computational efficiency, the optimal local time step in each of the cells are replaced by closest values from a pre-defined set of acceptable values with a certain minimum step. Following [20, 25], the update is not performed for all cells, but only for cells with a minimum next temporal state $t^{n+1}$. We will denote such cells as active cells. The calculation of conservative variables at asynchronous predictor and corrector steps is performed according to the standard algorithm, but for different moments in time $t_c^n + \bar{\tau}_c^n/2$ and $t_c^n + \bar{\tau}_c^n$, determined by the local step in each of the cells. Equations (3),(4) for 1D take the form

$$\frac{\mathbf{U}_c^{n+1/2} - \mathbf{U}_c^n}{\bar{\tau}_c^n/2} + \frac{(\mathbf{F}_{c+}^n - \mathbf{F}_{c-}^n)}{h_c} = \mathbf{Q}(\mathbf{U}_c^n),$$

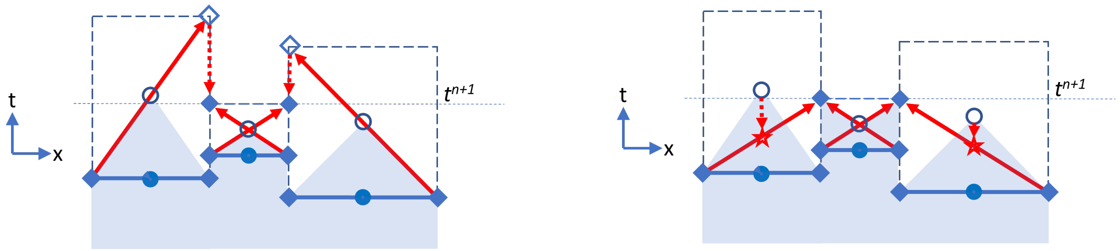
$$\frac{\mathbf{U}_c^{n+1} - \mathbf{U}_c^{n+1/2}}{\bar{\tau}_c^n/2} + \frac{(\mathbf{F}_{c+}^{n+1} - \mathbf{F}_{c-}^{n+1})}{h_c} = \mathbf{Q}(\mathbf{U}_c^{n+1/2}).$$

The main novelty of the suggested algorithm in the current work is due to the change made at the characteristic extrapolation step. Here, to calculate the fluxes through faces of an active cell, it is necessary to perform extrapolation inside the cell while obtaining external values extrapolated to faces from neighboring cells. Extrapolation within a cell is performed in a standard manner. However, the standard extrapolation in neighboring cells with large time steps leads to extrapolated values not for the required time point $t^{n+1}$, but for some future time points.

To obtain the required values, [20, 25] use inverse interpolation of extrapolated values (Fig. 1a). Instead, here we use a characteristic extrapolation algorithm based on the modified values of conservative variables in neighboring cells. Rather than using the intermediate values obtained in neighboring cells at the predictor step, the values obtained by their linear interpolation in time to the required point in time $\tilde{t}_c^{n+1}$ are used, ensuring the construction of the correct extrapolation stencil (Fig. 1b). For example, for extrapolation to the edge of an active cell $c$, this moment in time for the neighboring cell $cN$ is defined as $\tilde{t}_{cN}^{n+1/2} = (t_c^{n+1} - t_{cN}^n)/2$, and the intermediate values of conservative variables in the neighboring cell are interpolated to the desired time point as

$$\tilde{\mathbf{U}}_{cN}^{n+1/2} = \mathbf{U}_{cN}^n + (\mathbf{U}_{cN}^{n+1/2} - \mathbf{U}_{cN}^n)(\tilde{t}_{cN}^{n+1/2} - t_{cN}^n)/\bar{\tau}_{cN}^n.$$

This modification allows us to avoid the additional calculation and storage of large arrays of flow invariants at the centers of cells and the centers of faces, which increases the computational efficiency of the final algorithm. The extrapolation step is now performed in asynchronous mode according to the basic algorithm described in Section 2, replacing intermediate values of conservative variables in neighboring cells $\mathbf{U}_{cN}^{n+1/2}$ with their interpolated values $\tilde{\mathbf{U}}_{cN}^{n+1/2}$.



(a) (b)

**Figure 1.** The baseline characteristic extrapolation (a) and extrapolation with the new suggested pre-interpolated conservative variables at mid-time levels (b)

In Fig. 1, solid horizontal lines correspond to different points in time, depending on the update history of each cell. The dashed horizontal lines indicate the optimal next time steps for each of the cells. Arrows indicate extrapolation (solid arrows) and interpolation (dotted arrows), stars indicate interpolated values of the new algorithm.

The discrepancy between the time steps of contacting cells leads to the fact that one large time step in a certain cell may correspond to several small steps in a neighboring cell. To keep fluxes conservative in an asynchronous scheme, we apply fluxes synchronization proposed in the previous works [20, 25]: fluxes in a cell with a large step are adjusted according to the total values in cells with a smaller step. Algorithm 1 depicts a pseudocode of the resulting asynchronous algorithm.

## 4. Parallelization of Asynchronous Code

The asynchronous CABARET algorithm is implemented as a parallel code using the CUDA parallel computing architecture for NVIDIA GPU devices and the MPI message passing interface between parallel CPU host processes. Using these tools allows you to use two different parallelization approaches to speed up calculations on grids of several million cells and more.

To speed up calculations of the CABARET algorithm on multi-core CPUs, the MPI interface allows parallel calculations for individual parts of the decomposed mesh. In this case, the grid cells in each decomposition zone are processed by the corresponding CPU core sequentially one after another. The maximum ideal value for speeding up calculations is determined by the number of CPU cores. However, the need to exchange data between cells at the boundaries of decomposition zones by calling MPI-synchronization routines reduces the efficiency of MPI-parallelization.

The CUDA parallel computing architecture and modern NVIDIA GPU graphics accelerators allow a more efficient approach for grids with a moderate number of grid cells. If the mesh fits completely into the memory of the graphics accelerator, then mesh decomposition is not required. Grid cells are processed in parallel in groups which size is determined by the number of CUDA cores. Modern GPU card such as A100 and H100 have about 7000 and 15000 CUDA cores respectively and 80GB onboard memory. The current code implementation with modified asynchronous step requires 1GB of memory for 1 million grid cells in single precision mode. This allows maximum computational efficiency of the code for grids of up to 80 million cells. Combination of CUDA and MPI approaches will increase this limit for multi-GPU computers and clusters.

To assess the actual efficiency of MPI code parallelization for a CPU, a series of calculations of the jet flow from a test nozzle was performed on grids with different numbers of nodes, 2 million

**Algorithm 1** Asynchronous Time-stepping

```
// Initialize the initial states at $t = t^0$ for conservative variables in the centers of cells $\mathbf{U}_c^0$ and for primitive variables and fluxes $\mathbf{F}_{c-}^0$, $\mathbf{F}_{c+}^0$, initialize the initial values of the list of active/updated cells $Active_c^n = .false.$
Set $n = 0$, $t = t^0$
For all cells: initialize $\mathbf{U}_c^0$
For all faces: initialize $\mathbf{F}_{c-}^0$, $\mathbf{F}_{c+}^0$
For all cells: initialize $Active_c^n = .false.$
// Initialize the optimal and quantized values of time steps $\tau_c^n$ and $\bar{\tau}_c^n$
For all cells: initialize $\tau_c^n$, initialize $\bar{\tau}_c^n$
// Initialize the initial values of conservative variables on the local intermediate time layer $\mathbf{U}_c^{0+1/2}$
For all cells: initialize $\mathbf{U}_c^{0+1/2}$
// Start of cycle by time
do
    // Determine the next active step
    Define $\bar{\tau}^n$
    // Predictor for active cells – define conservative variables in the intermediate time layer
    For all Active cells: define $\mathbf{U}_c^{n+1/2}$
    // Updating the list of active cells
    For all cells with $\bar{\tau}_c^n = \bar{\tau}^n$ define $Active_c^n = .true.$
    // Extrapolation for active cells: interpolation of conservative variables on the intermediate layer in neighboring cells $\tilde{\mathbf{U}}_c^{n+1/2}$ and calculation of fluxes on the faces and with asynchronous correction $\mathbf{F}_{c-}^{n+1}$ and $\mathbf{F}_{c+}^{n+1}$
    For all Active cells: interpolate neighbour cells $\tilde{\mathbf{U}}_c^{n+1/2}$
    For all Active cells: define face fluxes with asynchronous correction $\mathbf{F}_{c-}^{n+1}$ and $\mathbf{F}_{c+}^{n+1}$
    // Corrector for active cells – define conservative variables on a new layer $\mathbf{U}_c^{n+1}$
    For all Active cells: define $\mathbf{U}_c^{n+1}$
    // Check the current time
    If $t^n \geq t_{END}$ then exit
    Set $t^{n+1} = t^n + \bar{\tau}^n$, set $n = n + 1$
    // Determining new optimal and quantized values of time steps $\tau_c^n$ and $\bar{\tau}_c^n$
    For all Active cells: define $\tau_c^n$, define $\bar{\tau}_c^n$
end do
```

and 16 million, using from 1 to 64 cores of an AMD EPYC 7742 CPU. Figure 2 shows MPI efficiency parameters for the solver with the suggested modification of the asynchronous step: $Speedup = T_1/T_{Ncpu}$, which correspond to the acceleration compared to calculations on a single CPU core. When using the maximum number of cores, the efficiency of parallelization decreases due to an increase in the share of losses for synchronizing the cores with each other. However, when the grid size increases (weak stability test) the developed code provides good scalability with increasing the number of MPI processes. In this case, the acceleration of calculations with increasing grid sizes approaches maximum theoretical values.

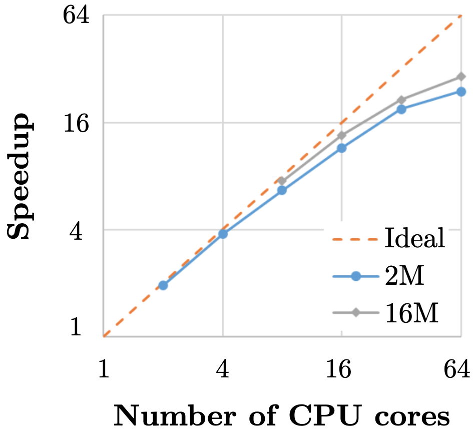


**Figure 2.** MPI CPU parallelization speedup for 2M and 16M grids

The efficiency of the modified asynchronous algorithm and its parallelization using graphics accelerators was assessed in test calculations of the jet flow from a test nozzle on a non-uniform grid of 16 million cells. For reliable modeling of noise generation by an outflowing jet, the mesh was significantly refined in the area of the nozzle edge (see  Section 5). The ratio of the linear size of the maximum cell to the size of the minimum cell was 20000. The use of the asynchronous step ensured acceleration of asynchronous calculations by 8–12 times compared to the global time step, both when using the CPU and when calculating on the GPU. Notably, the modified asynchronous step reduces the required memory by 25% and increases performance by approximately 1.2 times. In turn, parallelization using the CUDA architecture on the NVIDIA A100 graphics card provides a 600 times acceleration of calculations compared to calculations on a single CPU core. In total, this provides almost a 5000 times acceleration of calculations compared to a single CPU core, allowing to calculate 300 convective times of jet development per day on a grid of 16 million cells on a single GPU card.

Figure 3 compares calculation times for 500 convective times using CPU and GPU, and the achieved acceleration of the modified asynchronous mode compared to a single CPU in a global mode.

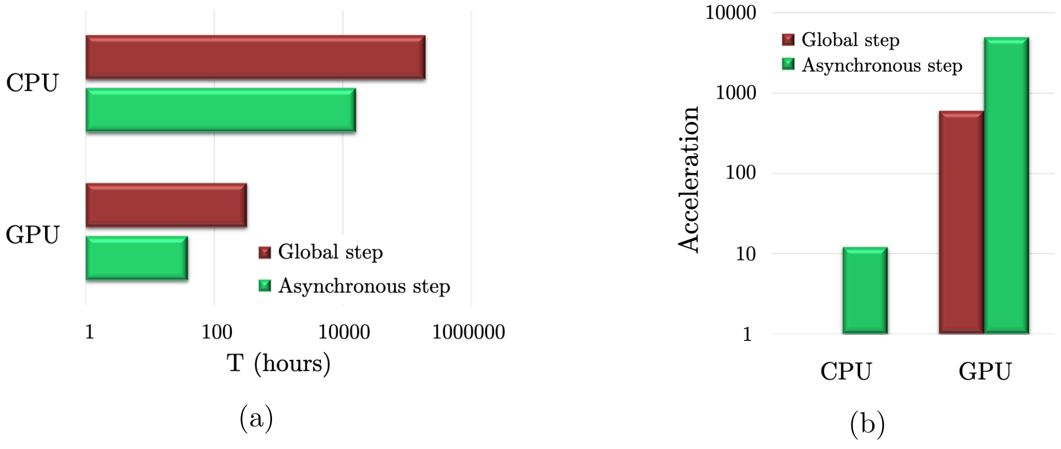
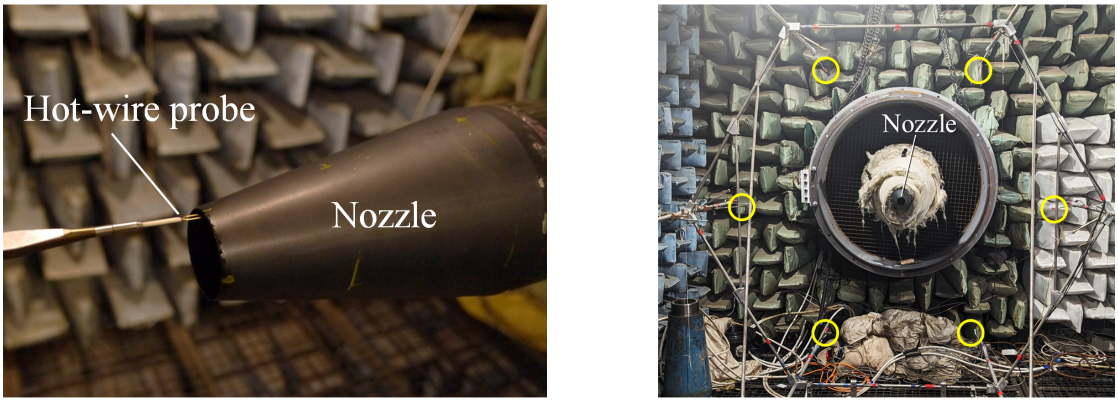


**Figure 3.** Efficiency of the modified asynchronous algorithm on a heterogeneous grid of 16 million cells for a CPU (AMD EPYC 7742, 1 core) and GPU (NVIDIA A100): calculation time (a) and calculation acceleration (b) on a logarithmic scale

## 5. Numerical Results and Validation

To verify the accuracy of the developed CABARET method, a single-stream axisymmetric static jet was simulated and its results were compared with experimental data. Experiments were carried out in the TsAGI anechoic chamber AC–2 designed for acoustic and aerodynamic measurements of isolated subsonic and supersonic jets. A round profiled nozzle of diameter $D = 40$ mm was used. An unheated jet with acoustic jet Mach number $M_j = 0.7$ was considered (jet exit velocity $U_j \approx 240$ m/s). The two types of experimental data were gathered. First, jet flow parameters were measured by a hot wire. Second, far-field acoustic pressure was measured by an azimuthal microphone array. The detailed description of the corresponding measurement procedures in AC–2 can be found in [5, 6, 16].

For the jet velocity measurements, Dantec probe 55R01 was used. The sensor of the probe had a fiber of diameter 70 $\mu$m and length of 1.25 mm. The probe was mounted on a Dantec 3D hot-wire traverse mechanism (model 41T33) so that the sensor was perpendicular to the mean flow (Fig. 4a). The traverse system was used to obtain the distribution of the mean velocity and velocity fluctuation spectra along the jet axis and along the lipline. The measurements on each of the lines were conducted in 31 points uniformly distributed along the jet axis from $x/D = 0$ to $x/D = 15$ with $0.5D$ increment.



(a) (b)

**Figure 4.** Experimental setup: (a) hot-wire measurements; (b) far-field noise measurements by the azimuthal array (microphones are highlighted by the circles)

Far-field measurements were carried out by the standard microphone array of radius $R = 0.8$ m typically used in AC–2 for jet noise measurements. The array is equipped by six $\frac{1}{2}$ Bruel&Kjaer microphones (type 4189) with B&K preamplifiers (type 2669, frequency range 40–25600 Hz, sensitivity 50 mV/Pa). The microphones are uniformly distributed over the azimuthal angle $\theta$ around the jet axis [6, 16] (Fig. 4b). The microphone array can be moved along its axis in the range $-31.25 \leq x/D \leq 62.5$, the origin $x = 0$ corresponding to the nozzle exit plane. The array is capable of measuring the first three azimuthal modes of jet noise at each jet cross-section $x$ and giving an assessment of the fourth cosine mode. Thus, if acoustic pressure fluctuations $p(x, R, \theta, t)$ can be represented with acceptable accuracy by a superposition of the first three azimuthal modes

$$p(x, R, \theta, t) \approx A_0(x, R, t) + \sum_{n=1}^{2} (A_n(x, R, t) \cos n\theta + B_n(x, R, t) \sin n\theta), \tag{5}$$

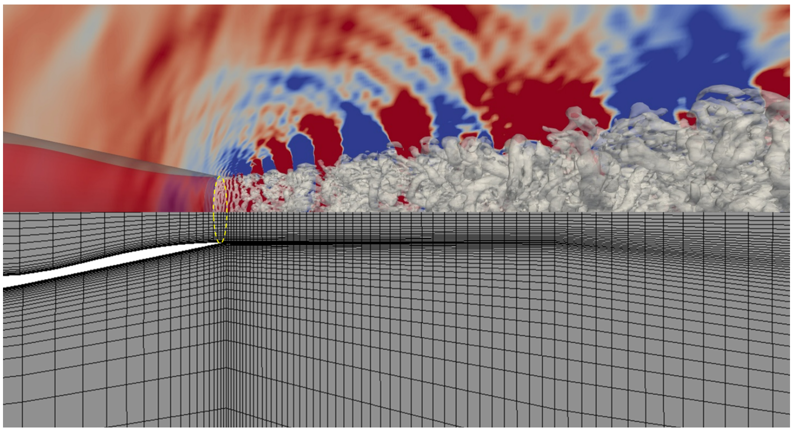


**Figure 5.** Mesh structure (every $4^{\text{th}}$ node is shown) and instantaneous pressure field with Q-criteria iso-surfaces (Q = 25000) in the jet symmetry plane

then the amplitudes $A_n(x, R, t)$ and $B_n(x, R, t)$ can be measured by the array. It is well-known [6, 12, 16] that for a subsonic jet issuing from a round nozzle $\overline{A_n^2} = \overline{B_n^2}$ $(n \geq 1)$, where the overbars stand for the time averaging, due to the axial symmetry of the statistical characteristics of the jet noise. Moreover, each azimuthal mode $n$ has specific directivity, and higher order modes $n \geq 3$ are negligibly small for low and moderate frequencies. Thus, the azimuthal content of jet noise serves as a good basis for validation of numerical simulation results [8, 15].

For the numerical simulation with turbulence modelling, the Wall Modelled Large Eddy Simulation framework was utilised, where the wall shear stress is computed from an iterative procedure using an algebraic wall function of the velocity profile and the streamwise velocity solution computed in the first-off-the-wall cell face [18]. For the upstream nozzle inlet boundary, the standard total pressure and total temperature boundary conditions are imposed. For the open lateral and outlet boundaries, non-reflecting characteristic boundary conditions are used. A cylindrical-conical computational solution domain is considered, which is aligned with the evolving jet flow downstream of the nozzle exit. The domain size in the streamwise flow direction is $227D$, where $32.5D$ corresponds to the cylindrical domain part upstream of the nozzle exit. The radius of the computational domain upstream of the nozzle exit is $50D$, linearly expanding to $125D$ in the outlet section.

A zoom in to the mesh structure (every $4^{\text{th}}$ node) and a snapshot of the instantaneous pressure field with Q-criteria iso-surfaces (Q = 25000) in the jet symmetry plane in the near nozzle region are shown in Fig. 5. Near the nozzle exit (at $x/D \sim 0.5$), somewhat delayed numerical transition to turbulent shear layer can be observed: quasi-regular vortical structures transform to stochastic ones. This is typical effect for LES solutions [7, 26], which correspond to the initially laminar inflow condition, like in the present case. In the future, it can be improved by further refinement of the initial shear region as well as using a synthetic turbulent inflow condition upstream of the nozzle exit. Pressure fluctuation pattern reveals near-field wave-packet-like structures, acoustic waves generated by the shear layer and spurious high-frequency noise related to the mesh stretching influence on the CABARET solution. Spurious noise can be reduced by utilizing a better quality Cartesian mesh in the shear region based on octree-type grid generation strategy [25]. Nevertheless, since the main goal of the current paper is to demonstrate adequate operation of the developed parallel code, mesh sensitivity study analogous to [25] was not performed.

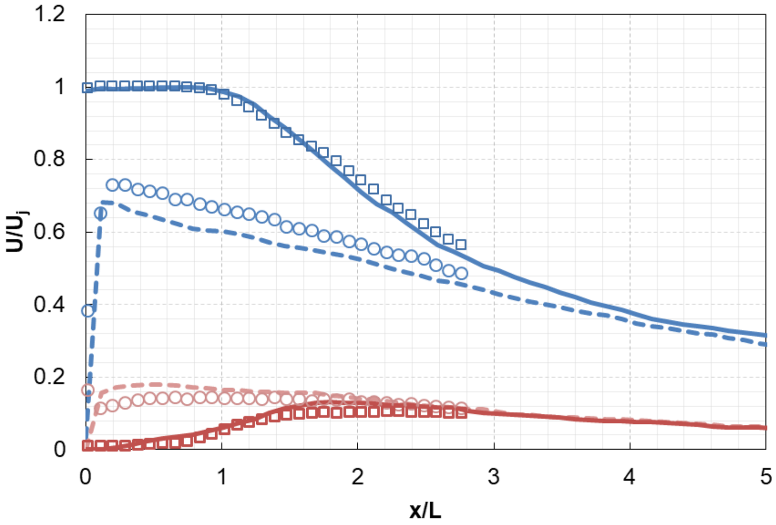
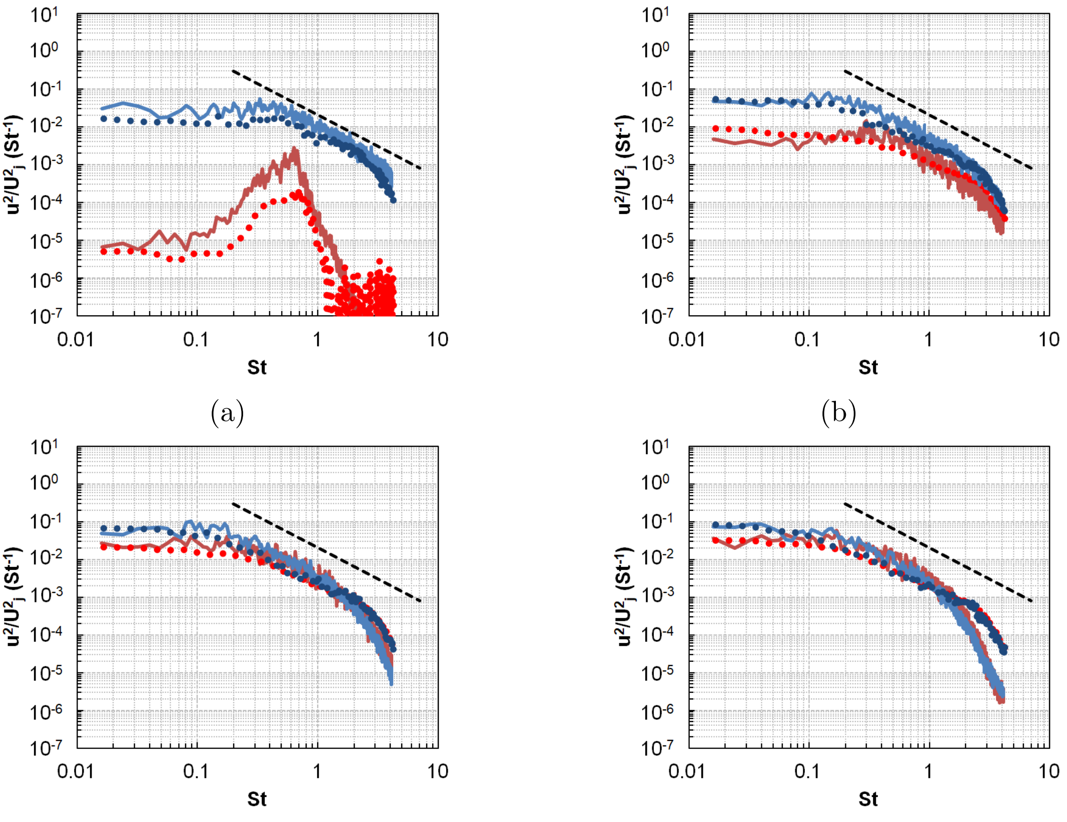


**Figure 6.** Mean (blue) and RMS (red) of the axial velocity fluctuations at the jet axis and at the lipline. Symbols – experiment (□ – axis, ○ – lipline), lines – simulation (solid – axis, dashed – lipline)

The main characteristics demonstrating general quality of physical modeling are related to the velocity field in the jet plume. Mean and RMS velocity distributions give general information on the jet development, while power density spectra of the velocity fluctuations indicate what turbulent scales are resolved in the simulation. To compare the results of the simulation with the experimental data, axial velocity component time histories were stored at the jet axis and the lipline during ~1500 convective time units $D/U_j$. The results of the comparison are shown in Fig. 6. Velocity is scaled by the jet exit velocity $U_j$ and axial coordinate is scaled by the length of the jet potential core $L$. Absolute values of $L$ was $4.5D$ and $5.5D$ for the simulation and experiment, respectively, so that the simulated jet is a bit shorter compared to the physical one, which is typical for relatively coarse meshes [7, 25]. In dimensionless coordinates ($x/L$, $U/U_j$), the results of the new parallel code are in good agreement with the experimental data.

A more subtle characteristic of the turbulent jet flow is its velocity spectra. It is known that, for a well developed mixing layer, high-frequency spectral tail should roll-off in accordance with the $-5/3$ slope of the Kolmogorov law for isotropic turbulence. Typical velocity spectra for several distances from the nozzle exit are shown in Fig. 7. Spectral density is scaled by the jet velocity $U_j$, and Strouhal number $St = fD/U_j$ represents dimensionless frequency. One can see that, for the points located in the turbulent region (all points at the jet lipline and axial points beyond the potential core of the jet), simulated and measured spectra are in good agreement with a slight overprediction in the numerical solution. A distinct $-5/3$ slope region can be observed with the highest resolved frequencies up to $St = 1...3$ depending on the point location. The resolution of the fine-scale fluctuations degrades for larger distances from the nozzle exit, because of the grid coarsening in the downstream direction. It is interesting to note that for a point located on the jet axis inside the potential core (Fig. 7a), the solution captures rather weak fluctuations related to the instability waves [5] (spectral hump around $St = 0.1...1$). Visible overestimation of the numerical solution in this point is partially related to the fact that the measured signal was masked by the self-noise of the probe and thus the resultant levels of the denoised experimental spectrum may be not precise enough.



(a)

(b)

(c)

(d)

**Figure 7.** Centerline (red) and lipline (blue) velocity spectra. Symbols – experiment, lines – simulation. (a) $x/L = 0.5$, (b) $x/L = 1.1$, (c) $x/L = 1.6$, (d) $x/L = 2.0$. Black dashed lines correspond to the $-5/3$ slope

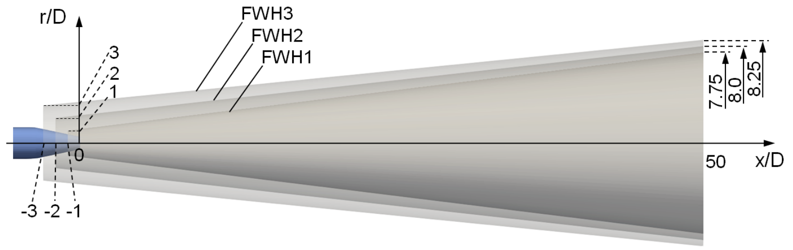


**Figure 8.** FWH surfaces for far-field noise calculation

For the far-field noise calculation, the Ffowcs Williams–Hawkings (FWH) integral method [9] in the permeable surface formulation given in [22] was used. It is known that closed FWH-surfaces with multiple closing discs [7, 26] may be preferable to improve low-frequency noise components simulation. However, for the purpose of the current study, we used a simplified approach with open control FWH-surfaces of an increased length to capture the main portion of jet noise without the disc-averaging technique. This allowed us to obtain adequate spectra in the frequency range of interest. Three conical FWH-surfaces of different radii were considered (Fig. 8).

The far-field observer points were distributed in axial ($-62.5 \leq x/D \leq 62.5$) and azimuthal directions over the cylindrical surface, of radius $R = 20D$, surrounding the jet. This allowed us

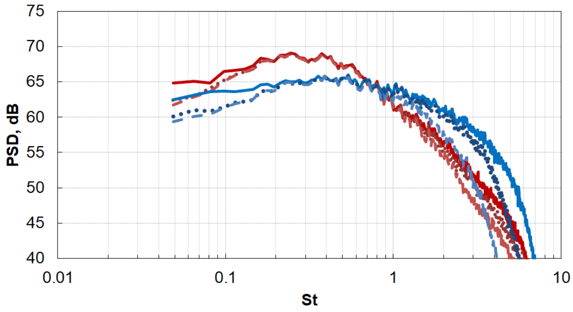


**Figure 9.** Sensitivity of the far-field spectra to the FWH surfaces.
Red – observation angle 32° ($x/D \approx 38$), blue – observation angle 90° ($x/D \approx 6$).
Solid lines – FWH1, dotted lines – FWH2, dashed lines – FWH3

to compare the simulation results with the experimental data obtained by the azimuthal array (Fig. 4b). Sensitivity of the results to the location of FWH-surface is shown in Fig. 9 for the sideline direction and for the direction of most intense radiation (observation angle is counted from the end of the potential core). It is seen that using FWH-surface of larger radius leads to reducing the resolution at high frequencies, but the spectral peak is reproduced identically by all FWH-surfaces. Further results are given for the acoustic control surface denoted as FWH1.

Figures 10 and 11 demonstrate the comparison of the simulation results with the experimental data, both in terms of the total noise and its azimuthal components. From Fig. 10, one can see that the simulation adequately reproduces the total jet noise spectra and the subtle structure of its modal content. There is some overprediction of absolute levels in the sideline direction, which can be improved by increasing the grid resolution of the initial shear layers, as mentioned previously.

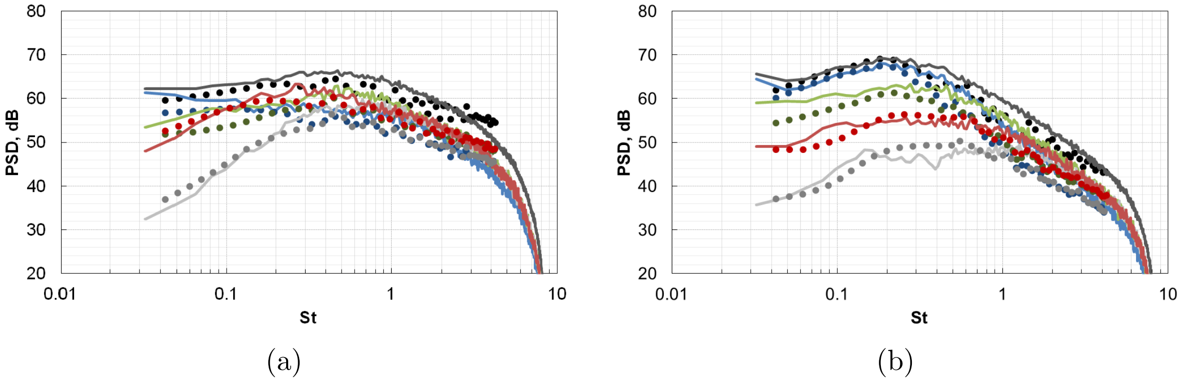
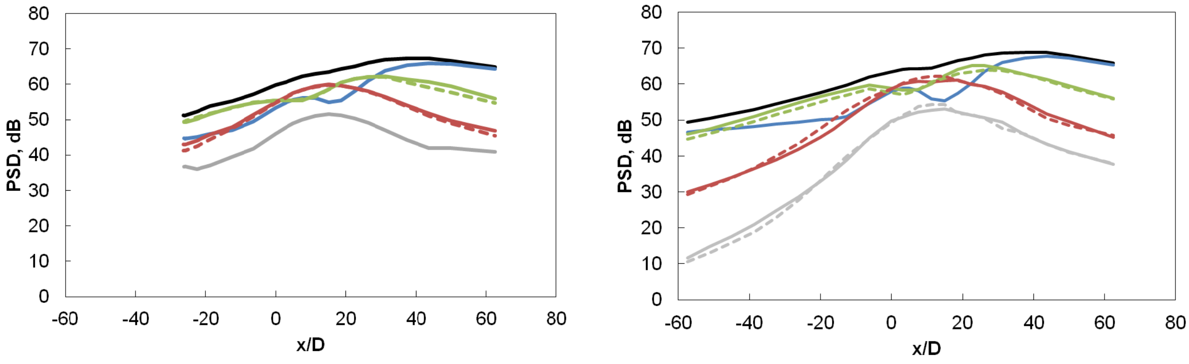


**Figure 10.** Far-field jet noise total and modal spectra. (a) sideline direction ($x/D \approx 6$), (b) direction of maximum radiation ($x/D \approx 38$). Symbols – experiment, lines – simulation. Black – total noise, blue – axisymmetric mode $n = 0$, green – mode $n = 1$, red – mode $n = 2$, grey – mode $n = 3$

Figure 11 presents the distribution of the total jet noise and its azimuthal modes on the cylindrical surface surrounding the jet for the spectral peak $St = 0.2$. Again, accurate simulation of the main features of jet noise is observed: the total directivity, the directivities of individual harmonics, the equivalence of the corresponding sine and cosine modes, and the domination of low-order modes $n = 0, 1, 2$ are all correctly captured. Thus, we have demonstrated acceptable accuracy of the developed parallel solver on the acoustically sensitive calculations. Even on the rather coarse mesh, encouraging simulation quality, capturing the main physical features of real turbulent jet, was obtained, both in terms of turbulent properties of the jet flow and its acoustic radiation, including rather subtle characteristics of individual azimuthal modes.



(a) (b)

**Figure 11.** Far-field distribution of jet noise and its azimuthal modes on the cylindrical surface surrounding the jet for the peak frequency $St = 0.2$. (a) experiment, (b) simulation. Black – total noise, blue – axisymmetric mode $n = 0$, green – mode $n = 1$, red – mode $n = 2$, grey – mode $n = 3$. Solid lines – cosine modes, dashed lines – sine modes

## Conclusion

A new asynchronous modification of the CABARET method is proposed for the solution of Navier–Stokes equations in the Large Eddy Simulation regime. A new asynchronous characteristic-extrapolation step is implemented increasing computational efficiency of the previously suggested asynchronous algorithm. It is shown that GPU parallelization with an asynchronous time step is an extremely effective tool for increasing the computational performance of the Navier–Stokes CABARET solver for performing acoustically sensitive calculations. The computational efficiency and encouraging accuracy of the developed parallel solver was demonstrated on the acoustics sensitive simulation of a single-stream Mach 0.7 round turbulent jet for a moderate grid resolution. Modified asynchronous CABARET algorithm provides an almost 5000 times acceleration of calculations compared to a single CPU core, thereby enabling the calculation of 300 convective times of jet development on a grid of 16 million cells per day. Comparison with the TsAGI experimental data is performed the mean velocity, turbulence intensity, and turbulent velocity spectra. It is shown that the structure of the jet flow is reproduced correctly, including reasonable resolution of fine-scale turbulent fluctuations in the inertial subrange of the energy spectra and capturing low-amplitude instability waves footprint in the jet potential core. Far-field noise predictions are performed using the permeable surface Ffowcs Williams–Hawking formulation. The conventional aeroacoustic postprocessing is supplemented by the azimuthal decomposition of the far-field noise. It is shown that the method correctly reproduces the nontrivial spectra and directivities of individual far-field acoustic modes. Thus, the developed parallel GPU CABARET solver is demonstrated to correctly capture the main physical features of a natural turbulent jet even on rather coarse meshes. Future work will include increasing the jet LES grid resolution utilising the significant acceleration offered by the developed in-house GPU CABARET solver.

## Acknowledgements

The part of the work related to the development of the numerical method was supported by Russian Science Foundation (grant No. 21–71–30016). Experimental part of the work was supported by Russian Science Foundation (grant No. 19–71–10064).

## References

1. Abid, H.A., Markesteijn, A.P., Karabasov, S.A.: Trailing Edge Noise Modelling of Flow over NACA Airfoils Informed by LES. AIAA 2021-2233 (2021). `https://doi.org/10.2514/6.2021-2233`

2. Advisory Council for Aviation Research and Innovation in Europe (ACARE): Realising Europes vision for aviation, in: Strategic research and innovation agenda, Vol. 1, Advisory Council for Aviation Research and Innovation in Europe, 2012; Strategic research and innovation agenda: the proposed European partnership for clean aviation., `http://www.clean-aviation.eu/`, July 2020

3. Block, P.J.W., Gentry, G.L.J.: Directivity and trends of noise generated by a propeller in a wake. NASA Technical Paper 2609 (1986)

4. Brelje, B.J., Martins, J.R.R.A.: Electric, hybrid, and turboelectric fixed-wing aircraft: a review of concepts, models, and design approaches. Prog. Aerosp. Sci. 104, 1–19 (2019), `https://doi.org/10.1016/j.paerosci.2018.06.004`

5. Bychkov, O., Faranosov, G., Kopiev, V., *et al.*: Jet installation noise modeling in static and flight conditions using centerline fluctuations. AIAA Journal 60(6), 3620–3634 (2022). `https://doi.org/10.2514/1.J060879`

6. Faranosov, G., Belyaev, I., Kopiev, V., Bychkov, O.: Azimuthal structure of low-frequency noise of installed jet. AIAA Journal 57(5), 1885–1898 (2019). `https://doi.org/10.2514/1.J057476`

7. Faranosov, G.A., Goloviznin, V.M., Karabasov, S.A., *et al.*: CABARET method on unstructured hexahedral grids for jet noise computation. Computers and Fluids 88, 165–179 (2013). `https://doi.org/10.1016/j.compfluid.2013.08.011`

8. Faranosov, G.A., Kopiev, V.F., Karabasov, S.A.: Application of Azimuthal Decomposition Technique for Validation of CAA Methods. AIAA-2013-2238, 19th AIAA/CEAS Aeroacoustics Conference (2013). `https://doi.org/10.2514/6.2013-2238`

9. Ffowcs Williams, J.E., Hawkings, D.L.: Sound generation by turbulence and surfaces in arbitrary motion. Phil. Trans. R. Soc. A 264, 321–343 (1969). `https://doi.org/10.1098/rsta.1969.0031`

10. Goloviznin, V.M., Samarskii, A.A.: Finite difference approximation of convective transport equation with space splitting time derivative. Jour Matem. Mod. 10(1), 86–100 (1998)

11. Gryazev, V., Kalyan, A., Markesteijn, A., Karabasov, S.: Broad band shock associated noise modelling for high-area-ratio under-expanded jets. Nature Research Journal of The Acoustical Society of America 150(2), 1534–1547 (2021). https://doi.org/10.1121/10.0005976

12. Juve, D., Sunyach, M., Comte-Bellot, G.: Filtered azimuthal correlations in the acoustic far field of a subsonic jet. AIAA Journal 17(1), 112–114 (1979). https://doi.org/10.2514/3.61076

13. Karabasov, S.A., Goloviznin, V.M.: Compact accurately boundary-adjusting high-resolution technique for fluid dynamics. Journal of Computational Physics 228(19), 7426–7451 (2009). https://doi.org/10.1016/j.jcp.2009.06.037

14. Kim, H.D.: Distributed propulsion vehicles. 27th Congress of the International Council of the Aeronautical Sciences, Nice, France, September 19-24, 2010 (2010)

15. Kopiev, V.F., Shur, M.L.: Azimuthal components of turbulent jet sound field: measurement results and their implementation for validation of modern noise computation techniques. TsAGI Science Journal 41(1), 3–13 (2010)

16. Kopiev, V.F., Zaitsev, M.Y., Chernyshev, S.A., Kotova, A.N.: The role of large-scale vortex in a turbulent jet noise. AIAA paper 99-1839 (1999). https://doi.org/10.2514/6.1999-1839

17. Magliozzi, B., Hanson, D.B., Amiet, R.K.: Propeller and propfan noise. NASA Langley Research Center, Aeroacoustics of Flight Vehicles: Theory and Practice, Vol. 1: Noise Sources pp. 1–64 (1991)

18. Markesteijn, A.P., Gryazev, V., Karabasov, S.A., *et al.*: Flow and noise predictions of coaxial jets. American Institute of Aeronautics and Astronautics AIAA Journal: Devoted to Aerospace Research and Development 58(12) (2020). https://doi.org/10.2514/1.J058881

19. Markesteijn, A.P., Jawahar, H.K., Karabasov, S.A., Azarpeyvand, M.: GPU CABARET Solutions for 30P30N Three-element High-lift Airfoil with Slat Modifications. AIAA 2021-2115, Airframe/High-Lift Noise I (2021)

20. Markesteijn, A.P., Karabasov, S.A.: CABARET solutions on graphics processing units for NASA jets: Grid sensitivity and unsteady inflow condition effect. Comptes Rendus – Mecanique 346(10), 948–963 (2018). https://doi.org/10.1016/j.crme.2018.07.004

21. Markesteijn, A.P., Karabasov, S.A.: Simulations of co-axial jet flows on graphics processing units: the flow and noise analysis. Philosophical Transactions of The Royal Society A: Mathematical, Physical and Engineering Sciences 377(2159) (2019). https://doi.org/10.1098/rsta.2019.0083

22. Najafi-Yazdi, A., Bres, G.A., Mongeau, L.: An acoustic analogy formulation for moving sources in uniformly moving media. Proc. Royal Soc. A 467(2125), 144–165 (2011). https://doi.org/10.1098/rspa.2010.0172

23. National Academies of Sciences Engineering and Medicine: Commercial Aircraft Propulsion and Energy Systems Research: Reducing Global Carbon Emissions. The National Academies Press, Washington, DC (2016). https://doi.org/10.17226/23490

24. Sahoo, S., Zhao, X., Kyprianidis, K.: A review of concepts, benefits, and challenges for future electrical propulsion-based aircraft. Aerospace 7(44) (2020), https://doi.org/10.3390/aerospace7040044

25. Semiletov, V.A., Karabasov, S.A.: CABARET scheme with conservation-flux asynchronous time-stepping for nonlinear aeroacoustics problems. Journal of Computational Physics 253, 157–165 (2013). https://doi.org/10.1016/j.jcp.2013.07.008

26. Shur, M.L., Spalart, P.R., Strelets, M.K.: Noise Prediction for Increasingly Complex Jets. Part I: Methods and Tests. Part II: Applications. Int. J. Aeroacoustics 4(3+4), 213–266 (2005)

27. Solntsev, I.A., Chintagunta, A., Markesteijn, A.P., Karabasov, S.A.: CABARET on rotating meshes. Applied Mathematics and Computation 446(127871) (2023). https://doi.org/10.1016/j.amc.2023.127871

**DOI: 10.14529/jsfi240204**

# Development of Compressible Mixing Layer Instability Simulated Using the Direct Simulation Monte Carlo Method

***Alexandr V. Kashkovsky*[1], *Alexey N. Kudryavtsev*[1], *Anton A. Shershnev*[1]**

The Kelvin–Helmholtz instability developing in the mixing layer between two supersonic streams is simulated with the Direct Simulation Monte Carlo (DSMC) method using the SMILE-GPU software. No initial perturbations are introduced into the flow so that the disturbances are excited by and develop from the statistical fluctuations inherent in the DSMC method because of its stochastic nature. Multiple graphics processing units (GPUs) are employed for numerical simulations and efficient parallelization strategies for DSMC implementation on GPU clusters are presented. Between 0.4 and 1.6 billion of test particles are used to reproduce the development of the flow instability. The influence of the number of particles on mean flow properties and pulsation characteristics is investigated and discussed. It is shown that the pulsation characteristics are substantially affected by the number of particles because of a delay in the instability onset and vortex formation at a lower level of statistical fluctuations. A new algorithm for identification and analysis of vortex motion in noisy flow data is considered and applied to flowfields resulted from the unsteady DSMC simulations.

*Keywords: rarefied gas flows, free shear flow instabilities, particle-based methods for kinetic equations, parallelization strategies, GPGPU computations with CUDA.*

## Introduction

The Direct Simulation Monte Carlo (DSMC) method proposed by G.A. Bird in 1963 is a particle-based method for solving the Boltzmann transport equation using probabilistic Monte Carlo techniques [1]. The gas in the DSMC method is modeled by a large number of test particles, the motion of which, at each time step, is divided into two stages: moving by inertia and collisions with other particles, selected in a stochastic way from the same grid cell.

For many years, the DSMC method has been intensively used to simulate rarefied gas flows, in particular, in applications to high-altitude aerodynamics and microfluid dynamics. In general, the DSMC method is applicable not only to rarefied flows, but also to those in the near-continuum and fully continuum regimes, though such simulations are possible only with a huge number of test particles. Nevertheless, the flows with Knudsen numbers $\mathrm{Kn} = \ell/L \sim 10^{-4} \div 10^{-5}$, where $\ell$ is the molecular mean free path and $L$ is the macroscopic flow characteristic length, can be routinely performed with modern day computers.

The DSMC method is most often used to simulate steady flows because, in this case, time averaging can be efficiently employed to decrease statistical fluctuations which are an inherent feature of this approach. In recent years, however, there has been a significant increase in simulating unsteady fluid dynamic phenomena, such as hydrodynamic instabilities, using this numerical approach. So, the Richtmeyer–Meshkov and Rayleigh–Taylor instabilities caused by inertial and gravitational body forces, respectively, were simulated in [2, 3] while the first results of DSMC simulations of a shear flow instability were presented in [4] where the Kelvin–Helmholtz (KH) instability developing in a compressible mixing layer between two parallel supersonic streams was reproduced and in [5] where an unstable plane supersonic jet was considered.

Moreover, the DSMC method was even used for simulating fully developed turbulence. In [6] the Taylor–Green vortex flow was simulated and the famous Kolmogorov $-5/3$ spectrum law

[1] Khristianovich Institute of Theoretical and Applied Mechanics, Novosibirsk, Russian Federation

was reproduced. Later, DSMC simulations of the Couette flow between two plane walls were performed and it was shown that the law of the wall and other features of wall-bounded turbulence can also be successfully reproduced using this approach [7]. Further, when simulating near-continuum turbulent flows [8, 9], it was revealed that the continuum approach is not able to describe correctly the dissipation range of turbulence spectrum because it does not take into account thermal fluctuations [9] while they are properly represented in DSMC simulations.

In [2, 3] the authors tried to reduce the statistical fluctuations by using in 2D simulations of flow instabilities up to $10^{11}$ test particles, which required tens hours of computer time on $1.57 \times 10^6$ processors of Sequoia, an IBM Blue Gene/Q supercomputer. Our approach in [4, 5] was different: we used a large but reasonable number of particles so that the computations could be performed on a conventional hybrid CPU/GPU cluster available in many laboratories.

It means that the level of statistical noise in the computed flowfields was rather noticeable though it could be reduced by averaging over time intervals short in comparison with the hydrodynamic characteristic time scales. In some aspects, the statistical fluctuations inherent in the DSMC method can be even useful because their presence allows us to initiate the development of hydrodynamic instabilities in a “natural” way, without exciting the flow with any artificial initial disturbances. However, the influence of the statistical noise with an amplitude much higher than the level of real-world thermal fluctuations on the results of numerical simulations is rather a delicate question. Also, there is a problem of analyzing the noisy computational flowfields and extracting from them the flow instability characteristics such as the frequencies of the most perceived and the most amplified disturbances, their growth rates, as well as identifying the locations, motion and merging of the vortices developing in the unstable shear flow. Both these subjects are addressed in the current paper.

The rest of the paper is organized as follows. In Section 1 a brief description of the DSMC method and the SMILE-GPU numerical code are given along with the specifics of the GPU-based computations and other features of the numerical algorithm, Section 2 contains the problem formulation and computation results. Section 3 deals with a special procedure for detailed analysis of the obtained numerical results. Brief summary is given in the Conclusion section.

## 1. Code Implementation of Parallel Computations in DSMC Method

As was mentioned in Introduction, the DSMC method is a particle-based kinetic approach to numerical simulation of gas flows, in which model particles are moving and colliding with each other in a manner quite similar to real gas molecules. This method allows one to obtain a solution satisfying the Boltzmann kinetic equation. Kinetic methods are usually used to study the rarefied flows when the continuum approach based on the Euler or Navier–Stokes equations is inapplicable. The process of numerical simulation is split into consecutive time steps, which, in turn, consist of two main stages: free-molecular movement of particles and binary collisions. The computational domain is divided into cells with sizes comparable to a local mean free path. In the simulated collision process randomly selected particles located in the same cell collide with one another. This algorithm requires the so-called particle index that indicates which cell the particle is located in. This index list is updated on each computational step. Also, on each step the particles and their velocities are sampled in cells and this statistical data is used to calculate the mean number density, velocity, temperature and other macroscopic parameters in each cell.

The number of particles used in a DSMC simulation is limited only by the magenta CPU computational power. Usually, the number of model particles is fewer than the number of real molecules by a factor of $10^{10}$–$10^{20}$. Thus, the DSMC method is typically employed for the numerical simulation of steady flows, where statistical fluctuations can be reduced averaging the solution over a sufficiently large time interval. When simulating unsteady flows, the information is averaged over a relatively small number of time steps, usually not exceeding few hundreds. This leads to large statistical fluctuations and makes the data analysis quite difficult. The fluctuations can be reduced by increasing the number of model particles, but since the fluctuations magnitude decrease proportionally to the inverse square root of samples, the sampling size has to be increased by orders of magnitude. Thus, DSMC simulations of unsteady flows, in particular of shear flow instabilities, inevitably require a large number of test particles used and can be performed only with efficiently parallelized software.

The SMILE-GPU software used in the current study is a highly efficient numerical solver designed to run on graphical processor units (GPUs). The amount of memory available on a GPU is quite limited, so computations were carried out on several GPUs using OpenMP and MPI technologies combined with domain decomposition technique.

## 1.1. Computations on GPU

A GPU is a device containing many multicore processors originally designed for manipulations with 3D graphical images. However, over time, their computational performance increased to levels where they became comparable with conventional CPUs. On the basic level, a modern GPU consists of a large number multicore processors and a specialized scheduler managing their workloads. Other specific feature of GPUs are as follows.

- A single core of GPU is considerably slower than a single core of CPU. However, the total number of GPU cores is an order of magnitudes higher than on a CPU. So, in theory, GPUs could performs computations faster.
- All computations on GPU are performed in the so-called SIMD (Single Instruction Multiple Data) regime, when all cores execute the same instruction in the code applied to a different set of data on each core. As a result, each statement in "if-else" operator is executed regardless of condition match. However, results of non-matching branch execution are not used. This specific manner of computation require appropriate numerical algorithms.
- GPU cores can only access the memory of their GPU and cannot use CPU memory. All memory exchange in initiated via CPU and is very slow. So, the best scenario for computation is to upload all the data to GPU, perform all required computations and download results into CPU memory.

SMILE-GPU uses CUDA (Compute Unified Device Architecture) application programming interface (API), developed by Nvidia for their GPU devices. This interface allows one to use any GPU as some abstract device that has some number of computational threads grouped into blocks. Computations are performed in so-called kernels, specific function executing on all cores of a device. Built-in global variables allow each thread to identify its index number and the index of its block, and the indices of thread and block can then be used to assign specific portion of data to process. The total number of computational threads can be larger than the number of physical cores on the device and CUDA scheduler manages their execution order to provide maximum efficiency (at least, in theory). This allows a programmer to focus on the numerical algorithm and not on details of hardware architecture.

Considering these features, the approach of parallelization by data, when all threads can simultaneously perform the same operation on different data sets, seems to be optimal for programs running on GPUs. Main entities in the DSMC method are particles and cells, so at any given time all computational threads should process either particle or cell. However, because of the SIMD approach number of operations for each thread would be the same. As a result, if the size of data is different on each thread, the efficiency would drop, because there would be threads performing basically useless computations. This can happen, for example, when there is a different number of particles in cells, because of different gas density.

Unfortunately, it is hard to make a theoretical estimate for the algorithm efficiency when using GPU, because it depends not only on the data size, but also on the data structures in program implementation, and their location in memory. So, usually, the most computationally efficient implementation is a result of trial and error. For example, during the SMILE-GPU development, 7 variants of collision algorithms and 4 algorithm of sampling were tested.

In binary collisions any pair of molecules can engage in chemical reaction resulting in pair of particles of different species. Chemical reactions occur relatively infrequently, requiring additional computations and subsequent rebuild of particle-cell index. The straightforward implementation of the algorithm leads to a dramatic order of magnitude drop in efficiency. So, in SMILE-GPU chemical reactions are "delayed", meaning that in collision procedure the pair of reacting particles is flagged and added to a list of dissociating/exchanging particles, and reaction itself is realized in separate kernel afterwards. This approach allows to even out the computations on different threads and the particles index will not be needed until next iteration of collisions, so it is rebuilt only on the next time step. But, unfortunately, this trick is not free because the preliminary flagging of particles makes them unavailable for other reactions in this cell, so the time step needs to be reduced to match the number of collisions in cell. Tests show that when the number of collisions is 10–20% of total number of particles delayed reactions have negligible effect on the efficiency. However, for the processes of internal energy exchange it leads to an unacceptably small time step, so these process are realized during main collision procedure despite the certain amount of useless computation.

## 1.2. Workload Distribution between GPUs

Domain decomposition is used for computations on multiple GPUs. The domain is split into subdomains and each is assigned to a GPU. If some particle moves outside of the domain, it is transferred to the respective GPU during the exchange stage of the algorithm. This is a conventional approach to the parallelization of a DSMC code, but there are some differences stemming from the GPU architecture.

Because the CPU-GPU transfers are usually very slow, the domain decomposition should minimize inter-partition exchange. This is achieved using the wind reference frame, when the velocity vector is oriented along the $X$ axis. The subdomains are formed by grouping cells along the inflow velocity vector. So, the particles following straight streamlines will fly through the domain without crossing partition boundaries. In this case, only particles with relatively large transversal thermal velocity can move to neighboring partitions. In real problems of practical interest with complex geometries, shock waves, mixing and boundary layers this simple assumption about velocity direction does not always hold, however, it allows to considerably increase efficiency of the computation.

Another important aspect of the computation is the balanced workload on all computational devices. This problem applies to both CPU and GPU codes. It is virtually impossible to obtain a perfectly balanced data distribution in a DSMC run because of the statistical nature of the method and fluctuations in instant flowfields. This also leads to fluctuation in a single step execution time. A popular approach used in CPU DSMC codes is to distribute cells randomly among the threads to ensure that on each thread there are cells with both high and low computational loads. This allows for balanced computation with acceptable efficiency achieved via more extensive exchange. This technique cannot be used in GPU computations because the exchange is slower, and the wall time for each step is considerably lower, so any slow thread would drop total efficiency and performance drastically.

DSMC computations of steady flows usually start from uniform flowfield which evolves during the simulation to a resulting flowfield with some distribution of gasdynamic parameters. The dynamic balancing is required for this unsteady stage of computations. For unsteady flows balancing is iteratively carried out throughout the whole computation. The main idea behind the balancing process is similar to the greedy algorithm approach: transfer (i.e. reassign) cells from GPUs with high workload to GPUs with low workload. We call this algorithm Direct Timer Load Balance (DTLB). Each GPU has a correction factor $C_i$ which is proportional to $N_i$, the number of cells assigned to $i$th GPU. If $N_c$ is the total number of cells in the domain and $N_g$ is the total number of GPUs, then $N_i$ is calculated as

$$N_i = C_i N_c / N_g. \tag{1}$$

The coefficients $C_i$ are recalculated using measurement of timers $T_{c,i}$. At the start of computation $C_i = 1$ for all GPUs. We calculate the maximum $T_{c,max}$ of all $T_{c,i}$, and then the value of $C_i$ is updated according to formula

$$C_i^{(n)} = C_i^{(n-1)} \cdot \left(1 + \frac{T_{c,max} - T_{c,i}}{T_{c,max}}\right), \tag{2}$$

where $(n)$ is the recalculation iterator, $T_{c,max} - T_{c,i}$ is the $i$th GPU time of wait and should be equal to zero for a perfectly balanced distribution. As can be seen, $C_i$ for the GPU with maximum workload ($T_i \equiv T_{max}$) will remain unchanged, and for the rest of GPUs it will increase. Finally, the factors are renormalized

$$C_i = C_i^{(n)} / \sum_k^{N_g} C_k^{(n)}. \tag{3}$$

Tests show that after 10–20 iteration this approach yields approximately 95–99% efficiency.

## 2. Numerical Simulation of the KH Instability in Compressible Mixing Layer

### 2.1. Problem Formulation and Numerical Setup

It is well known that a mixing layer between two parallel streams is unstable even at low Reynolds numbers [11]. The linear instability of a compressible mixing layers was investigated in detail in [12–14]. The linear stability analysis shows that the instability characteristics of the compressible mixing layers are strongly affected by the so-called convective Mach number

$$\mathrm{M}_c = \frac{U_1 - U_2}{a_1 + a_2}, \tag{4}$$

where $U_1$ and $U_2$ are the velocities of two streams and $a_1$ and $a_2$ are the sound speeds in them. At subsonic $\mathrm{M}_c$ the KH instability dominates in the mixing layer while at $\mathrm{M}_c > 1$ two new, the so-called supersonic instability, modes emerge whose disturbances radiate from the mixing layer as acoustic waves. Moreover, at $\mathrm{M}_c \leq 0.6$, 2D disturbances are the most unstable and grow faster than the 3D ones.

The mixing layer is a canonical example of a highly-unstable free shear flow and its instability occurs even at low Reynolds numbers, so that the instability should manifest itself in rarefied flows. Nevertheless, it seems that until recently the mixing layer instability was simulated using a kinetic approach only in one work [15] where the simulation was performed by solving the model kinetic equations with a deterministic finite difference method. Lately, for this purpose, DSMC simulations were also used [4] as well as the simulations based on the model kinetic equations and the Boltzmann transport equation [16].

In the current study, we simulate a plane mixing layer between two streams of monatomic gas (argon, Ar) with the temperature $T_1 = T_2 = 300$ K and the pressure $p_1 = p_2 = 2.5$ Pa. These conditions correspond to the mean free path $\lambda_1 = \lambda_2 = 0.0029$ m. The flow velocities of two streams are equal to $U_1 = 805$ m/s and $U_2 = 483$ m/s so that their Mach numbers are $\mathrm{M} = 2.5$ and $\mathrm{M} = 1.5$, respectively. Thus, the convective Mach number $\mathrm{M}_c = 0.5$ and the most unstable mode is the 2D KH disturbances.

The numerical simulations are performed in a rectangular domain 40 × 16 meters. The two streams start mixing on the inflow boundary where model particles are introduced with the corresponding Maxwellian distributions. No artificial disturbances are imposed on the flow and instability emerges in a natural manner from statistical fluctuations inherent to the DSMC approach. However, the number of model particles is many order of magnitude smaller than the number of real molecules so that the amplitude of fluctuations is considerably larger than in nature. To investigate the effects of fluctuations on the instability emergence and development three simulations with $N_p = 0.4$, 0.8 and 1.6 billion ($10^9$, bln) of model particles, respectively, are carried out.

The computations are performed using a two-level grid. The first level grid consists of 1000 × 400 cells, while the second level grid used for particle collisions is constructed automatically according to local flow gradients to ensure there are 5 or more model particles in a collision cell. The time step is set to $\Delta t = 10^{-6}$ s. From 6 up to 12 Nvidia V100 GPUs are used in computations.

All obtained flow parameters are averaged over a relatively small interval equal to 100 time steps (or $10^{-4}$ s). During first 100,000 time steps no data are gathered because this period corresponds to a transient starting process. The averaged flowfields will be referred below as *frames*. The number of frames $n_f$ obtained in the three simulations are 2000, 2358 and 5254, respectively.

Reducing the averaging period will increase the fluctuations in flowfields, while its increasing will smooth out some of the features and disturbances in the flow. Moreover, from the generalization of the Kotelnikov [17] theorems for restoring a continuous multi-frequency signal from discrete samples, it follows that the maximum correctly restored frequency is 2 times higher than the polling frequency. For the case considered, the maximum frequency equals 5 kHz, but in fact most frequencies of interest in our study are lower than 1 kHz.

A typical frame of the transversal velocity flowfield $u_y(x, y)$ is shown in Fig. 1. It is seen that the flow is unstable and there are well known KH “billows” inside the mixing layer. The vortices emerge close to the left (inflow) boundary, grow as they travel downstream and merge into larger vortices. The statistical noise is also clearly visible in Fig. 1.

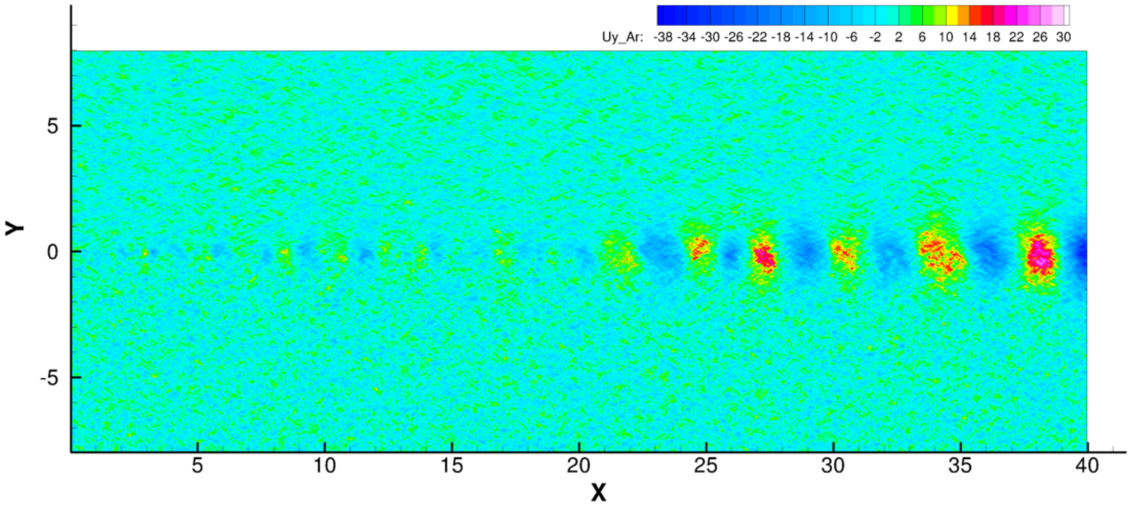


**Figure 1.** Flowfield of transversal velocity $v(x, y)$ averaged over 100 time steps for $N_p = 0.4$ bln

## 2.2. Mean Flowfields

The mean flowfield $Q(x, y)$ and the root mean square (RMS) pulsation flowfield $q'_{rms}(x, y)$ of an arbitrary flow quantity $q(x, y, t)$ are calculated by averaging over $n$ frames:

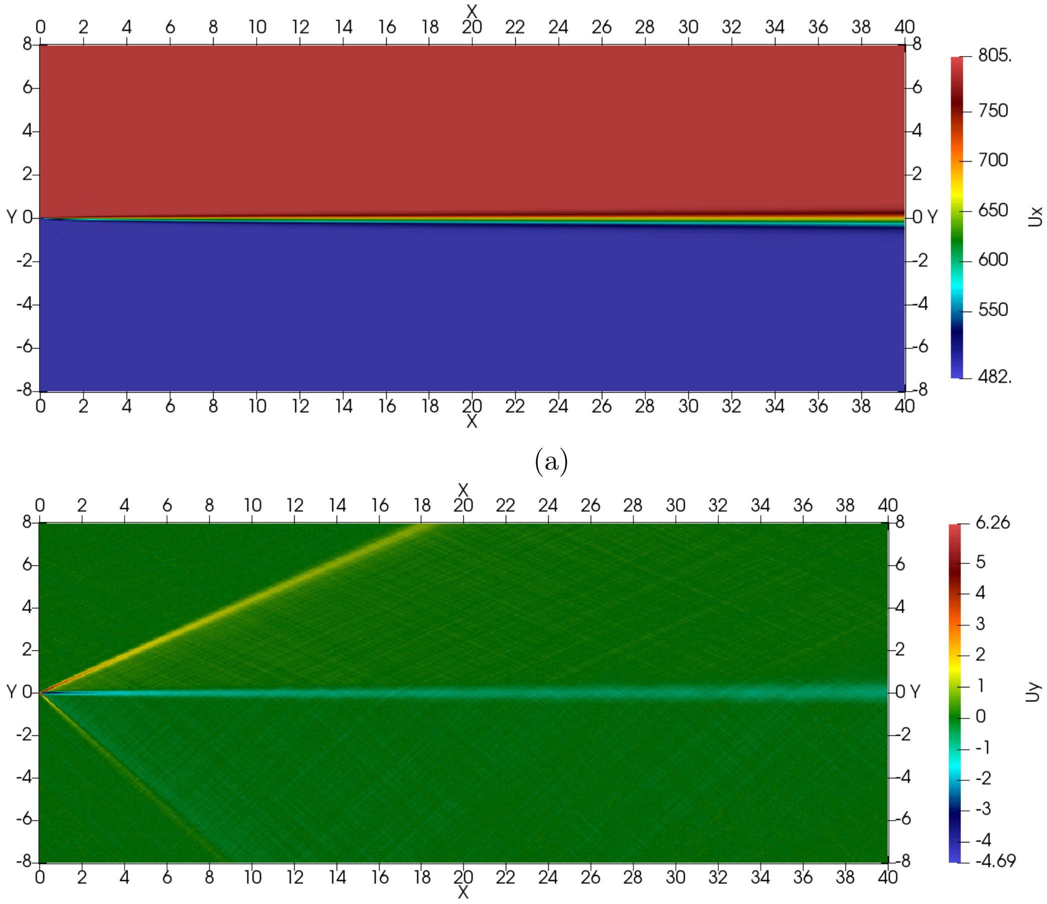
$$Q = \frac{\sum_{i=1}^{n} q_i}{n}, \qquad q'_{rms} = \sqrt{\frac{\sum_{i=1}^{n}(q_i - Q)^2}{n}}. \tag{5}$$

In Fig. 2 flowfielfds of mean longitudinal $U(x, y)$ and transversal $V(x, y)$ velocities are presented for $N_p = 0.4$ bln. It is seen that the mixing layer thickness gradually grows downstream. The weak shock waves (Mach waves) originating from the meeting point of two streams at the left boundary are clearly visible in the transversal velocity flowfield $V(x, y)$. The inclination angles of the weak shock waves in the upper and lower streams equal 23.6° and 41.8°, respectively. They match perfectly with the Mach angles $\alpha_M = \sin^{-1}(1/M)$ for these streams. The oblique intersecting straight lines behind the shock waves are acoustic characteristic lines of two families. They are visible due to weak velocity perturbations propagating from the mixing layer as well as from the upper and lower boundaries. The same oblique intersecting Mach lines can be seen in many experimental schlieren visualizations of supersonic flows.

The mean profiles of longitudinal velocity $U(y)$ close to the outflow boundary, at $x = 39$ m, are shown in Fig. 3. The profiles obtained in the simulations with different numbers of test particles are compared in Fig. 3a. It is evident that they coincide almost perfectly.

In Fig. 3b the computed velocity profile at the same cross-section $x = 39$ m is compared with the analytical expression

$$U(x, y) = U_c + \frac{\Delta U}{2}\,\mathrm{erf}\left[\frac{\sqrt{\pi}(y - y_c)}{\delta_\omega(x)}\right], \qquad U_c = \frac{U_1 + U_2}{2}, \qquad \Delta U = U_1 - U_2 \tag{6}$$



(a)

(b)

**Figure 2.** Mean longitudinal $U(x, y)$ (a) and transversal $V(x, y)$ (b) velocity flowfields for $N_p = 0.4$ bln

obtained in [18] as an approximate solution of the boundary layer equations. Here $\delta_\omega$ is the so-called vorticity thickness defined as

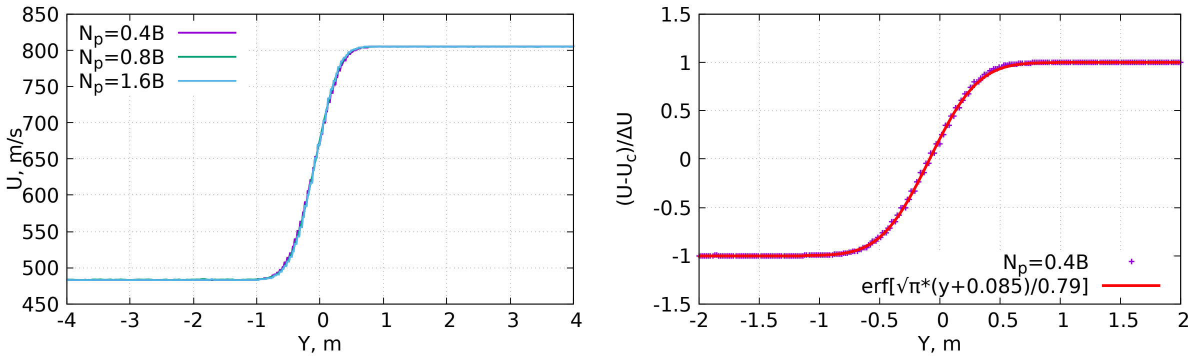
$$\delta_\omega = \frac{\Delta U}{(\partial U/\partial y)_{\max}} \tag{7}$$

and $y_c$ is the coordinate of the point where $U = U_c$.

One can see that the numerical and analytical solutions are in close agreement and the values of $\delta_\omega$ and $y_c$ for this cross-section equal 0.79 m and 0.085 m, respectively. It is worth noting than another analytical profile $U(x, y) = U_c + (\Delta U/2)\tanh(2y/\delta_\omega)$, which is also frequently used to approximate the mixing layer profile, does not yield so good agreement with our numerical results.

The Görtler approximate solution [18] predicts that $y_c = 0$, i. e. the mixing layer centerline coincides with the axis $y = 0$. However, as was shown in [19], the next order approximation implies a centerline deflection. The DSMC simulation reproduces this peculiar feature. The centerline location as a function of the longitudinal coordinate $y_c(x)$ is shown in Fig. 4a. As can be seen, it slightly deflects from the line $y = 0$ toward the lower (slower) stream and the deflection grows downstream.

The mixing layer thickness increases with the longitudinal coordinate. It is well known that initially the mixing layer spreads similar to a laminar boundary layer on a flat plate so that its

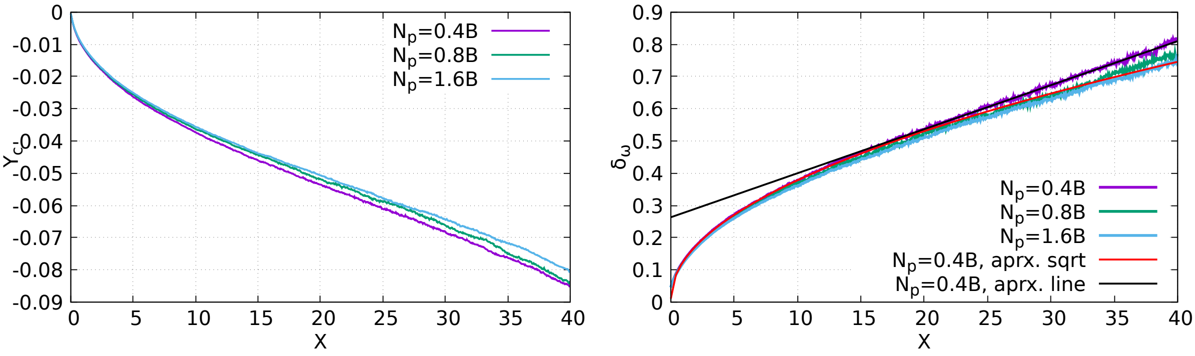


(a) (b)

**Figure 3.** Mean velocity profiles $U(y)$ at $x = 39$ m, comparison of the profiles for different numbers of test particles (a) and comparison with an analytical expression (b)

thickness grows proportional to $\sqrt{x}$. However, after the emergence of large-scale vortices, their pairing becomes the main mechanism of thickness growth [20]. It seems that this mechanism continues to dominate even in the turbulent regime when the large-scale vortices persist on the background of small-scale 3D pulsations [21]. As a result of successive vortex pairings, the mixing layer thickness grows linearly with the $x$ coordinate.

To determine the character of this dependency in the numerical simulations performed, a fitting curve has been calculated using a built-in function of the open source visualization software Gnuplot [22]. The result is shown in Fig. 4b. It can be seen that at $x \leq 15$ m the square root fit reproduces closely the behavior of the numerical curves. Farther downstream they are well-fitted by linear functions, however, the slopes are slightly different with the maximum slope for $N_p = 0.4$ bln and the minimum one for $N_p = 1.6$ bln. It can be explained by a smaller statistical noise level at a large particle number. As a result, the mixing layer is later excites and the growth of disturbances delays (see below). Thus, this characteristic reveals a dependence on the number of test particles.

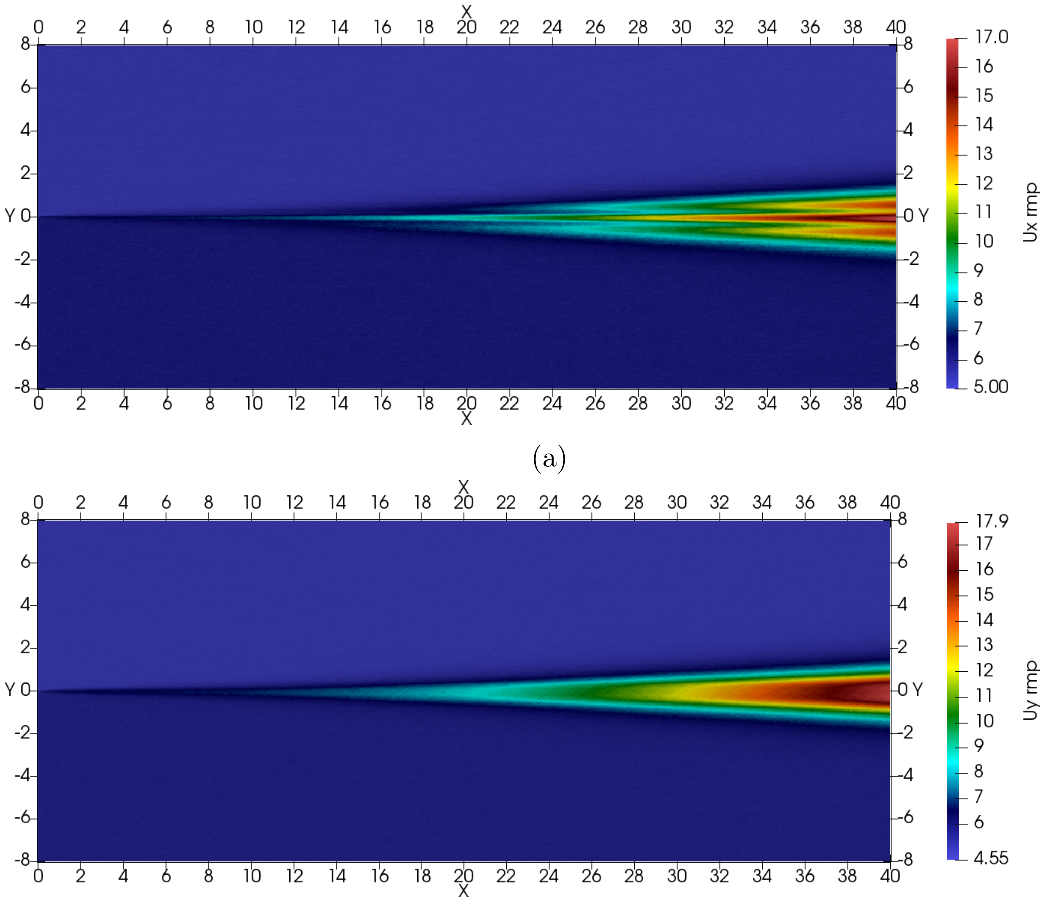


(a) (b)

**Figure 4.** Centerline deflection (a) and growth of vorticity thickness (b) for mixing layer

## 2.3. Growth of Disturbances

Figure 5 shows the flowfields of $u'_{rms}$ and $v'_{rms}$, RMS pulsations of two velocity components, for $N_p = 0.4$ bln. One can see their fast growth starting approximately from the streamwise location where the layer thickness begins grows linearly. A specific feature of the $u'_{rms}$ flowfield that, closer to the outflow boundary, its distribution across the mixing layers has three maxima.



(a)

(b)

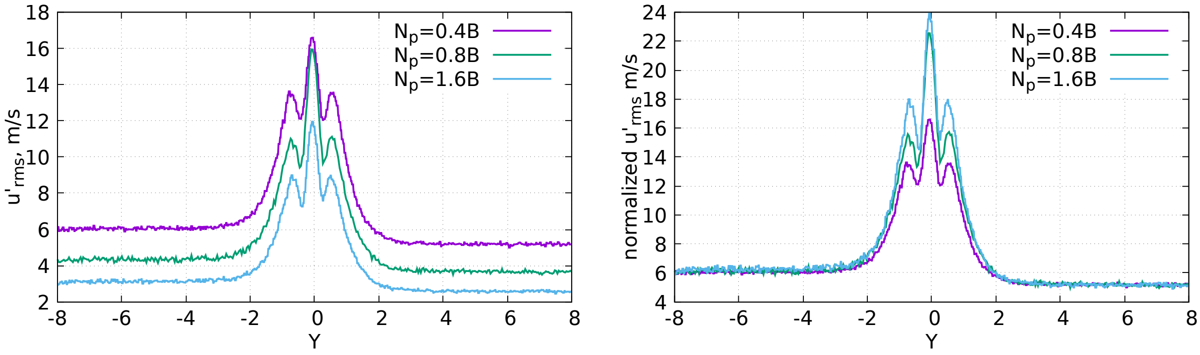
**Figure 5.** Flowfields of $u'_{rms}$ and $v'_{rms}$, RMS pulsations of longitudinal (a) and transversal (b) velocity components, for $N_p = 0.4$ bln

The profiles of RMS pulsations of $u'_{rms}$ in the cross-section $x = 39$ m for the same case $N_p = 0.4$ bln can be seen in Fig. 6.

Figure 6a shows that the profiles for the different number of particles are similar on qualitative level but differ quantitatively. This difference is resulted from the different level of statistical noise. It is well known that an increase in the number of samples by a factor of $N$ reduces the standard deviation of a quantity by a factor of $\sqrt{N}$. So, the RMS pulsation profiles are multiplied by the factor $\sqrt{N_p/N_p^0}$ where $N_p = 0.4$ bln and the normalized RMS pulsations $\sqrt{N_p/N_p^0}\, u'_{rms}$ are presented in Fig. 6b. As can be seen, after that the pulsations in the free stream regions coincide but in the mixing layer itself they differ substantially.

Thus, the characteristics of pulsations depend substantially on the number of particles used in a simulation $N_p$. The free stream values of fluctuations are simply proportional to $\sqrt{N_p}$. Thus, they will decrease with an increase in $N_p$ and vanish as $N_p \to \infty$. For the number of test particles equal to the number of real molecules, the magnitude of the free stream pulsations would be equal to that of thermal fluctuations in real gas.

However, the amplitude of disturbances in the mixing layer does not follow this normalization because they are resulted from the physical process of disturbances amplification due to the flow instability. The disturbances in the mixing layer are instability waves (with superimposed statistical fluctuations). They are originated from statistical fluctuations but then they are selectively amplified due to pumping of the energy by the Reynolds stresses from the mean



(a) (b)

**Figure 6.** Original $u'_{rms}$ (a) and normalized $\sqrt{N_p/N_p^0}\,u'_{rms}$ (b) profiles of RMS pulsations of longitudinal velocity in the cross-section $x = 39$ m

shear flow. At early stages of the hydrodynamics instability they grow exponentially, later their growth saturates because of nonlinear effects. With an increase in the number of particles, the magnitude of statistical fluctuations decreases so that the initial amplitude of instability waves decreases too. As a result, they will grow and reach approximately the same amplitude, however it will happen farther downstream from the meeting point of two mixing streams.

It is interesting to compare the RMS profiles with the results of the linear stability theory (LST). The linearized compressible Navier–Stokes equations along with the disturbance boundedness conditions at $y \to \pm\infty$ comprise a LST eigenvalue problem, see [11]. The eigenvalue problem is solved using the VMLS3D code developed by one of the authors. The LST computations are performed for the analytical profile (6). In Fig. 7 the profile $v'_{rms}$ at $x = 39$ m is compared with the LST eigenfuction of a spatially growing disturbance for the frequency corresponding to the maximum growth rate. The distributions are different in the freestreams where the LST eigenfunction exponentially decays while the flow disturbance in the DSMC simulation approaches a nearly constant value equal to the freestream level of statistical noise. At the same time, the two distributions are in close agreement inside the mixing layer.

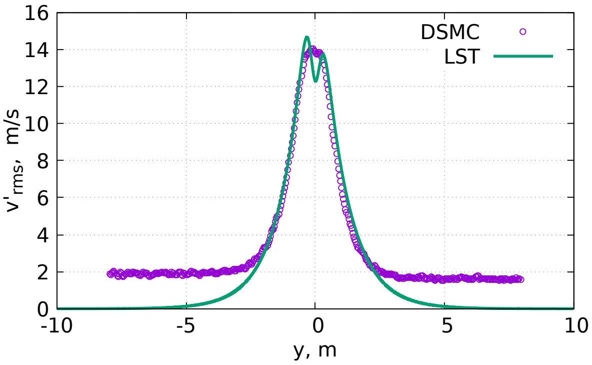


**Figure 7.** Comparison of the DSMC profile of $v'_{rms}$ at $x = 39$ m ($N_p = 0.4$ bln) with a LST eigenfunction

The statistical noise introduced by the DSMC method has a broadband spectrum. However, the flow instability amplifies its harmonic components selectively. As a result, the most unstable disturbance predicted by the linear theory starts dominating at some distance downstream. Even farther downstream nonlinear interactions between the harmonics result in an amplification of the subharmonic waves and the vortex pairing. A few successive pairings occur within the computational domain, however, as can be seen, even near the outflow boundary the transverse velocity fluctuation profile $v'_{rms}$ within the mixing layer keeps resemblance with the LST eigenfuction.

## 3. Identification of Vortices and Analysis of Vortex Motion

This section deals with a specialized procedure developed specifically for an analysis of instabilities in the presence of statistical fluctuations. Calculation of vorticity, Q-criterion and other techniques based on the spatial derivatives of flow quantities do not work in the case of statistically noisy data, so another approach, based on some integral quantities, was proposed.

For any frame, for each cell $i$ we calculate the quantity

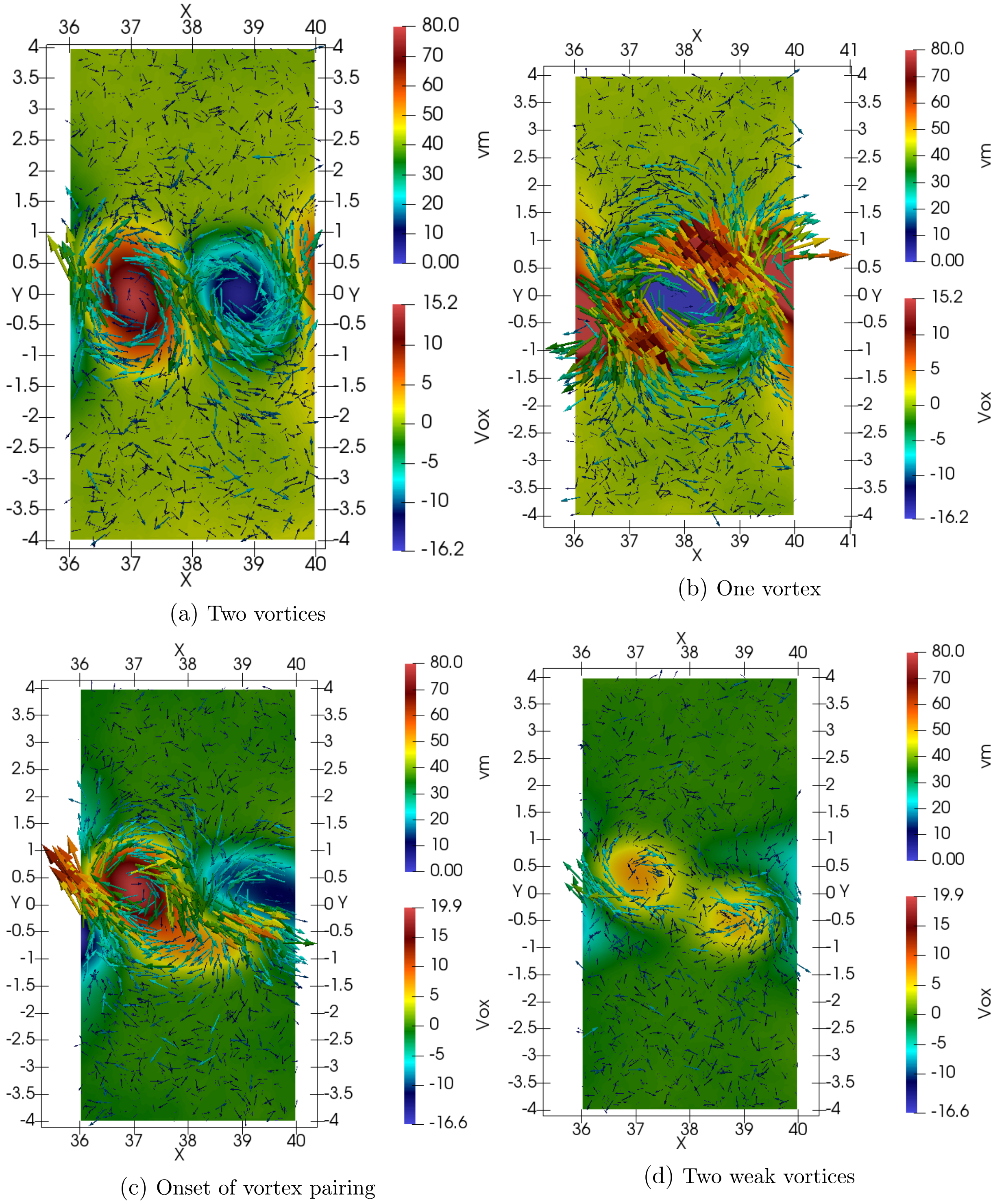
$$L_i = \frac{1}{\mathcal{N}} \sum_{j \in \mathrm{Nei}(i,\delta)} \left[ u'_j (y_j - y_i) - v'_j (x_j - x_i) \right], \qquad u'_j \equiv u_j - U_j, \quad v'_j \equiv v_j - V_j, \tag{8}$$

where $x_i$, $y_i$ and $x_j$, $y_j$ are the coordinates of cell centers $i$ and $j$, respectively, $\mathrm{Nei}(i,\delta)$ is the list of cells in the $\delta$-vicinity of the cell $i$, i.e. $|\mathbf{r}_j - \mathbf{r}_i| \equiv \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} < \delta$, $\mathcal{N}$ is the number of cells in the list, $u_j$, $v_j$ and $u'_j$, $v'_j$ are the velocity components and their perturbations in the center of the cell $j$ for this frame, $U_j$, $V_j$ are the velocity components at the same cell averaged over all frames. Thus, the velocity disturbance flowfield imposed on the mean flow is analyzed. It is easy to see that Eq. (8) is the $z$-component of a vector product including the vectors of perturbed velocity and relative position of the cell centers $i$ and $j$; for a 2D flow only this $z$-component is of importance. The parameter $\delta$ can be varied, in the present paper it is taken equal to 1 m.

The sign of $L$ determines if the vortex rotates in the clockwise, or counterclockwise direction. The definition ensures that in the center of a vortex the value of $L$ will reach a local maximum. Another feature of this quantity that it grows as the velocity of fluid rotating around the point $x_i$, $y_i$ increases.

The resulting flowfields of $L$ along with the original vector fields of the perturbed velocity $\mathbf{u}' \equiv (u', v')$ in the region $x > 36$ m near the outflow boundary are shown in Fig. 8 for 4 different time moments. The color palette **Vox** shows the vortex intensity $L$, while the vector length and their color in accordance with the scale **vm** show the magnitude of the perturbed velocity $|\mathbf{u}'|$. The frames presented are chosen in such a way that they demonstrate different dynamic events and phases of vortex motion. It is evident that they enable us to identify and recognize characteristic flow features despite the background statistical fluctuations.

The obtained flowfields of $L$ are further post-processed. For each frame $k$ the cells with maximum and minimum values of $L$ are found. They are considered centers of vortices rotating in the clockwise and counterclockwise direction, respectively, because, in our case, the maximum value is always positive while the minimum value is negative. After that, the mean values of $y$-coordinates of vortex centers for both these types of vortices are deduced by averaging over $n_f$ frames: $\overline{Y} = \sum_{k=1}^{n_f} Y^{(k)}/n_f$ where $Y^{(k}$ is the $y$-coordinate of the vortex center. The respective standard deviations $\sigma_Y = \sqrt{\sum_{k=1}^{n_f} (Y^{(k)} - \overline{Y})^2/n_f}$. The results are shown in Tab. 1. As one can see, on the average, the centers of counterclockwise vortices are located lower than those of clockwise vortices. Taking into account that the mixing layer centerline is somewhere between $y = -0.078$ m and $y = -0.085$ m, the intuitive description of the process would be that the



(a) Two vortices

(b) One vortex

(c) Onset of vortex pairing

(d) Two weak vortices

**Figure 8.** Vortices identified with the new criterion. Flowfields of $L$ (color scale **Vox**) with imposed perturbed velocity vectors (arrows, their length and color, in accordance with color scale **vm**, correspond to the magnitude of perturbed velocity)

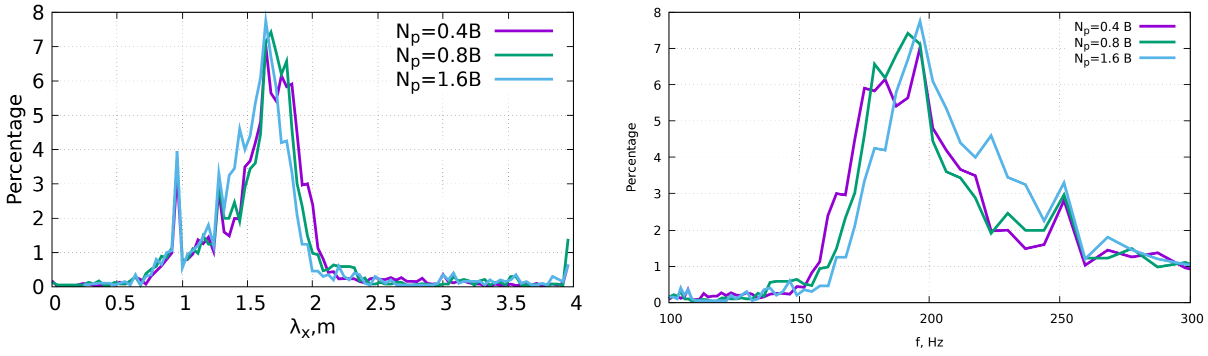
counterclockwise vortices move along the flow centerline with clockwise vortices slight above them.

The range 36 m $< x <$ 40 m (see Fig. 8) is chosen in such a way that for each frame it contains two vortices, one rotating clockwise and another – counterclockwise. The distribution of $\lambda_x$, the distances along $x$ between vortex centers, is shown for all frames in Fig. 9a. The distances

**Table 1.** Averaged positions of clockwise and counterclockwise vortices

| $N_p$, $10^9$ | Clockwise | | Counterclockwise | |
|---|---|---|---|---|
| | $\overline{Y}$, m | $\sigma_Y$, m | $\overline{Y}$, m | $\sigma_Y$, m |
| 0.4 | −0.086 | 0.29 | −0.042 | 0.31 |
| 0.8 | −0.076 | 0.31 | −0.044 | 0.33 |
| 1.6 | −0.077 | 0.27 | −0.029 | 0.29 |

of 1.6–1.8 m dominate. The results of numerical simulations with different numbers of particles are quite similar. In all simulations there is a suspicious peak at $\lambda_x = 0.9$ m; it is possible that the peak is connected with an unknown numerical artifact. If one supposes that the vortices convect downstream with the average velocity $U_c = (U_1 + U_2)/2 = 644$ m/s, then the frequency of occurrence of vortices rotating in the same direction is $f = U_c/(2\lambda_x)$. The percentage of vortices as a function of the frequency is shown in Fig. 9b. The most probable frequencies are 170–200 Hz.



(a) (b)

**Figure 9.** Distribution of distances along $x$ between vortex centers (a) and distribution of frequency of vortex occurrence (b)

It can be concluded that the number of test particles and, therefore, the level of statistical noise have no impact on such characteristics of vortices as the distance between them and the frequency of their occurrence. It looks reasonable because these quantities are connected with parameters of the KH instability. They are determined by the wavelength of the most unstable disturbance and its frequency, respectively, i. e. by quantities that depend on mean flow properties but not on the initial amplitude of disturbances. At the same time, the initial amplitude of disturbances should have impact on their amplitudes in successive cross-sections and, consequently, on the intensity of vortex motion.

The vortex intensity can be measured for each frame as a difference between the maximum and minimum values of $L$. Time evolution of this parameter calculated over the range 36 m $< x <$ 40 m in three simulations with different numbers of particles is shown in Fig. 10. As can be seen, most of the time the vortex intensity is the largest in the simulation with $N_p = 0.4$ bln, and the smallest – at $N_p = 1.6$ bln. The time averaged vortex intensity in this cross-section is equal to 21.5, 17.4 and 13.95 for $N_p = 0.4$ bln, 0.8 bln and 1.6 bln, respectively.

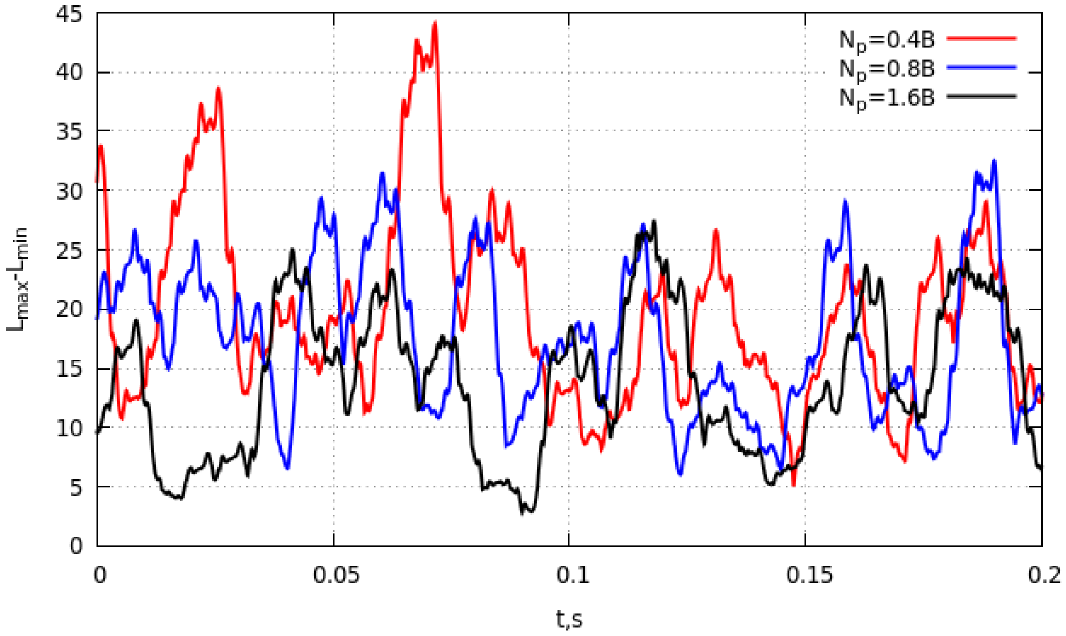


**Figure 10.** Time evolution of vortex intensity for different numbers of model particles

## Conclusion

A numerical study of the instability developing in a supersonic mixing layer of monatomic gas (argon) is performed using the DSMC method, a particle-based method for solving the Boltzmann transport equation. The computations are carried out with the SMILE-GPU software system implemented on a hybrid multiple GPU/CPU computational cluster. No other flow excitation except statistical noise resulted from the stochastic DSMC method is used. A new algorithm for postprocessing and analyzing noisy computational flowfields is proposed and applied to extract data on vortex motion in the unstable mixing layer.

The main features of the Kelvin–Helmholtz instability such as vortex formation and successive vortex pairings are reproduced in the DSMC simulations. It is shown that the number of test particles used has no impact on instability parameters independent on the initial amplitudes of flow disturbances but can substantially change the pulsation characteristics and the intensity of vortex motion at a fixed distance from the mixing layer starting point.

## Acknowledgments

This work was supported by the Russian Science Foundation, project No. 23-11-00258. All computations were performed using the equipment of the Equipment Sharing Center “Mechanics” of ITAM SB RAS.

We are also grateful to an anonymous reviewer whose remarks helped us noticeably improve the paper.

## References

1. Bird, G.A.: Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon Press, Oxford (1994).

2. Gallis, M.A., Koehler, T.P., Torczynski, J.R., Plimpton, S.J.: Direct simulation Monte Carlo investigation of the Richtmyer–Meshkov instability. Phys. Fluids 27(8), 084105 (2015). https://doi.org/10.1063/1.4928338

3. Gallis, M.A., Koehler, T.P., Torczynski, J.R., Plimpton, S.J.: Direct simulation Monte Carlo investigation of the Rayleigh–Taylor instability. Phys. Rev. Fluids 1(4), 043403 (2016). https://doi.org/10.1103/PhysRevFluids.1.043403

4. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: Investigation of Kelvin–Helmholtz instability with the DSMC method. AIP Conf. Proceedings 2125, 030028 (2019). https://doi.org/10.1063/1.5117410

5. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: DSMC simulation of instability of plane supersonic jet in coflow. AIP Conf. Proceedings 2504, 030087 (2023). https://doi.org/10.1063/5.0132264

6. Gallis, M.A., Bitter, N.P., Koehler, T.P., *et al.*: Molecular-level simulations of turbulence and its decay. Phys. Rev. Lett. 118(6), 064501 (2017). https://doi.org/10.1103/PhysRevLett.118.064501

7. Gallis, M.A., Torczynski, J.R., Bitter, N.P., *et al.*: Gas-kinetic simulation of sustained turbulence in minimal Couette flow. Phys. Rev. Fluids. 3(7), 071402 (2018). https://doi.org/10.1103/PhysRevFluids.3.071402

8. Gallis, M.A., Torczynski, J.R., Krygier, M.C., *et al.*: Turbulence at the edge of continuum. Phys. Rev. Fluids. 6(1), 013401 (2021). https://doi.org/10.1103/PhysRevFluids.6.013401

9. McMullen, R.M., Krygier, M.C., Torczynski, J.R., Gallis, M.A.: Navier–Stokes equations do not describe the smallest scales of turbulence in gases. Phys. Rev. Lett. 128(11), 11450 (2022). https://doi.org/10.1103/PhysRevLett.118.064501

10. Kashkovsky, A.V.: 3D DSMC computations on a heterogeneous CPU-GPU cluster with a large number of GPUs. AIP Conf. Proceedings 1628(1), 192–198 (2014). https://doi.org/10.1063/1.4902592

11. Betchov, R., Criminale W.O.: Stability of Parallel Flows, Academic Press, New York (1967).

12. Jackson, T.L., Grosch, C.E.: Inviscid spatial stability of a compressible mixing layer. J. Fluid Mech. 208, 609–637 (1989). https://doi.org/10.1017/S002211208900296X

13. Ragab, S.A., Wu, J.L.: Linear instabilities in two-dimensional compressible mixing layers. Phys. Fluids A 1(6), 957–966 (1989). https://doi.org/10.1063/1.857407

14. Kudryavtsev, A.N., Solov'ev, A.S.: Stability in the shear layer of compressible gas. J. Appl. Mech. Techn. Phys. 30(6), 949–956 (1989). https://doi.org/10.1007/BF00851504

15. Yang, J.-Y., Chang, J.-W.: Rarefied flow instability simulation using model Boltzmann equations. AIAA Paper No. 97-2017 (1997). https://doi.org/10.2514/6.1997-2017

16. Kudryavtsev, A.N., Poleshkin, S.O., Shershnev, A.A.: Numerical simulation of the instability development in a compressible mixing layer using kinetic and continuum approaches. AIP Conf. Proceedings 2125, 030033 (2019). https://doi.org/10.1063/1.5117415

17. Kotel'nikov, V.A.: On the transmission capacity of 'ether' and wire in electric communications. Physics–Uspekhi 49(7), 736–744 (2006). https://doi.org/10.1070/PU2006v049n07ABEH006160

18. Görtler, H.: Berechnung von Aufgaben der freien Turbulenz auf Grund eines neuen Näherungsansatzes. ZAMM 22(5), 244–254 (1942). https://doi.org/10.1002/zamm.19420220503

19. Ting, L.: On the mixing of two parallel streams. J. Math. Phys. 38, 153–165 (1959). https://doi.org/10.1002/sapm1959381153

20. Winant, C.D., Browand, F.K: Vortex pairing: the mechanism of turbulent mixing layer growth at moderate Reynolds number. J. Fluid Mech. 63(2), 237–255 (1974). https://doi.org/10.1017/S0022112074001121

21. Brown, G.L., Roshko, A.: On density effects and large structure in turbulent mixing layers. J. Fluid Mech. 64(4), 775–816 (1974). https:/doi.org/10.1017/S002211207400190X

22. Gnuplot homepage. http://www.gnuplot.info

DOI: 10.14529/jsfi240205

# Numerical Simulation of Rarefied Flow Instabilities Using Kinetic Approach

*Anton A. Shershnev*[1], *Alexey N. Kudryavtsev*[1], *Alexander V. Kashkovsky*[1], *Semyon P. Borisov*[1], *Sergey O. Poleshkin*[1]

The results of numerical investigations of hydrodynamic instabilities in rarefied flows obtained using the kinetic approach are presented and discussed. The rarefied flow instabilities are simulated using both statistical and deterministic methods to solve kinetic equations for the velocity distribution function. The Direct Simulation Monte Carlo method is employed for statistical modeling of rarefied flows while the deterministic methods used are high-order finite-difference schemes for solving the Boltzmann transport equation and model kinetic equations in a multidimensional phase space. Numerical solvers used for the simulations are run on multiple-GPU clusters using CUDA platform and MPI communication interface. The development of instabilities in compressible free shear flows such as a mixing layer between two parallel streams and a plane jet in a coflow is considered. The Rayleigh–Taylor and Richtmyer–Meshkov instabilities induced by external body forces are also simulated. The results of kinetic simulations are compared with data from the linear stability theory and Navier–Stokes computations and it is shown that good agreement is observed between kinetic and continuum approaches.

*Keywords: DSMC method, the Boltzmann transport equation, model kinetic equations, mixing layer, plane jet, the Rayleigh–Taylor instability, the Richtmyer–Meshkov instability.*

## Introduction

Fluid flow instabilities are a common physical phenomenon important for many problems in science and engineering. They are usually considered on the basis of the continuum approach since they happen at high Reynolds numbers and, therefore, low Knudsen numbers. However, in several important cases, flow instabilities may emerge in conditions when the effects of flow rarefaction are pronounced. Those cases include, for example, instabilities associated with the action of mass forces (gravity force in the Rayleigh–Taylor instability and Rayleigh–Benard convection, inertial force in the Richtmeyer–Meshkov instability). Flow instabilities may also emerge at low Reynolds numbers in free shear flows (mixing layers, jets and wakes), flows with curved streamlines (such as the Taylor–Couette flow) and, as an extreme case, in a collisionless medium with long-range forces. Rarefaction effects may become noticeable at instability of high-speed flows. Numerical simulation of such phenomena demands a step beyond the standard continuum approach based on the Navier–Stokes equations and should be based on the molecular-kinetic description taking into account atomistic structure of matter.

There is a number of kinetic approaches that can be used for this purpose: the Molecular Dynamics (MD), the Direct Monte Carlo Simulation (DSMC), finite-difference methods that solve the Boltzmann transport equation or its simplified versions, the so-called model kinetic equations, on a grid in a multidimensional phase space. Since 1980s the MD and the DSMC were applied to the simulation of the "stability switch" in problems of convection in the planar layer of gas [1–4], and formation of the Taylor vortices in the cylindrical Couette flow [5–7]. The stability switch means that, as a result of an instability, one steady flow is replaced by another but also steady flow ("secondary flow"). Slightly later the stationary Taylor–Görtler longitudinal vortices

[1] Khristianovich Institute of Theoretical and Applied Mechanics Siberian Branch of Russian Academy of Sciences, Novosibirsk, Russian Federation

in the curved mixing layer of a supersonic underexpanded jet were simulated by directly solving the Boltzmann equation in the 6D phase space [8, 9].

However, in general, the instability development is connected with time dependent disturbances. Even if a steady secondary flow forms, unsteady processes will dominate in later stages of the instability leading finally to transition to turbulence. Kinetic simulations of such complicated unsteady phenomena require a huge amount of computer resources. It seems that the first time-accurate kinetic simulation of a flow instability was performed in [10] where early stages of the Kelvin–Helmholtz instability were simulated by solving the model kinetic equations.

In recent years, with increasing computer power, numerical investigation of flow instabilities on the kinetic level has become an actively developing research field. Rarefied flow instabilities were simulated using the MD [11, 12], the DMSC method [13–19] and deterministic solution of model kinetic equations or, even, the Boltzmann transport equation [20–22].

In the current paper, we overview and discuss the results of numerical investigations of hydrodynamic instabilities in rarefied flows obtained in recent years in Laboratory of Computational Aerodynamics at Khristianovich Institute of Theoretical and Applied Mechanics, Novosibirsk (ITAM SB RAS). The numerical simulations have been performed with a number of numerical kinetic techniques including both the statistical DSMC method and deterministic approaches for solving kinetic equations in the multidimensional phase space. Navier–Stokes computations have been undertaken for comparison and, sometimes, for parametric investigation.

All numerical codes used for these simulations are developed in the Laboratory and designed for massive parallel computations on hybrid (GPU/CPU) computational clusters employing, for parallelization, NVIDIA CUDA platform and MPI communication interface.

The problems simulated include instabilities of high-speed free shear flows – the Kelvin–Helmholtz instability of a supersonic mixing layer, the instabilities developing in supersonic plane jets with co-flow, the instabilities driven by body forces – the Rayleigh–Taylor instability of the interface between heavier and lighter gases in a gravitational field, the Richtmyer–Meshkov instability developing on a contact surface after passing a shock wave. The instabilities in gravitating collisionless media such as those presented in [23] are out of scope of this paper.

The remainder of the paper is organized as follows. In Section 1 the numerical solvers and parallelization techniques used are reviewed. The results of numerical simulations of rarefied flow instabilities are presented and discussed in Section 2. The study is briefly summarized in Conclusion.

# 1. Solvers and Parallelization Techniques

In this section a brief description of the solvers used for the numerical simulations is presented. Although each solver is based on the different set of equations, they all are developed using the same paradigm. The codes are written in C++ and use CUDA API to perform computations on GPUs. The domain decomposition technique is employed for parallelization and data exchange between processors is performed via MPI library. Some details on code implementation are given below.

## 1.1. Direct Simulation Monte Carlo Solver

Software systems of the SMILE family, namely SMILE++ and SMILE-GPU, are used in the DSMC simulations. The former is a well-established numerical code which takes most of the advantages of object-oriented programming including inheritance and encapsulation, and contains a large variety of physical models [24]. It is based on the efficient majorant frequency scheme for collisions and uses a nested octree-like Cartesian computational grid. Since the DSMC is a particle-based method, the computational load depends directly on the number of model particles in the simulation, conventional domain decomposition does not allow to divide the computational load evenly. In the case of the SMILE++ code this problem is treated by distributing computational cells randomly across the CPUs, which yields relatively balanced computational times and a good parallelization efficiency.

The SMILE-GPU code [25] is a high-performance version of the DSMC solver specifically for hybrid architectures and is designed for large 3D computations. Computations on multiple GPUs are performed using the domain decomposition approach along with adaptive load balancing. In accordance with Direct Timing Load Balance algorithm [25], computational *cells* are periodically (usually every 1000 time steps) redistributed between GPUs depending on the average wall clock time spent by each GPU on the one iteration. The smaller the time of computation on a certain GPU the more additional cells will be assigned to it at the next load balance correction. Usually after 10–20 iterations of balance corrections it yields approximately 95–99% efficiency. It should be noted that this algorithm works quite well even for unsteady flows.

## 1.2. Solver for Boltzmann and Model Kinetic Equations

Another computational code used for numerical simulation of rarefied flow instabilities is based on finite-difference methods for kinetic equations governing the evolution of the particle velocity distribution function. The code employs the method of discrete ordinates in the velocity space and high-order TVD and WENO schemes for the spatial approximation. Intermolecular collisions are taken into account using either the full multidimensional Boltzmann collision integral or a simplified relaxation-type term. In the former case, the collision integral is evaluated using the numerical techniques described in [26, 27]. In the latter case, the relaxation-type term can be taken in the form corresponding to the BGK equation [28], the statistical ellipsoidal model [29] or the Shakhov model [30]. The composite Simpson rule or the Gauss–Hermite quadratures are used to evaluate integrals in the velocity space [31]).

The code is written in C++ programming language and implements CUDA API and MPI library to run simulations on multiple GPUs. Decomposition of the computational domain in the physical space is used to distribute the load between processor units [27].

## 1.3. Navier–Stokes Solver

To compare results of kinetic and continuum simulations, some Navier–Stokes computations have also been performed. In these cases, the HyCFS-R solver for compressible flows of gas mixtures [32] is used. The Navier–Stokes equations in general curvilinear coordinates are solved on a multiblock structured grid using TVD and WENO shock-capturing schemes for spatial discretization and Runge–Kutta TVD schemes for time advancement. The code is primarily designed for hybrid CPU/GPU computational environments and its multilevel parallelization is based on the domain decomposition used with CUDA API, OpenMP API and MPI protocol. The domain is divided into separate partitions, each partition is assigned to a specific GPU and then inter-partition exchange is carried out using either OpenMP if both blocks are located on the same computational node, or using MPI library otherwise.

# 2. Results of Numerical Simulations

## 2.1. Kelvin–Helmholtz Instability in a Mixing Layer

A mixing later between two parallel streams can be considered as a prototype of all free shear flows such as jets and wakes. The Kelvin–Helmholtz (KH) instability [33] is the dominant instability mechanism in the mixing layer at low convective Mach numbers $M_c = (U_1 - U_2)/(a_1 + a_2)$, where $U_1$, $U_2$ are the velocities of two streams and $a_1$, $a_2$ are speeds of sounds in them. The mixing layer is subject to the KH instability even at low Reynolds numbers.

The KH instability can be simulated as developing either in time or in space. In the former case the flow in assumed to be periodic along the streamwise coordinate so that the mean flow and disturbances evolve in time. In the latter case they develop spatially; this formulation is closer to practice but more difficult to simulate numerically because it requires a much longer computational domain.

Numerical simulations based on the kinetic approach are performed in both formulations. DSMC simulations of the mixing layer are considered in the companion paper [34].

When the flow is simulated by solving the model kinetic equations, a computational grid in 4D (2 physical coordinate and 2 velocity components) phase space is used. It has been shown [31] that for low-speed flows the Gauss–Hermite quadratures are particularly efficient for evaluation of integrals in the velocity space. Numerical tests showed that the grid of 11 × 11 points in the velocity space ensures integration of the initial flowfield with machine accuracy when the temporal mixing layer is simulated at the convective Mach number $\mathrm{M}_c = 0.2$ by solving the BGK equation.

In these simulations, the mean flow velocity is taken as $U(y) = \tanh(y)$, the temperatures of two streams are assumed to be equal, $T_1 = T_2$. An initial perturbation imposed on the mean flow is a superposition of the fundamental harmonic (most unstable disturbance in accordance with the linear stability theory) and its first and second subharmonics. The initial amplitudes of the subharmonics are much smaller than that of the fundamental harmonic.

The main stages of the instability development include the initial growth and saturation of the most unstable disturbance, the growth of the first subharmonic via the subharmonic resonance mechanism, then the growth of the second subharmonic. The fast growth of two subharmonics looks in the physical space as two successive vortex pairings. The process of vortex pairing reproduced on the grid of 200 × 200 points in the coordinate space is shown in Fig. 1.

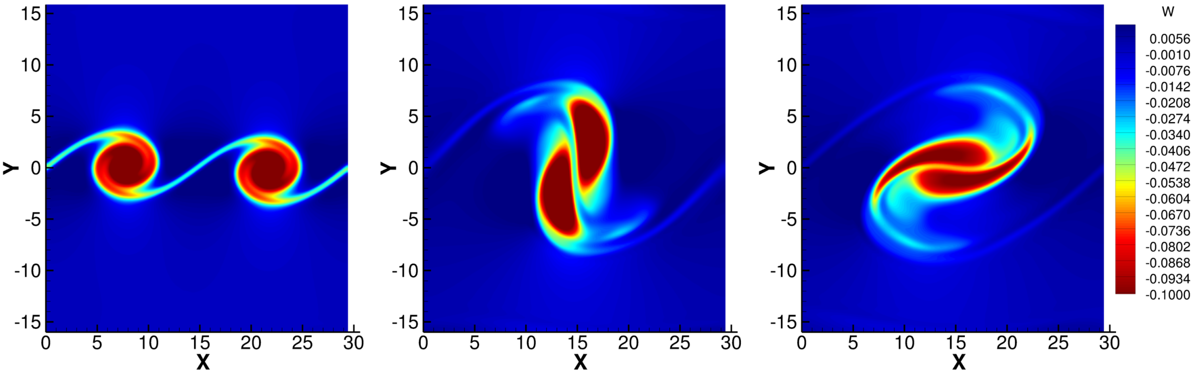
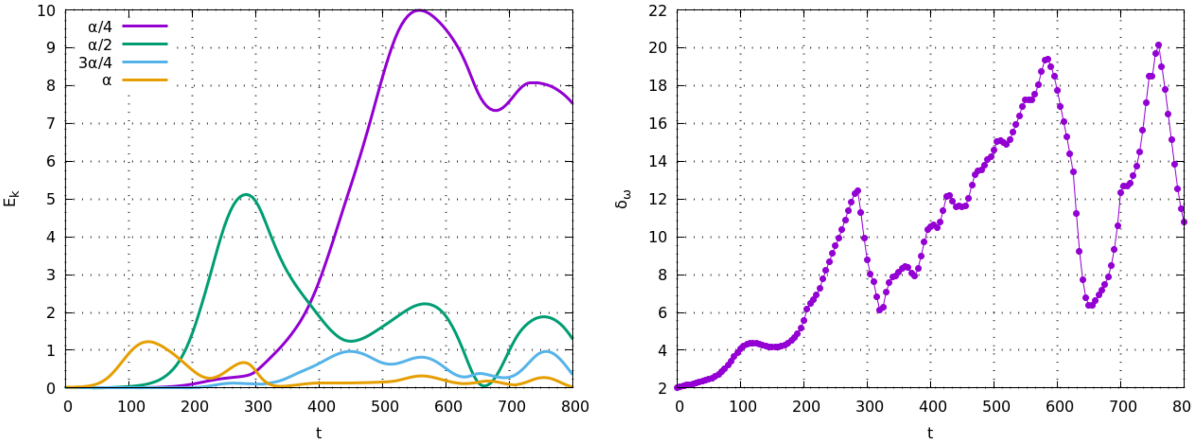


**Figure 1.** Vortex pairing in numerical simulation of the KH instability using the BGK kinetic equation. Vorticity flowfields in three time moments. $\mathrm{M}_c = 0.2$, $T_2/T_1 = 1$

Figure 2a shows the time evolution of energies of the fundamental harmonic and two subharmonics with the wave numbers $\alpha$, $\alpha/2$ and $\alpha/4$, respectively. As a result of the subharmonic resonance [35], the energy is pumped, first, from the fundamental harmonics to its subharmonic and, then, from the first subharmonics to the second one. In the physical space it looks like successive vortex pairings.



(a) Energy of harmonics as a function of time

(b) Vorticity thickness as a function of time

**Figure 2.** Time evolution of the supersonic mixing layer

Figure 2b shows the so-called vorticity thickness $\delta_\omega = (U_1 - U_1)/(\partial U/\partial y)_{\max}$ as a function of time. The peaks at $t \approx 290$, 590 and 750 correspond to the time moments when interacting pairs of neighboring vortices roll up around one another. Immediately after the vortex mergings the vorticity thickness abruptly drops.

The kinetic simulation successfully reproduces all stages of the KH instability development preceding the onset of 3D instability and transition to turbulence.

## 2.2. Instability of Supersonic Plane Jets in Coflow

A plane jet is another free shear flow unstable even at low Reynolds numbers. The linear theory predicts [36] that, at subsonic convective Mach numbers, there are two unstable modes of disturbances, the varicose (symmetric) mode and the sinuous (antisymmetric) mode. At higher supersonic $\mathrm{M}_c$ there appear multiple new unstable modes connected with resonant reflection of disturbances from the mixing layers bounding the jet. As a rule, the sinuous mode is the most unstable one.

Two different cases are considered with the jet Mach numbers $\mathrm{M}_j = 2.5$ and 4.5. In both cases there is a parallel external coflow with the Mach number $\mathrm{M}_a = 1.5$. The jet and coflow temperatures are the same, so that the convective Mach numbers are equal to 0.5 and 1.5, respectively. The Knudsen number based on the jet width at the inflow and the mean free path in the coflow is 0.0058 in both cases.

The numerical simulations presented here are performed using the DSMC method. No initial perturbations are imposed on the mean flow because it can be expected that flow disturbances will develop from statistical fluctuations always present in DSMC simulations.

The SMILE-GPU code running on a computational cluster equipped with Nvidia Tesla V100 GPUs is employed for these simulations. For $\mathrm{M}_j = 2.5$ the total number of test particles is $2 \cdot 10^9$

and the computation is performed on 12 GPUs. The wall-clock time per iteration is equal to 0.7 seconds. The corresponding figures for the $M_j = 4.5$ case are $10^9$ test particles, 8 GPUs and 0.5 seconds.

The computed flowfields sampled on the grid of 1000 × 400 cells and averaged over a small time interval (equal to 200 time steps) are virtually indistinguishable from the results of Navier–Stokes simulations. As predicted by the linear stability theory, the sinuous mode of instability develops quite rapidly at the subsonic convective Mach number $M_c = 0.5$, see Fig. 3. Vortex interactions and pairings are observed at early stages of the instability development. The mean velocity profiles show that the jet mixing layers spread and meet one another, after that the centerline velocity starts to decrease while the jet width grows. Near the outflow boundary, the velocity on the jet axis reduces by about 1.5 times, and the mean velocity profile takes a characteristic "bell-like" shape, typical for a developed jet flow.

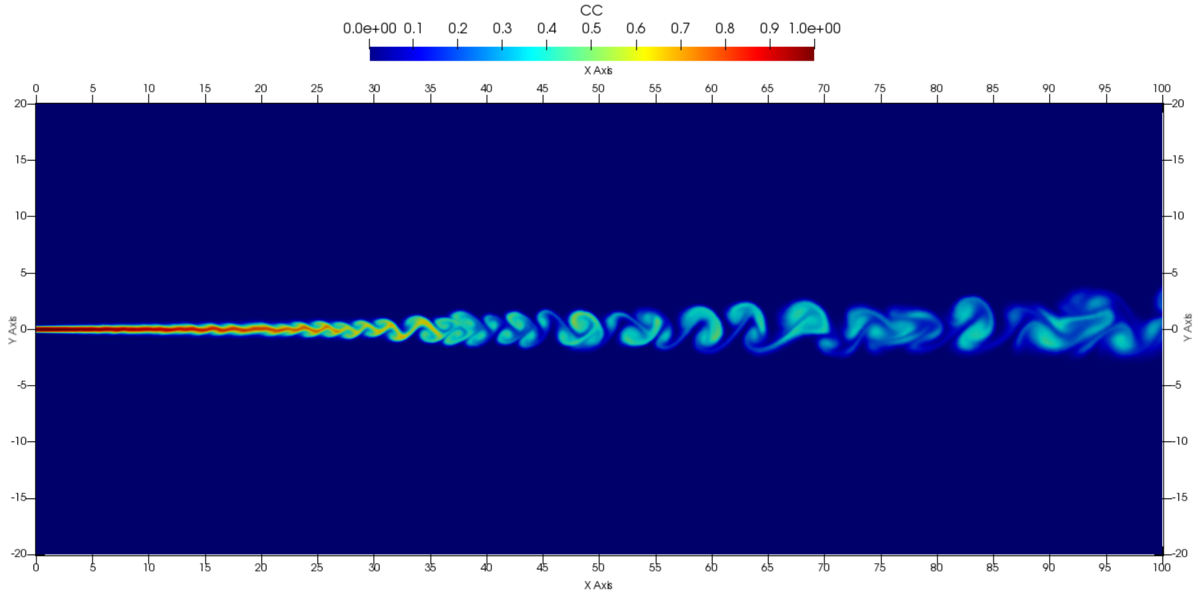


**Figure 3.** Plane jet instability. Flowfield of concentration of particles issuing from the nozzle. $M_j = 2.5$, $M_a = 1.5$

The antisymmetric mode also dominates at the supersonic convective Mach number $M_c = 1.5$, see Fig. 4. The vortex structures forming in this case are larger in size, and the breakdown of the jet column occurs at shorter distances from the nozzle exit. Weak oblique shock waves radiated from the jet are generated by the vortex structures moving with a supersonic speed relative to the external coflow.

Due to inherently stochastic excitation the flow instability develops more naturally in DSMC simulations than in computations with imposed artificial disturbances based on the deterministic approach. Thus, locations of successive vortex pairings are not fixed but scatter over some interval, so that the time-average jet width monotonically increases with the distance and does not include such abrupt peaks and drops as in Fig. 2b.

## 2.3. Rayleigh–Taylor Instability

The Rayleigh–Taylor (RT) instability occurs when a heavier fluid in a gravitational field is located above a lighter fluid. It is observed in a wide variety of natural phenomena and technical

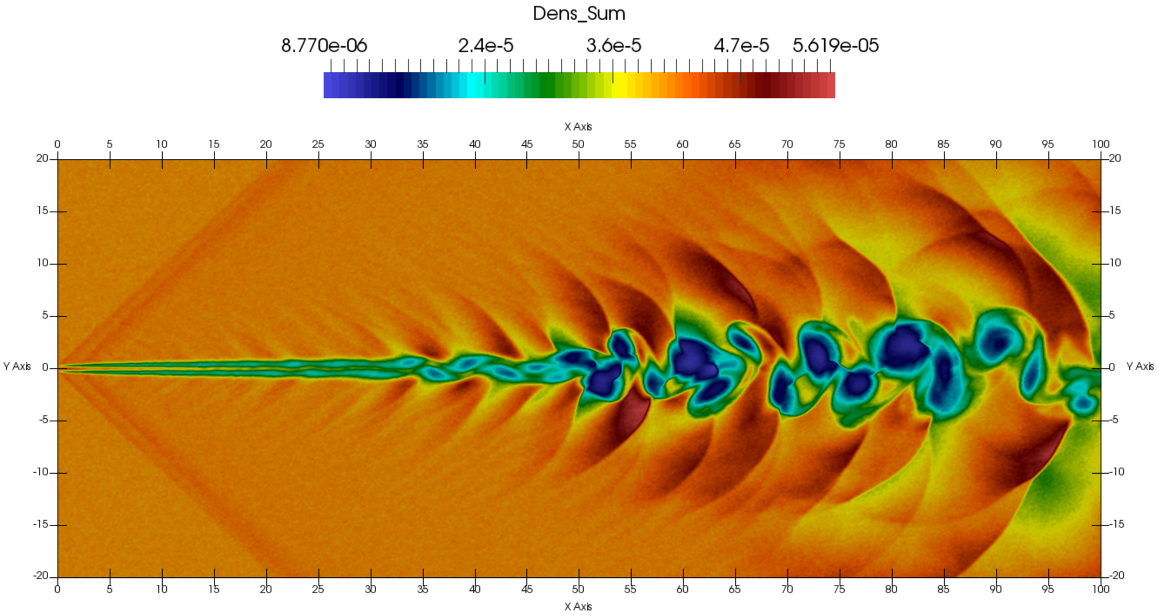


**Figure 4.** Plane jet instability. Density flowfield. $\mathrm{M}_j = 4.5$, $\mathrm{M}_a = 1.5$

applications, from supernova explosions and accretion of matter on black holes to inertial fusion, the formation of oddly shaped clouds and the flow of liquid from an inverted glass. The RT instability has been investigated in many theoretical, experimental, and numerical studies, see, for example, recent comprehensive reviews [37, 38].

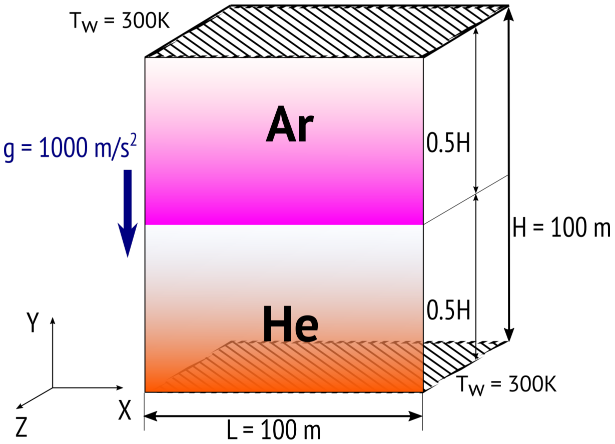


**Figure 5.** Rayleigh–Taylor instability. General scheme and computational domain geometry

We simulated numerically the RT instability developing in a rarefied Ar/He mixture in 2D and 3D formulations using the SMILE++ and HyCFS-R codes to assess the effects of rarefaction and non-equilibrium. A sketch of domain and general problem setup are shown in Fig. 5. A two-layer system consisting of argon layer over a helium layer in a gravitational field is simulated. The interface between gases at the initial moment of time is parallel to the $X$ axis. Densities and pressures on both sides of the interface are distributed at the initial moment in accordance with barometric formulas.

The size of the computational domain is 100 × 100 m, the mean free paths in gases at the interface are $\lambda_{\mathrm{He}} = 2.8 \times 10^{-2}$ and $\lambda_{\mathrm{Ar}} = 8.8 \times 10^{-3}$ m, and the densities are $\rho_{\mathrm{He}} = 10^{-6}$ and $\rho_{\mathrm{Ar}} = 10^{-5}$ kg/m$^3$, respectively. The pressure at the interface is $p = 0.62$ Pa. If the wavelength of emerging disturbances (see below) is taken as a flow characteristic scale, then the Knudsen number is $\mathrm{Kn} = 3.6 \cdot 10^{-3}$.

In order to reduce the difference between the characteristic time of instability development and the characteristic molecular times, the acceleration of gravity $g$ is increased to 1000 $\mathrm{m/s^2}$. The Atwood number $\mathrm{At} = (\rho_1 - \rho_2)/(\rho_1 + \rho_2)$ of the problem is equal to $9/11$.

The DSMC computations are carried out using up to 60 million model particles, time step is equal to $4 \times 10^{-6}$ s, the flow macroparameters are averaged over 1000 time steps for total of 2 million time steps. The continuum computations are performed on a grid consisting of up to 8 million cells using 2nd order TVD scheme and 2nd order explicit RK TVD time integration scheme.

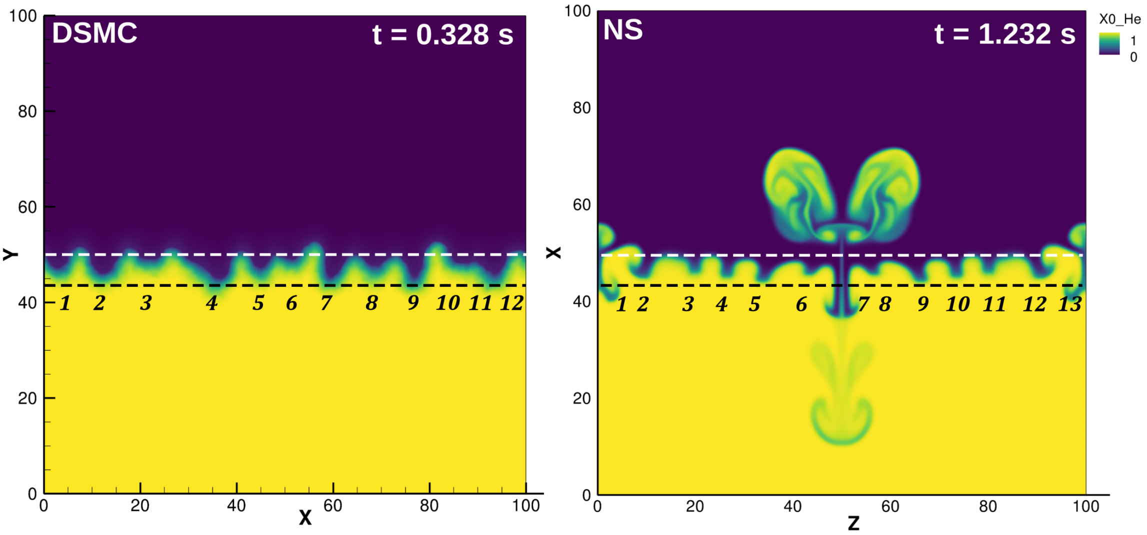


**Figure 6.** Comparison of He molar fraction in DSMC and NS computations

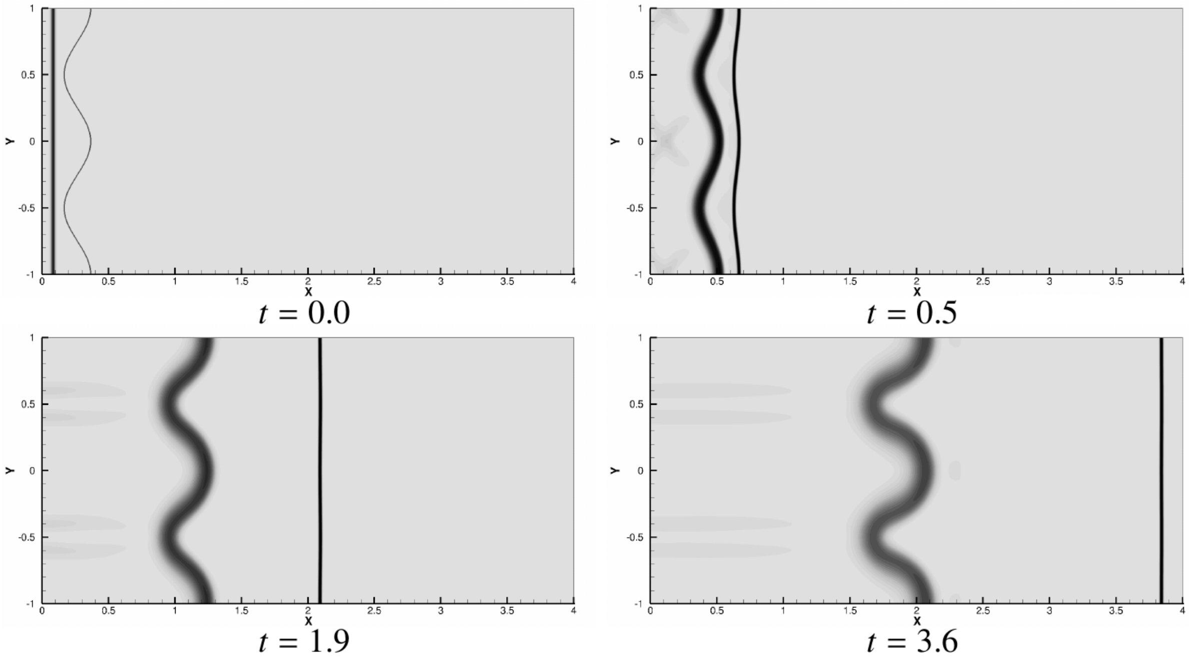
Because of different types of disturbances used for initial flow forcing in kinetic and continuum computations, the instabilities emerge in slightly different ways and at different rates. However, qualitative analysis shows that at the early stage of the instability development both approaches predict similar wavelengths of emerging disturbances although at different time instants. In Fig. 6 the molar fraction of helium is presented for both simulations. The white dashed line indicates the original position of the Ar/He interface, and the black dashed line shows the maximum amplitude of the spikes. As can be seen, the numbers of spikes are similar and their shapes are roughly the same. So it seems that in both cases, instability waves corresponding to the same fastest growing frequency develop from a wide spectrum of superposed disturbances. In the DSMC computation, the spikes are considerably less uniform than in the continuum simulation because the DSMC method induces a high level of the macroparameters fluctuations as mentioned above.

## 2.4. Richtmyer–Meshkov Instability

The Richtmyer–Meshkov (RM) instability develops when lighter and heavier fluids separated by a contact surface are impulsively accelerated by a passing shock wave. It can be considered as a Rayleigh–Taylor instability with the gravity force replaced by an inertial force acting during a small time interval. This instability is important in astrophysics, controlled thermonuclear fusion, supersonic combustion and many other areas [39].

Numerical simulations of the RM instability are performed by solving the BGK model kinetic equation, by using the DSMC method and on the basis of the Navier–Stokes equations. The Mach number of the incident shock $\mathrm{M}_s$ is varied from 1.5 to 8, the density ratio across the contact surface $\rho_2/\rho_1$ – from 2 to 10. The rarefaction degree is determined by the value of the Reynolds number Re based on the wavelength of a sinusoidal perturbation imposed on the contact surface at the initial moment or by the corresponding value of the Knudsen number Kn. The Reynolds number is varied from 50 to infinity (in an inviscid continuum computation).

A typical pattern of the RM instability development is shown in Fig. 7 where the results of a BGK simulation performed for a monatomic gas are presented. The computational grid contains 1200 × 300 points in the physical space and 33 × 33 points in the velocity space.



$t = 0.0$

$t = 0.5$

$t = 1.9$

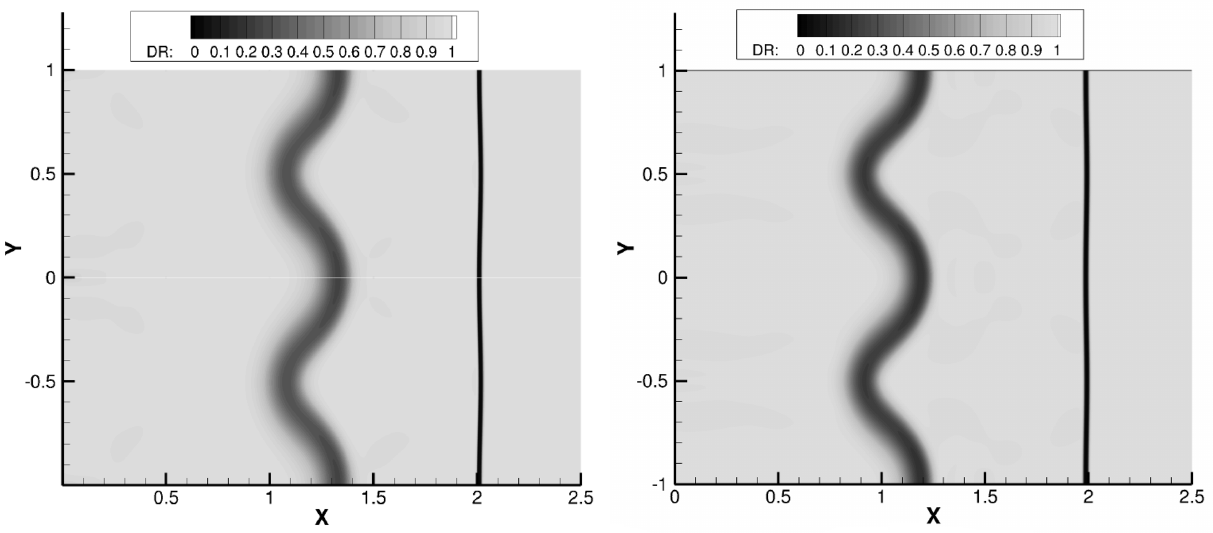
$t = 3.6$

**Figure 7.** Richtmyer–Meshkov instability development in kinetic simulation at $\mathrm{M}_s = 1.5$, $\rho_2/\rho_1 = 2$, $\mathrm{Re} = 400$, $\mathrm{Kn} = 0.041$. Density gradient flowfields

As a result of the interaction with the passing shock wave, the contact surface begins to move in the same direction. Simultaneously, the amplitude of the contact surface perturbation, which is measured as a distance $\Delta x_c$ between developing "bubbles" and "spikes", starts growing. At high Reynolds numbers, the secondary KH instability arises on the distorted contact surface and the lighter and heavier gases are effectively mixed by emerging small-scale pulsations.

The results of kinetic simulations are in close agreement with each other. It can be seen in Fig. 8 where numerical flowfields at the same time moment are compared. The largest deviation of the molecular distribution function in the kinetic simulations from the equilibrium function is predictably observed inside the shock wave.

The instability grows significantly faster at higher Mach numbers of the incident shock wave. A higher density ratio across the contact surface also promotes the development of the RM instability. As can be expected, the growth of perturbations is suppressed by rarefaction effects. At lower Reynolds numbers (larger Knudsen numbers) the interface between two gases is smeared and the secondary instability does not develop. Moreover, at the lowest of the Reynolds



(a) Continuum simulation

(b) Kinetic simulation

**Figure 8.** Density gradient flowfields in continuum (a) and kinetic (b) simulations at $\mathrm{M}_s = 1.5$, $\rho_2/\rho_1 = 2$, $\mathrm{Re} = 400$, $\mathrm{Kn} = 0.041$

numbers studied, Re = 50, the distance between spikes and bubbles stops growing after some time, see Fig. 9.

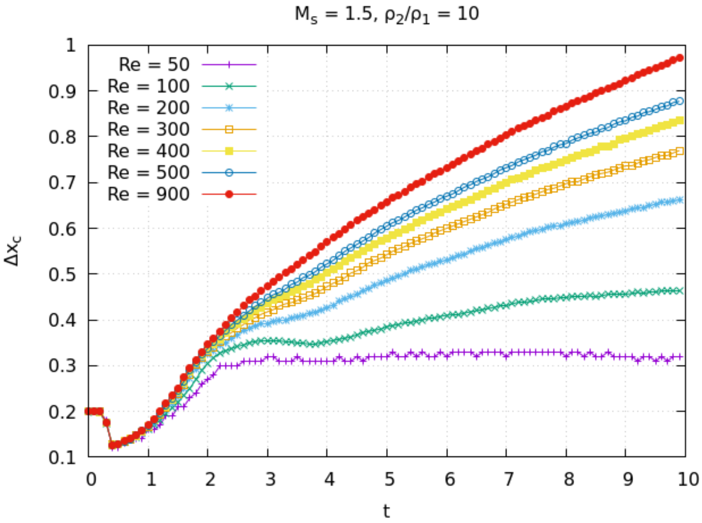


**Figure 9.** Growth of amplitude of contact surface perturbation at various Reynolds numbers. $\mathrm{M}_s = 1.5$, $\rho_2/\rho_1 = 10$

## Conclusion

Statistical and deterministic numerical methods for solving the kinetic equations currently reached a level, at which it is possible to use them for numerical simulation of complex unsteady flows associated with the emergence of hydrodynamic instabilities. The numerical simulations can be performed now using a typical laboratory-level computational cluster, especially if it is equipped with modern GPU hardware. In the presented results of computations based on the kinetic approach the development of instabilities emerging in free shear flows or induced by external body forces has been successfully reproduced. Good agreement is observed between kinetic and continuum approaches when continuum simulations are also performed. The most interesting results are obtained using the DSMC method, because the statistical fluctuations inherent to the method lead to scenarios of the instability development similar to the one observed in experiments.

## Acknowledgements

This work was supported by the Russian Science Foundation, grant No. 23-11-00258.

## References

1. Rapaport, D.C.: Molecular-dynamics study of Rayleigh–Bénard convection. Phys. Rev. Lett. 60(24), 2480–2483 (1988). https://doi.org/10.1103/physrevlett.60.2480

2. Puhl, A., Malek Mansour, M., Marechal, M.: Quantitative comparison of molecular dynamics with hydrodynamics in Rayleigh–Bénard convection. Phys. Rev. A 40(4), 1999–2012 (1989). https://doi.org/10.1103/PhysRevA.40.1999

3. Stefanov, S., Cercignani, C.: Monte Carlo simulation of Bénard's instability in a rarefied gas. Eur. J. Mech. B/Fluids 11(5), 543–554 (1992).

4. Golshtein, E., Elperin, T.: Convective instabilities in rarefied gases by Direct Simulation Monte Carlo method. J. Thermophys. Heat Transfer 10(2), 250–256 (1996). https://doi.org/10.2514/3.782

5. Stefanov, S., Cercignani, C.: Monte Carlo simulation of the Taylor–Couette flow of a rarefied gas. J. Fluid Mech. 256, 199–213 (1993). https://doi.org/10.1017/S0022112093002769

6. Reichelman, D., Nanbu, K.: Monte Carlo direct simulation of the Taylor instability in a rarefied gas. Phys. Fluids A 5, 2585–2587 (1993). https://doi.org/10.1063/1.858775

7. Hirshfeld, D., Rapaport, D.C.: Molecular dynamics simulation of Taylor–Couette vortex formation. Phys. Rev. Lett. 80(24), 5337–5340 (1998). https://doi.org/10.1103/PhysRevLett.80.5337

8. Aristov, V.V., Zabelok, S.A.: Vortex structure of the unstable jet studied on the basis of the Boltzmann equation. Matem. Modelirovanie 13(6), 87–92 (2001, in Russian).

9. Rovenskaya, O.I., Aristov, V.V., Farkhutdinov, T.I.: Invesigation of the Taylor–Görtler instability in jets with help of kinetic approach. Matematika i Matem. Modelirovanie 01, 38–54 (2016, in Russian). https://doi.org/10.7463/mathm.0116.0833621

10. Yang, J.-Y., Chang, J.-W.: Rarefied flow instability simulation using model Boltzmann equations. AIAA Paper No. 97-2017 (1997). https://doi.org/10.2514/6.1997-2017

11. Kadau, K., Germann, T.C., Hadjiconstantinou, N.G., *et al.*: Nanohydrodynamics simulations: an atomistic view of the Rayleigh–Taylor instability. Proc. Nat. Acad. Sci. 101(6), 5851–5855 (2004). https://doi.org/10.1073/pnas.0401228101

12. Kadau, K., Barber, J.L., Germann, T.C, *et al.*: Atomistic methods in fluid simulation. Phil. Trans. R. Soc. A (368), 1547–1560 (2010). https://doi.org/10.1098/rsta.2009.0218

13. Gallis, M.A., Koehler, T.P., Torczynski, J.R., Plimpton, S.J.: Direct simulation Monte Carlo investigation of the Richtmyer–Meshkov instability. Phys. Fluids 27(8), 084105 (2015). https://doi.org/10.1063/1.4928338

14. Gallis, M.A., Koehler, T.P., Torczynski, J.R., Plimpton, S.J.: Direct simulation Monte Carlo investigation of the Rayleigh–Taylor instability. Phys. Rev. Fluids 1(4), 043403 (2016). https://doi.org/10.1103/PhysRevFluids.1.043403

15. Kudryavtsev, A.N., Kashkovsky, A.V., Shershnev, A.A.: Investigation of Kelvin–Helmholtz instability with the DSMC method. AIP Conf. Proc. 2125, 030028 (2019). https://doi.org/10.1063/1.5117410

16. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: Numerical simulation of the Rayleigh–Taylor instability in rarefied Ar/He mixture using the Direct Simulation Monte Carlo method. J. Phys. Conf. Ser. 1382, 012154 (2019). https://doi.org/10.1088/1742-6596/1382/1/012154

17. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: Direct Monte Carlo simulation of development of the Richtmyer–Meshkov on the Ar/He interface. J. Phys. Conf. Ser. 1404, 012109 (2019). https://doi.org/10.1088/1742-6596/1404/1/012109

18. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: DSMC simulation of instability of plane supersonic jet in coflow. AIP Conf. Proceedings 2504, 030087 (2023). https://doi.org/10.1063/5.0132264

19. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: Numerical simulation of the Rayleigh–Taylor instability in rarefied mixture of monatomic gases using continuum and kinetic approaches. AIP Conf. Proceedings 2504, 030111 (2023). https://doi.org/10.1063/5.0132266

20. Kudryavtsev, A.N, Epstein, D.B.: Richtmyer–Meshkov instability in a rarefied gas, results of continuum and kinetic numerical simulations. AIP Conf. Proceedings 2125, 020010 (2019). https://doi.org/10.1063/1.5117370

21. Kudryavtsev, A.N., Poleshkin, S.O., Shershnev, A.A.: Numerical simulation of the instability development in a compressible mixing layer using kinetic and continuum approaches. AIP Conf. Proc. 2125, 030033 (2019). https://doi.org/10.1063/1.5117415

22. Poleshkin, S.O., Kudryavtsev, A.N.: Kinetic simulation of the Rayleigh–Taylor instability. AIP Conf. Proc. 2288, 030010 (2020). https://doi.org/10.1063/5.0028881

23. Malkov, E.A., Kudryavtsev, A.N.: Non-stationary Antonov self-gravitating layer. MNRAS 491(3), 3952–3966 (2020). https://doi.org/10.1093/mnras/stz3276

24. Ivanov, M.S., Kashkovsky, A.V., Vashchenkov, P.V., Bondar, Ye.A.: Parallel object-oriented software system for DSMC modeling of high-altitude aerothermodynamic problems. AIP Conf. Proceedings 1333, 211–218 (2011). https://doi.org/10.1063/1.3562650

25. Kashkovsky, A.V.: 3D DSMC computations on a heterogeneous CPU-GPU cluster with a large number of GPUs. AIP Conf. Proceedings 1628(1), 192–198 (2014) https://doi.org/10.1063/1.4902592

26. Malkov, E.A., Bondar, Ye.A., Kokhanchik, A.A., *et al.*: High-accuracy deterministic solution of the Boltzmann equation for the shock wave structure. Shock Waves 25(4), 387–397 (2015). https://doi.org/10.1007/s00193-015-0563-6

27. Malkov, E.A., Poleshkin, S.O., Kudryavtsev, A.N., Shershnev, A.A.: Numerical approach for solving kinetic equations in two-dimensional case on hybrid computational clusters. AIP Conf. Proceedings 1770, 030077 (2016). https://doi.org/10.1063/1.4964019

28. Bhantnagar, P.L., Gross, E.P., Krook, M.: A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys. Rev. 94(3), 511–525 (1954). https://doi.org/10.1103/PhysRev.94.511

29. Holway, L.H.: New statistical models for kinetic theory: methods of construction. Phys. Fluids 9(9), 1658–1673 (1966). http://doi.org/10.1063/1.1761920

30. Shakhov, E.M.: Approximate kinetic equations in rarefied gas theory. Fluid Dynamics 3(1), 112–115 (1968). http://doi.org/10.1007/BF01016254

31. Kudryavtsev, A.N., Shershnev, A.A.: Numerical simulation of microflows using direct solving of kinetic equations with WENO schemes. J. Sci. Comp. 57(1), 42–73 (2013). http://doi.org/10.1007/s10915-013-9694-z

32. Shershnev, A.A., Kudryavtsev, A.N., Kashkovsky, A.V., *et al.*: A numerical code for a wide range of compressible flows on hybrid computational architectures. Supercomputing Frontiers and Innovations 9(4), 85–99 (2022). https://doi.org/10.14529/jsfi220408

33. Ho, C.-M., Huerre, P.: Perturbed free shear layers. Ann. Rev. Fluid Mech. 16, 365–424 (1984). https://doi.org/10.1146/annurev.fl.16.010184.002053

34. Kashkovsky, A.V., Kudryavtsev, A.N., Shershnev, A.A.: Development of compressible mixing layer instability simulated using the Direct Simulation Monte Carlo method. Supercomputing Frontiers and Innovations 11(2), 48–64 (2024). https://doi.org/10.14529/jsfi240204

35. Kelly, R.E.: On the stability of an inviscid shear layer which is periodic in space and time. J. Fluid Mech. 27(4), 657–689 (1967). https://doi.org/10.1017/S0022112067002538

36. Watanabe, D., Maekawa, H.: Transition of supersonic plane jet due to symmetric/antisymmetric unstable modes. J. Turbulence 3, 047 (2002). https://doi.org/10.1088/1468-5248/3/1/047

37. Zhou, Y.: Rayleigh–Taylor and Richtmyer–Meshkov instability induced flow, turbulence, and mixing. I. Phys. Rep. 720–722, 1–136 (2017). https://doi.org/10.1016/j.physrep.2017.07.005

38. Zhou, Y.: Rayleigh–Taylor and Richtmyer–Meshkov instability induced flow, turbulence, and mixing. II. Phys. Rep. 723–725, 1–160 (2017). https://doi.org/10.1016/j.physrep.2017.07.008

39. Brouillette, M.: The Richtmyer–Meshkov instability. Ann. Rev. Fluid Mech. 34, 445–468 (2002). https://doi.org/10.1146/annurev.fluid.34.090101.162238

DOI: 10.14529/jsfi240206

# NOISEtte CFD&CAA Supercomputer Code for Research and Applications

***Ilya V. Abalakin***[1], ***Pavel A. Bakhvalov***[1], ***Vladimir G. Bobkov***[1], ***Alexey P. Duben***[1], ***Andrey V. Gorobets***[1], ***Tatiana K. Kozubskaya***[1], ***Pavel V. Rodionov***[1], ***Natalia S. Zhdanova***[1]

The paper presents an overview of the CFD/CAA code NOISEtte. The code development began in the 2000s. At first it was a research code intended for elaboration of new methods and techniques in CFD and CAA. Nowadays NOISEtte is actively used as a means for solving numerically various applied problems in aviation industry, turbomachinery, helicopter manufacturing, and space rocket engineering. The code operates on mixed-element unstructured meshes, its numerical algorithm is built on higher-accuracy finite-volume methods using quasi-one-dimensional edge-based reconstruction of flow variables. It is well suited for simulating complex turbulent flows, and especially for high-fidelity scale-resolving simulation of non-stationary turbulent flows using novel RANS-LES methods. A remarkable feature of NOISEtte is its original parallel model, which allows computing with high efficiency on modern supercomputers with arbitrary architectures including CPU cores and GPU accelerators.

*Keywords: CFD code, computational fluid dynamics, aeroacoustics, turbulent flows, scale-resolving simulation, hybrid RANS-LES approach, mixed-element mesh, higher-accuracy method, CPU+GPU, MPI+OpenMP+OpenCL.*

## Introduction

The last decade in Russia was characterized by rapid development of software as a means of solving a wide range of applied problems in various industries. In particular, this trend is clearly visible in industries related to high-speed gas-dynamic flows. Among the currently actively developing Computational Fluid Dynamics (CFD) codes we can note LOGOS [36], FlowVision [9, 35], CADFlo [75], SigmaFlow [34], SINF/Flag-S [48], Platform INMOST [45], HyCFS-R [66], Gerbera [76], FlowModellium [55] and some others. These codes have different features and different levels of universality.

The NOISEtte code development began in the 2000s. At first it was a research code used for elaboration of new methods and techniques in CFD and computational aeroacoustics (CAA). The code operates on mixed-element unstructured meshes, its numerical algorithm is built on higher-accuracy finite-volume methods using quasi-one-dimensional edge-based reconstruction of flow variables. It is intended for simulating complex turbulent flows and acoustic fields, both near and far, associated with them.

From the beginning of its creation, the development of NOISEtte was focused on the ability to work efficiently on various computing systems, which, in turn, were also actively developing. The initial version of the code was written in FORTRAN and was intended for computations in single-processor mode. Since then the NOISEtte has changed revolutionary. Its current version is written in C++ using MPI, OpenMP and OpenCL frameworks for parallel implementation on modern CPU-based, GPU-based and hybrid computing systems.

While remaining a research code, nowadays the NOISEtte is actively used for solving numerically various applied problems in aviation industry, turbomachinery, helicopter manufacturing, space rocket engineering, etc. Its specialization is high-fidelity scale-resolving simulations of

[1] Keldysh Institute of Applied Mathematics, RAS, Moscow, Russian Federation

complex non-stationary turbulent flows using novel hybrid RANS-LES methods. In the field of simulating turbulent flows and aeroacoustics, the closest analogue of the NOISEtte within the world of CFD/CAA software is ANSYS [11].

This paper presents for the first time an almost complete overview of mathematical models, numerical methods and computational technologies implemented in the NOISEtte. It also provides representative examples of its use in solving applied problems involving simulations of turbulent flows and assessment of the acoustic fields generated by turbulent flows including when interacting with nearby solid bodies. Special attention is paid to the modular architecture of the code and its efficient heterogeneous parallel model on the basis of which the NOISEtte is being developed.

The paper is organized as follows. Section 1 presents mathematical models and numerical methods implemented in the NOISEtte. Section 2 describes the code architecture and parallel implementation. The simulation techniques the code is equipped with are outlined in Section 3. The last Section 4 presents the simulation results of various validation and industrial-oriented problems. Conclusion summarizes the results of the work.

# 1. Mathematical Models and Numerical Methods

## 1.1. Governing Equations

The basic mathematical model used in the NOISEtte package for calculating the flow of viscous compressible heat-conducting gas is the system of Navier–Stokes equations, written in the form of the laws of mass, momentum and total energy conservation.

To take into account the flow around rotating solid non-deformable bodies, the system of Navier–Stokes equations is considered in a non-inertial rotating frame of reference. The rotation of the axes of this reference frame occurs around a selected fixed axis with a time-independent angular velocity vector $\boldsymbol{\omega}$ with the magnitude equal to the rotation speed of the solid body. With this description, the streamlined solid body remains motionless, and the direction of the external flow changes in time depending on the azimuthal angle $\psi = |\boldsymbol{\omega}|t$.

We introduce the vector of conservative variables to write the system of Navier–Stokes equations in the conservation laws form

$$\boldsymbol{Q} = (\rho,\ \rho\boldsymbol{u},\ E)^{\mathsf{T}},$$

where $\boldsymbol{u} = (u_1, u_2, u_3)^{\mathsf{T}}$ is the velocity vector in absolute frame of reference, $\rho$ is the density, $E = \rho\boldsymbol{u}^2/2 + \rho\varepsilon$ is the total energy, $\varepsilon$ is the specific internal energy, $p = \rho\varepsilon\,(\gamma - 1)$ is the pressure defined by the ideal perfect gas state of equation, the constant $\gamma$ is the specific ratio.

Let us also introduce the linear tangential velocity vector $\boldsymbol{V} = (V_1, V_2, V_3)^{\mathsf{T}} = \boldsymbol{\omega} \times \boldsymbol{r}$ where $\boldsymbol{r}$ is the position vector. Then the system of Navier–Stokes equations in a non-inertial rotating frame of reference can be written in the following vector form [4]:

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \nabla \cdot \left(\mathcal{F}^C(\boldsymbol{Q}) - \mathcal{F}^R(\boldsymbol{Q}) - \mathcal{F}^D(\boldsymbol{Q}, \nabla\boldsymbol{Q})\right) = \boldsymbol{S}(\boldsymbol{Q}, \nabla\boldsymbol{Q})\,. \tag{1}$$

System (1) includes composite vectors $\mathcal{F}^C$, $\mathcal{F}^R$ and $\mathcal{F}^D$, each component of which $\boldsymbol{F}_i^C$, $\boldsymbol{F}_i^R$ and $\boldsymbol{F}_i^D$ in coordinate direction $x_i$ $(i = 1, 2, 3)$ represents the convective transport, the rotation transport and diffusion flux vectors, respectively. Operator $(\nabla\cdot)$ is the divergence operator.

The convective transport and the rotation flux vectors are given as a function of the physical variables $\rho$, $\boldsymbol{u}$, $p$:

$$\begin{aligned} \boldsymbol{F}_i^C(\boldsymbol{Q}) &= (\rho u_i,\ \rho\boldsymbol{u}u_i + p\boldsymbol{e}_i,\ (E+p)\,u_i)^{\mathsf{T}}\,, \\ \boldsymbol{F}_i^R(\boldsymbol{Q}) &= (\rho V_i,\ \rho\boldsymbol{u}V_i,\ EV_i)^{\mathsf{T}}\,, \end{aligned} \tag{2}$$

where $\boldsymbol{e_i} = (\delta_{i1}, \delta_{i2}, \delta_{i3})^{\mathsf{T}}$ is the row-vector of the identity matrix and $\delta_{ij}$ is the Kronecker symbol. The diffusion flux vector is defined as a function of physical variables and their gradients as

$$\boldsymbol{F}_i^D(\boldsymbol{Q}, \nabla\boldsymbol{Q}) = (0,\ \tau_{i1},\ \tau_{i2},\ \tau_{i3},\ \tau_{ij}u_j + q_i)^{\mathsf{T}}\,, \tag{3}$$

where the components of the viscous stress tensor $\tau_{ij}$ and the heat flux vector $q_i$ can be written as follows:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right), \quad q_i = \frac{\mu}{\mathrm{Pr}}\frac{\partial\varepsilon}{\partial x_i}\ , \tag{4}$$

where $\mu$ is the molecular viscosity coefficient, Pr the molecular Prandtl number.

Vector $\boldsymbol{S}(\boldsymbol{Q}, \nabla\boldsymbol{Q})$ is a source term describing the influence of the external forces that are not related to the transfer processes of the target variables $\boldsymbol{Q}$:

$$\boldsymbol{S}(\boldsymbol{Q}, \nabla\boldsymbol{Q}) = (0\,, -\rho\,(\boldsymbol{\omega}\times\boldsymbol{u})\ , 0)^{\mathsf{T}}\,. \tag{5}$$

It should be noted that, from the standpoint of an observer in the stationary frame of reference, the system of equations (1)–(5) describes the evolution of conservative variables due to their transport in the rotating (with velocity $\boldsymbol{V}$) media, the pressure gradient and the velocity vector turn to the angle equal to $|\boldsymbol{\omega}|\,t$ (implemented by the term $-\rho\,(\boldsymbol{\omega}\times\boldsymbol{u})$ in the momentum equation). Note that in the numerical implementation of this system, the rotation velocity can be interpreted as the velocity of moving mesh.

## 1.2. Turbulence Modeling

The numerical algorithm in the NOISEtte code includes both time-averaged and scale-resolving approaches based on the system of Navier–Stokes equations for simulation of turbulent flows. The former considers the Reynolds-Averaged Navier–Stokes (RANS) equations with the Boussinesq eddy viscosity assumption as a closure. Several semi-empirical differential turbulence models introducing up to four additional equations are implemented. They include the Spalart–Allmaras [72] (SA) and the Menter SST [52] (two equations) models. Also, a few SST-based laminar-turbulent (LT) transition models, both differential (the four equations Langtry–Menter $\gamma - \tilde{\mathrm{Re}}_{\theta,t}$ [49] and one-equation $\gamma$ [54]) and algebraic (LCTM [53] and KD-SST [74]) models are realized [51].

The set of sub-grid scale (SGS) models for Large Eddy Simulation (LES) implemented in the NOISEtte includes the Smagorinsky [71] and several modern enhanced LES models: WALE, $\sigma$ models [58]; S3PQR models [78].

We mostly use hybrid RANS-LES methods (HRLM) for scale resolving simulations in practice, thanks to optimal combination of accuracy and reduced computational cost [39]. Recent versions [43, 69], along with the original ones, of the Detached Eddy Simulation (DES) are realized. They are equipped by grey-area mitigation (GAM) [56] techniques for accelerating RANS-to-LES transition in shear layers. The GAM techniques are based on combinations of dynamic adapting sub-grid scales and/or alternative LES models (e.g., $\sigma$ [58] or a S3PQR model [78]). A few enhanced dynamic sub-grid scale models are implemented: $\tilde{\Delta}_\omega$, $\Delta_{\mathrm{SLA}}$ [56, 69]; $\Delta_{\mathrm{lsq}}$ [62, 79]. All sub-grid scale models, both static and dynamic, are realized [29] in accordance with the edge-based methodology utilized in the NOISEtte . Also, a DES version with enhanced boundary layer protection, namely, ZDES mode 2 EP [23], is implemented and being used for scale-resolving simulation of turbulent flows.

Scale-resolving simulation often demands plausible turbulent pulsations upstream sensitive regions of configurations under investigation, especially when attached or mildly separated flows are considered. We impose artificial turbulence pulsations for these cases using the synthetic turbulence generator (STG) [68] either on inlet boundaries or in a form of a distributed volume source [67] (VSTG).

### 1.3. Higher-Accuracy Reconstruction-Based Methods

NOISEtte is a vertex-centered code on unstructured meshes. A 2D mesh may consist of triangles and quadrilaterals. In 3D, tetrahedra, quadrilateral pyramids, triangular prisms, and hexahedra mesh elements are allowed. A quadrilateral face of a 3D element is not assumed to be planar.

A numerical solution is prescribed by the set of conservative variables at mesh nodes. For the discretization of the convective terms, the family of edge-based reconstruction (EBR) finite-volume schemes is used. Initially developed for smooth problems [2], they were generalized to shock-capturing methods [13] and adapted for high-Reynolds number flows [14].

Like in other high-accuracy finite-volume schemes, the numerical fluxes are defined by a Riemann solver applied to reconstructed values. The main feature of EBR schemes is the quasi-one-dimensional reconstruction. Its computational costs are much lower compared to finite-volume methods with the polynomial reconstruction. On unstructured meshes, the EBR schemes are at most 1-exact. However, on uniform grid-like meshes (both hexahedral, prismatic, and tetrahedral) they reduce to high-order finite-difference methods. For problems involving discontinuities or shocks, WENO approach is adapted to the EBR paradigm, as well as many classical slope limiters. The following numerical fluxes are implemented: Local Lax–Friedrichs (or Rusanov) [50], HLLE, HLLC [77], Roe [64], Roe with a low-Mach preconditioner [42].

Scale-resolving simulation of turbulent flows using HRLM methods (e.g., DES) is effective and resilient when a hybrid scheme for convective fluxes is applied. It is needed to maintain both stability of the solution and low level of numerical dissipation in LES regions. The hybrid scheme uses the adapting blend of the central-difference (CD) and upwind schemes based on an extended numerical stencil with the special hybridizing function [44]. The hybrid version of EBR scheme [28], along with CD and upwind, can involve local flow-dependent WENO-reconstructions when subsonic and supersonic flows with shocks are considered.

For the discretization of the diffusion terms, the standard P1-Galerkin method and the method of averaged element splittings (AES) are used [15]. On simplicial meshes, they coincide. On Cartesian hexahedral meshes, AES yields the 7-point approximation of the Laplace operator (the P1-Galerkin method yields the 27-point approximation). On non-Cartesian hexahedral meshes, both methods yield the 27-point approximation. However, the difference between them is revealed when using implicit time integration for high-Reynolds number flows. For AES, we keep only edge-connected nodes in the portrait of the flux Jacobian. This significantly reduces the memory storage and the computational costs with marginal effects on the timestep size and on the stability of the computation. Our numerical experiments show that this is not possible for the P1-Galerkin method [15].

For the time integration, backward differentiation formulas BDF1 and BDF2 are used. To solve nonlinear algebraic systems, a simplified Newton method is used, and the linear systems are solved using the BiCGStab solver [65]. Explicit Runge–Kutta schemes (up to 4th order) are implemented as well.

## 2. Software Architecture and Parallel Computing

The code is written in C++ using MPI, OpenMP and OpenCL frameworks for parallel implementation. It consists of a core computational library and connectable functional modules that are linked to console applications for preprocessing, running simulations, and postprocessing results (Fig. 1). The code is designed for maximum portability and can be used on Windows and Linux, on a wide range of computing systems from workstations to hybrid supercomputers. It has been tested on various computing architectures, including multicore CPUs (Intel, AMD, IBM, ARM, Elbrus), manycore accelerators (Intel Xeon Phi KNC, KNL), GPUs (Intel, AMD, NVIDIA), systems on a chip combining central and graphics processors.

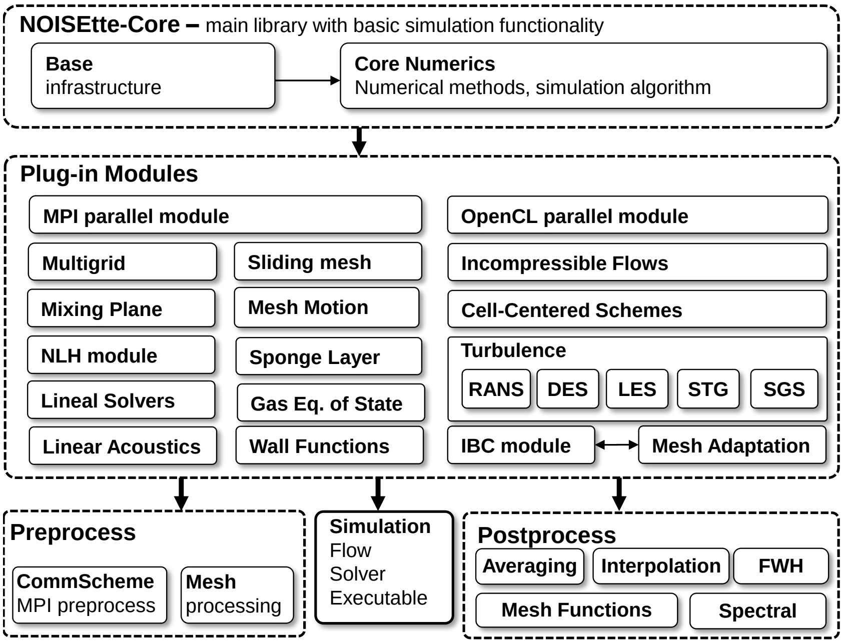


**Figure 1.** Code structure diagram

In order for the code to be used as a research platform for development of new numerical methods and models, it is important that new functionality can be easily added. Therefore, most components of numerical technology such as reconstruction of variables, Riemann solvers, viscous terms, turbulence models, source terms, etc., allow extending the set of implemented options or replacement with an external implementation. On the other hand, with such a modular architecture, redundant functionality can be easily removed from the assembly, which allows us to obtain configurable problem-specific solutions.

Since the NOISEtte code is designed primarily for high-fidelity, resource-intensive supercomputer simulations, the efficient use of modern hybrid high-performance computing systems is crucial. For this purpose, the computational algorithm is based on hierarchical multilevel parallelization. At the top level, the MPI standard is used to couple multiple nodes of a cluster system. Then, second-level mesh partitioning is used to further distribute the workload among computing devices of hybrid cluster nodes, such as CPUs and GPUs. MPI parallelization uses asynchronous, non-blocking exchanges and allows communications to be hidden behind computations to improve parallel efficiency, which is especially important for GPU computing. To reduce the data transfer overhead, multi-threaded message processing is used as well.

At the next level, the OpenMP standard is used for multi-core central processors and many-core accelerators. OpenMP parallelization is also based on multi-level mesh partitioning: subdomains of computing devices are further decomposed among parallel threads. To eliminate race conditions between threads, the interface zones between threads' subdomains are separated and moved to the next level, where a similar partitioning is applied.

Finally, the OpenCL open standard is used for massively parallel accelerators such as GPUs (and can be used for CPUs as well). The implementation supports an automatic testing procedure that ensures consistency of each kernel in use for a particular case, overlapping exchanges and computations, and flexible configuration for different types of devices.

More information about our parallel technology can be found in [39]. The heterogeneous parallel algorithm and its implementation are described in detail in [38], where the parallel efficiency and performance on various supercomputers is presented.

The parallel capabilities can be summarized as follows. On CPU-based supercomputers, at least several tens of thousands of cores can be efficiently used, provided that there are no less than about 5–10 thousand mesh nodes per core. On GPU-based supercomputers, at least dozens of GPUs can be used (perhaps, hundreds, but we have never tested that much), provided that there are at least 0.5–1 million mesh nodes per device, depending on a model. The practical equivalent of performance obtained from a modern GPU compared to a modern server processor is around 100–200 cores. For a large number of devices, this can be around 300 CPU cores per GPU device due to better parallel efficiency at the MPI level. For instance, 36 NVIDIA V100 GPUs perform as fast as about 10 thousand CPU cores (dual-CPU nodes with 24-core Intel Xeon Platinum) on a mesh of about 80 million nodes.

## 3. Simulation Technologies

### 3.1. Multigrid Convergence Accelerator

Although the NOISEtte code is focused on time-accurate high-fidelity simulations, it can also be used in stationary RANS simulations in industrial applications. For this purpose, the code is equipped with a full approximation scheme multigrid (FAS MG) convergence accelerator [40]. The FAS MG speeds up RANS simulations by an order of magnitude by using 2–3 mesh levels created by uniformly refining the coarse mesh and smoothing the refined ones [41]. In Fig. 2, an example of RANS simulation acceleration is shown. It shows comparison of runtime on a single mesh of about 20 million cells with a multigrid-accelerated simulation with 3 mesh levels. The test case is a multistage axial compressor, the simulation is running on a single 32-core CPU AMD EPYC 7542. The resulting acceleration is about 15 times. It should be noted that in this case the NOISEtte code has outperformed the Numeca FINE/Turbo problem-oriented commercial code (used as a reference) by a factor of ×1.7.

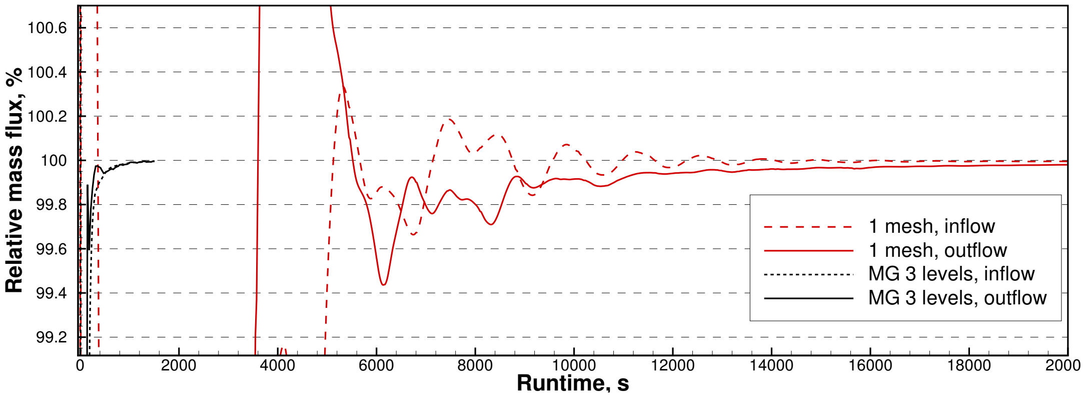


**Figure 2.** Multigrid convergence acceleration example: input and output fluxes (in relative values compared to the reference value) of a multistage axial compressor on a mesh of about 20 million nodes

## 3.2. Dedicated Turbomachinery Techniques

Turbomachines (compressors and turbines) are characterized by the presence of many closely located blade rows which rotate relative to each other. So, the flow in turbomachines is fundamentally unsteady in nature and characterized by many interacting harmonics. The distance between adjacent rows is usually very low (about 5–50% of the blade chord), so their interaction can be significant. However, the main aerodynamic characteristics of turbomachine units (mass flow rate, total pressure ratio, efficiency) can be obtained considering only the steady component of the rows interaction. The Mixing Plane (MP) technology [24] for matching the flows on (rotor-stator) interfaces between rows was designed to meet these objectives. It assumes the absence of circumferential flow non-uniformity in one of the matching rows when determining characteristics in the other, and vice versa. The use of the MP significantly reduces computational cost of turbomachines' simulations since it allows using only one vane channel (the vane wheel periodicity sector) per row for computation. The MP technique is implemented in the NOISEtte and successfully validated [26] on a set of relevant turbomachinery problems. It ensures conservation and low-reflection properties. The MP approach is realized within the parallel framework of the NOISEtte supercomputer simulation code.

## 3.3. Sliding Meshes

The sliding meshes approach is used for the direct simulation of the governing equations when two or more mesh parts move relative to each other.

In our implementation [5], each interface between these parts is either a planar solid of revolution ("base") or a lateral surface of a finite circular cylinder ("side"). These interfaces appear naturally for axial and centrifugal fans, correspondingly. In the first case, an unstructured triangular mesh should be used on the interface. In the latter case, the interface should be meshes by a Cartesian (in cylindrical coordinates) generally non-uniform mesh.

The interfaces split the computational domain into several parts, and they do not change in time. Each part is represented as a union of control volumes, which do not deform. In case of the "side" interface, a control volume near a circular cylinder is no longer a polyhedron but a curvilinear object. Control volumes never overlap and their shape is well-defined.

A sliding interface introduces a strong irregularity to the mesh, so a linear reconstruction is a good trade-off between the accuracy and robustness. The use of a slope limiter is recommended even for shock-free problems.

### 3.4. Wall Functions

Although RANS, hybrid RANS-LES, and DES approaches significantly reduce the near-wall mesh resolution requirements as compared with the direct numerical simulation, the cell size restrictions remain severe. Mesh resolution restrictions in the near-wall region can be considerably reduced by applying wall function methods. To provide near wall accuracy a few types of wall function methods are realized in the NOISEtte code.

The first one is a traditional approach, where the no-slip boundary conditions on the wall is replaced by a matching condition between the outer turbulent boundary layer and the wall function solutions at the exchange locations at the mesh points nearest to the wall. In this case, the exchange location, expressed in terms of normalized coordinate, varies from node to node and depends on a first near wall cell size [25]. It demonstrates high efficiency in a case of zero pressure gradient flows and also is applicable for flows with the adverse pressure gradients but with some restrictions.

An alternative wall function formulation in terms of a differential wall-stress boundary condition admitting transfer of shear stress from the outer region of the boundary layer to the wall has been recently proposed and implemented in the code. The developed penalized wall function (PWF) method completely eliminates the need to explicitly determine the position of the exchange location, as well as to interpolate the solution to this location [80, 84]. A characteristic-based volume penalization method is used to transfer the friction velocity to the wall, and the exchange location is specified implicitly by a localized source term in the boundary layer equation that is written as a function of the normalized distance to the wall. The wall shear stress, in turn, is determined by solving an auxiliary equation for the wall stress imposing the analytical wall function solution through the characteristic-based volume penalization method. This approach reduces the system of differential equations with nonlinear algebraic constraints for the matching condition to a system of equations with a differential feedback loop provided by characteristic penalization functions. The PWF method is successfully generalized to flows with separation [81].

The provided methods are applicable with Spalart–Allmaras and $k - \omega$ RANS turbulence models.

### 3.5. Immersed Boundary Method

The immersed boundary method (IBM) makes it possible to avoid the cost and difficulties related to the construction of meshes and set boundary conditions on the surface of solid bodies without positioning mesh nodes on the boundary of the obstacles, which greatly simplifies the construction of the computational mesh, which is then solved in the entire domain of the problem definition, including the rigid body. In the NOISEtte code the influence of an obstacle on the flow could be mimicked by the Brinkman volume penalization (BP) method [8, 83] or Characteristic-Based Volume Penalization (CBVP) method [3, 7]. These methods correspond to a separate subclass of IBM, in which the effect of the presence of an obstacle is modelled by introducing additional terms in differential equations that describe the evolution of a liquid or gas flow, after which the modified equations are discretized and solved using an appropriate computational method.

In the BP method the obstacle is modelled as a porous medium with low permeability [10]. The formulation utilizes Brinkman-type penalization terms applied to the momentum and energy equations inside of the obstacle. One of advantages of the BP method is the ability to control the error through the penalization parameter with the proven convergence of the solution of the penalized Navier–Stokes equations to the exact solution in the limit when the penalization parameter tends to zero. The BP method, despite its ability to rigorously control the error of the solution, is limited to problems with Dirichlet boundary conditions.

The CBVP method [20] exploits the hyperbolicity of characteristic-based forcing terms to impose general homogeneous and inhomogeneous Neumann and Robin boundary conditions. The penalized Navier–Stokes equations are solved to simulate flow around obstacles. In the solid region the hyperbolic penalization equations are solved to propagate the solution from the surface along the inward-pointing characteristics that enforce the desired value of derivative with an a priori defined accuracy. The CBVP method maintains rigorous control of the error through a priori chosen parameters for all type of boundary conditions.

The aforementioned methods could be used for modeling of flow in the presence of obstacles moving under influence of external or induced forces [83].

### 3.6. Far Field Acoustics

We use the Lighthill acoustic analogy in the form of a modified version of the integral Ffowcs Williams and Hawkings (FWH) method [33] in terms of retarded times based on formulation 1A as proposed by Farassat [31] to predict far-field acoustics. The technology considers accumulation of the data for further acoustical postprocessing on control surfaces excluding the "quadrupole" volume terms. Control surfaces could either coincide with solid boundaries (solid control surfaces) or be located in the flow (permeable control surfaces). The latter is mostly used in practice. We apply additional techniques which enhance the results, e.g., reduce spurious non-physical noise. They involve usage of multiple nested permeable control surfaces with consequent averaging the results over closing surfaces located downstream [70] and/or lateral surfaces [46]. Also, the density-by-pressure substitution assuming the isentropic relations [73] may be applied. The program implementation of the acoustical postprocessor has combined MPI+OpenMP parallelization.

The case of a permeable control surface near a rotor requires a special treatment. If the control surface is moving with the rotor, then the velocity of its points may approach or exceed the sound speed, especially if the background flow is present. In this case the formulation 1A fails, and a much more complex formulation (for instance, the emission-surface formulation [19]) is required. In order to avoid these difficulties, to predict the rotor noise, we use axial symmetric control surfaces. Then the motion of the control surface reduces to a straight movement, which is usually subsonic. The rotation of the control surface is replaced by a simple 1D interpolation in the angular variable [12].

### 3.7. Arbitrary Lagrangian-Eulerian Approach

The NOISEtte package implements the Arbitrary-Lagrangian-Eulerian approach (ALE) to constructing schemes for the moving hybrid [16]. The movement of the mesh involves recomputing the new position of all the nodes and rebuilding the finite (control) volumes at each time step.

The basic equation defining the finite-volume ALE scheme was obtained by integrating the equation (1) over a moving arbitrary control volume $C(t)$ using the Gauss theorem for the integral of the divergence of the transport fluxes and then applying the Reynolds transport theorem to the integral from the partial derivative with respect to time.

Let $C(t)$ be a computational cell on a given mesh with volume $|C_i(t)|$ and $\bar{\boldsymbol{Q}}_i$ the integral average of $\boldsymbol{Q}(t)$ over this cell. Then

$$\frac{d}{dt}\int\limits_{C_i(t)} \boldsymbol{Q}\,dV = \frac{d\bar{\boldsymbol{Q}}_i|C_i(t)|}{dt}, \qquad |C_i(t)| = \int\limits_{C_i(t)} dV, \qquad \bar{\boldsymbol{Q}}_i = \frac{1}{|C_i(t)|}\int\limits_{C_i(t)} \boldsymbol{Q}\,dV$$

and the basic equation of the ALE method can be expressed as follows

$$\begin{aligned}\frac{d\bar{\boldsymbol{Q}}_i|C_i(t)|}{dt} &+ \int\limits_{\partial C_i(t)} \mathcal{F}^C(\boldsymbol{Q})\cdot\boldsymbol{n}\,dS - \int\limits_{\partial C_i(t)} \boldsymbol{Q}\,(\boldsymbol{V}+\boldsymbol{V}_c)\cdot\boldsymbol{n}\,dS = \\ &= \int\limits_{C_i(t)} \mathcal{F}^D(\boldsymbol{Q},\nabla\boldsymbol{Q})\,dV + \int\limits_{C_i(t)} \boldsymbol{S}(\boldsymbol{Q},\nabla\boldsymbol{Q})\,dV,\end{aligned}$$

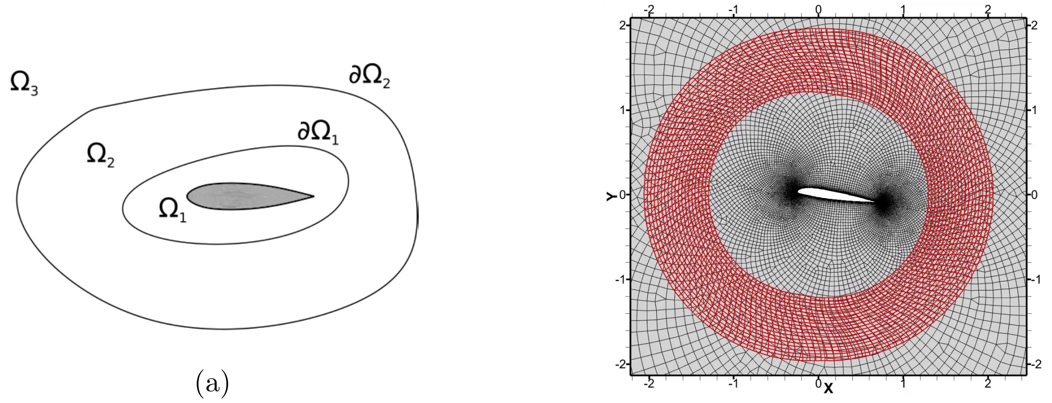
where $\partial C_i(t)$ is the boundary of cell, $n$ is the unit external normal to the boundary $\partial C_i(t)$, $\boldsymbol{V}_c$ is velocity of the boundary of the moving cell.

An important requirement for an ALE scheme is the fulfillment of the geometric conservation law (GCL). The article [16] provides conditions for executing GCL for the explicit Runge–Kutta methods and for the implicit schemes of the first and second order. The condition for the GCL property to hold for an implicit first-order scheme is similar to the condition from the article [59], where it is provided only for a simplicial mesh.

### 3.8. Elastic Mesh for Cyclic Pitch Control and Other Small Displacements

In the NOISEtte code the method of deformation of an unstructured hybrid mesh for simulating the flow near solid bodies performing small movements (according to external laws or under the action of aerodynamic forces) is implemented. The proposed method is based on the use of an auxiliary strand mesh, “tied” to a limited area of the computational mesh. Along each strand, a one-dimensional compression-tension problem is solved to ensure smooth resizing of mesh elements. Within the approach the computational mesh is divided into three disjoint subdomains $\Omega_1$, $\Omega_2$, and $\Omega_3$ (Fig. 3a).

Domain $\Omega_1$ contains a moving object, the mesh nodes included in $\Omega_1$ move with this solid body, and the mesh elements do not deform. Domain $\Omega_1$, as a rule, contains a boundary layer resolution zone, the deformation of which can affect the quality of the mesh and, accordingly, the calculation as a whole. Domain $\Omega_3$ contains all nodes located far from the surface of the body that must remain motionless. Domain $\Omega_2$ lying between them is the deformation region, in which the nodes of the computational mesh of an arbitrary structure are redistributed in accordance with the law of the bodys motion so that at small displacements the mesh retains its original topology and the quality of the elements and ensures a smooth change in the size of the cells between regions $\Omega_1$ and $\Omega_3$.



(a)

(b)

**Figure 3.** Deformation method: deformation zones (a) and mesh deformation example (b)

The efficiency of the method is demonstrated by solving two-dimensional and three-dimensional problems of simulating the flow around isolated moving bodies of simple configuration [17].

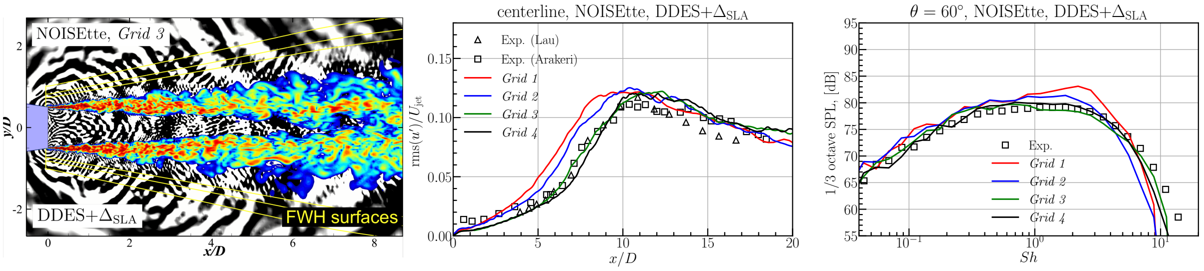
The proposed method allows to simulate turbulent flow near moving solid bodies and applicable for modelling a wide range of problems such as oscillating airfoils and wings, hinged rotor blades movement with pitch control, etc.

# 4. Applications

## 4.1. Jet Flows

The NOISEtte scale-resolving algorithm has been successfully applied for simulations of turbulent jets. Both near-field aerodynamics and far-field aeroacoustics were evaluated during the investigations. The latter is strongly sensitive to resolving and dissipative features of a numerical algorithm in use.

The subsonic ($M_{\text{jet}} = 0.9$) immersed round jet was considered in [28, 29, 62], where different sides of scale-resolving simulation of jets were evaluated, including numerical schemes and GAM properties of the HRLM approaches in use. The scale-resolving algorithm realized in NOISEtte has demonstrated a possibility to provide sufficient accuracy on rather coarse meshes. The grid convergence was demonstrated in [29] for the set of meshes containing from 1.5 to 33 million nodes. Figure 4 demonstrates the results obtained using DDES+$\Delta_{\text{SLA}}$.



(a)

(b)

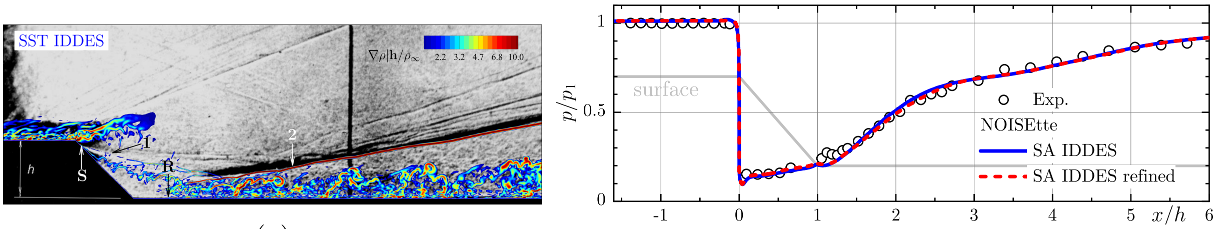
(c)

**Figure 4.** Subsonic turbulent jet: flow visualization (a) and mesh convergence results for the DDES+$\Delta_{\text{SLA}}$, both aerodynamics (b) and aeroacoustics (c)

Turbulent round jets with shocks were also simulated using the NOISEtte. The underexpanded hot round jet was considered in [28]. The far-field noise results obtained on a rather coarse mesh (about 4.5M nodes) are in a good agreement with the reference data (experimental measurements and the computation using a structured research code). The paper [27] presents the results of near-field characteristics of the dual-stream jet (slightly underexpanded supersonic at the bypass duct and subsonic at the main duct). They are close to the experimental values as well.

## 4.2. Transonic Flows

The NOISEtte scale-resolving algorithm has been successfully applied for simulations of transonic turbulent flows. They include simple scientific configurations and complex industry-oriented turbulent flows. Simulations of the M219 cavity were used [22] for analysis of self-oscillation processes in open cavities. The supersonic flow over the inclined back-facing step (BFS45) [27] was successfully simulated using the NOISEtte scale-resolving algorithm. Figure 5 demonstrates the results of computations validated using the experimental data available for this benchmark test case. The meshes containing 15M and 108M (denoted as "refined") were used for simulations.



(a)

(b)

**Figure 5.** BFS45: the instantaneous field of density gradient magnitude ("numerical" shchlieren) superimposed on the experimental photo (a) and averaged static pressure distribution over the surface (b)

The NOISEtte was used to solve industry-oriented problems considering simulation of transonic turbulent flows. Among variety of studied flow characteristics, pressure loads on solid surfaces of a space rocket under different flow regimes were investigated (see Fig. 6).

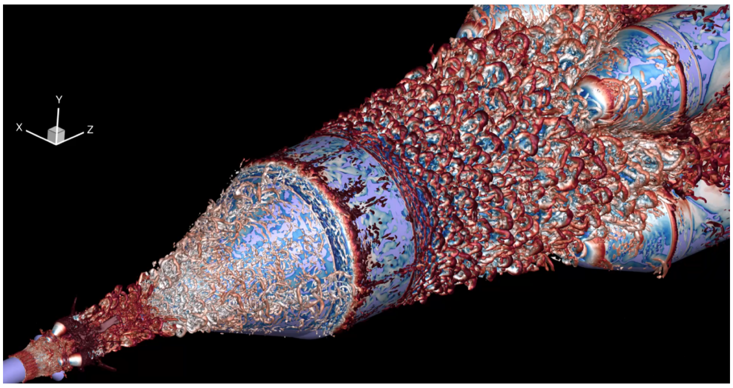
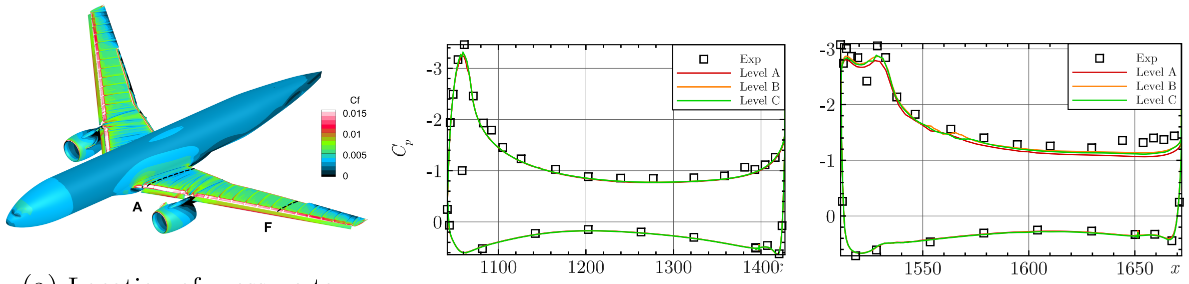


**Figure 6.** Space rocket: flow visualizations (isosurfaces of $Q$ criterion colored by Mach number levels)

### 4.3. Airframe Flows

To assess the ability of the NOISEtte code to reproduce aerodynamics of an airframe during landing, we use the test case of the 4-th AIAA CFD High-Lift Prediction Workshop (HLPW4) [1]. This test case is based on the experimental data obtained in QinetiQ Low-Speed Wind Tunnel for the 10%-scale NASA Common Research Model in High-Lift configuration (CRM-HL) [30]. Surface distributions of pressure coefficient obtained by RANS SA simulations on common workshop meshes demonstrate decent agreement with the corresponding experimental values (Fig. 7).



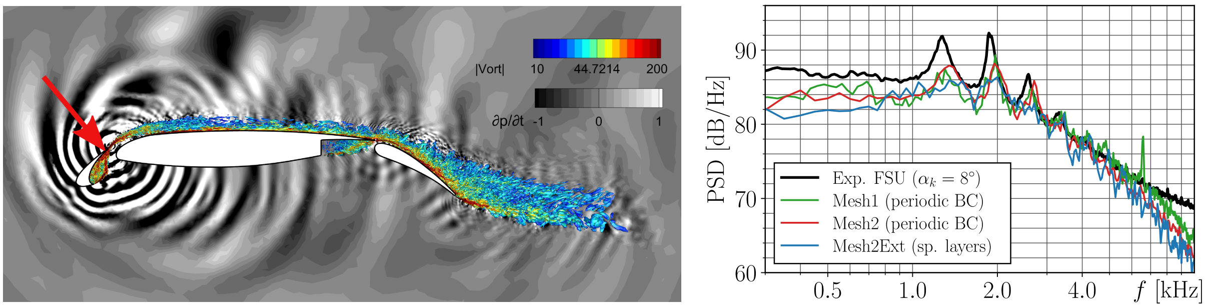
(a) Location of pressure tap rows

(b) Row A

(c) Row F

**Figure 7.** HLPW4 test case: simulation results (M = 0.2, Re = $5.49 \times 10^6$, $\alpha_{\text{corr}} = 7.05°$)

We use the well-known 30P30N validation case [57, 60, 61] to test our code on solving airframe noise problems [37]. The considered geometry is unswept wing segment based on the three-element 30P30N airfoil in high-lift configuration (Fig. 8). We perform scale-resolving simulation using IDDES method and accumulate history of pressure pulsations in the points defined by experiment setup. Comparison between corresponding numerical and experimental spectra shows their reasonable agreement.

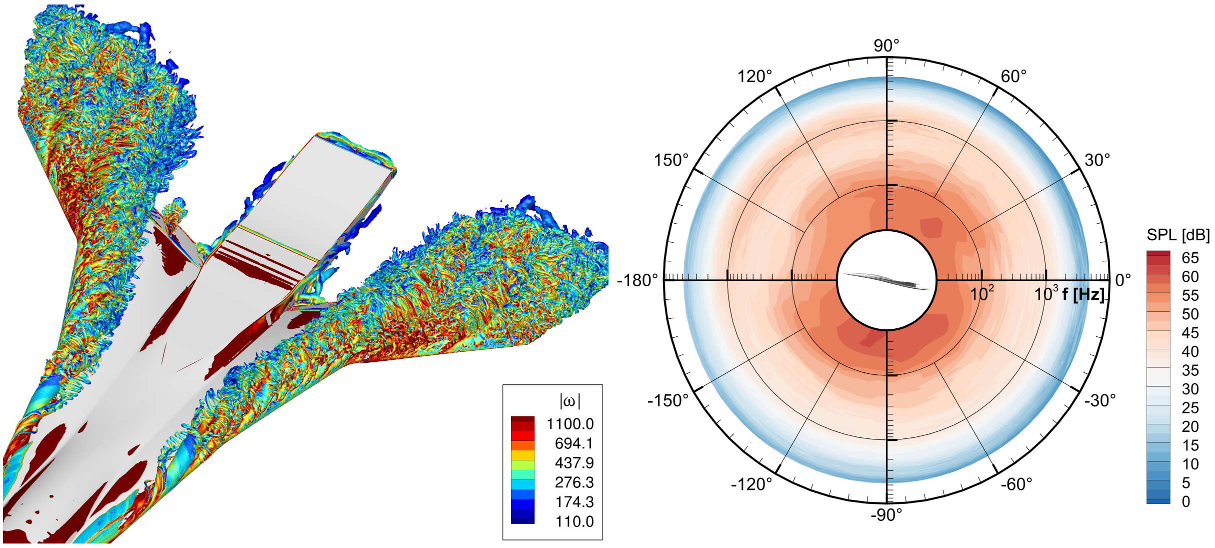


(a) Instantaneous flow field

(b) Spectra at Point 6 (red arrow)

**Figure 8.** 30P30N test case: simulation results (M = 0.17, Re = $1.7 \times 10^6$, $\alpha = 5.5°$)

After testing the NOISEtte code on validation cases related to aerodynamics and aeroacoustics of an airframe, we apply it for numerical assessment of noise generated by the wing of supersonic business jet (SSBJ) during landing. Example of analogous study can be found in [32, 47, 63]. We modelled the flow around half of full-scale SSBJ airframe prototype in high-lift configuration using DDES method on hybrid unstructured mesh contained 62 million nodes and 220 million cells with the zone of increased mesh resolution above the wing surface (Fig. 9). During computation data is stored on FWH surface and at some near-field points to monitor simulation state and to localize preliminarily the most prominent noise sources. When required amount of data is accumulated, the FWH method is applied to obtain noise properties in the far-field. To perform this simulation (including data accumulation stage), 24 GPUs were used for 3–4 days of pure computational time.



(a) Instantaneous flow field (iso-surfaces of Q-criterion = 5000 [1/s$^2$])

(b) Far-field noise spectra on circle with radius of 150 m

**Figure 9.** SSBJ case: simulation results (M = 0.2, $\mathrm{Re}_{1\mathrm{m}} = 4.6 \times 10^6$, $\alpha = 10°$)

## 4.4. Helicopter and Drone Rotors

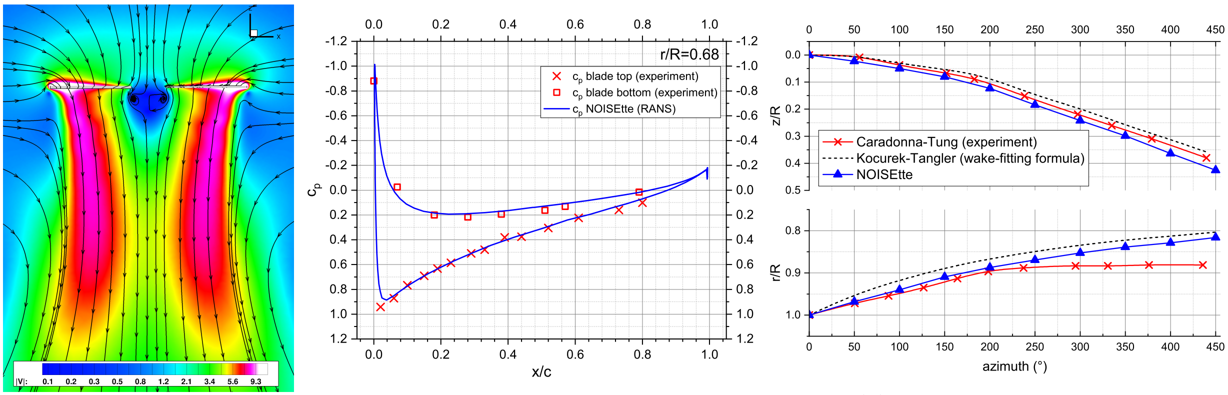
The methods described above were applied for simulating the turbulent flow around rotorcraft rotors and calculating its aerodynamic and acoustic characteristics.

Validation of the developed technique is carried out by simulating the turbulent flow around the Caradonna-Tung rotor, the KNRTU-KAI four-blade model rotor in hover mode, tail rotor in duct, rigid main rotor in forward flight regime, and the turbulent flow and far-field acoustics of unmanned aerial vehicle (UAV) rotor. The numerical results are compared with the available experimental data.

The first case was modelling of flow near the two-bladed Caradonna-Tung rotor [21]. The RANS simulation results are in good agreement with experiment: the overall flow field (Fig. 10a), the pressure coefficient distribution (Fig. 10b) and tip vortex evolution (Fig. 10c) are in good agreement with the experiment.

In the second case the experiment performed in the Kazan Aviation Institute (KAI) was reproduced in the numerical experiment. The near flow around scaled model of the four-bladed rotor was modelled where the acoustics in near field was measured in the set of probes (Fig. 11a). The pressure pulsation predicted in the numerical simulation fits in the experimental one taking into account their scatter (Fig. 11b, 11c).

The third case was devoted to the numerical simulation of flow around the rigid helicopter main rotor in forward flight [6]. The numerically obtained aerodynamic characteristics of the main rotor were compared with the data of physical experiment. The locations of the cores of the tip vortices repeat the trajectories of the blade tip motions with account for rotor rotation and the oncoming flow. Such tip vortices behavior was successfully reproduced within the numerical experiment. In Fig. 12a the trajectories of the tip vortices visualized by iso-surfaces of Q-criterion.

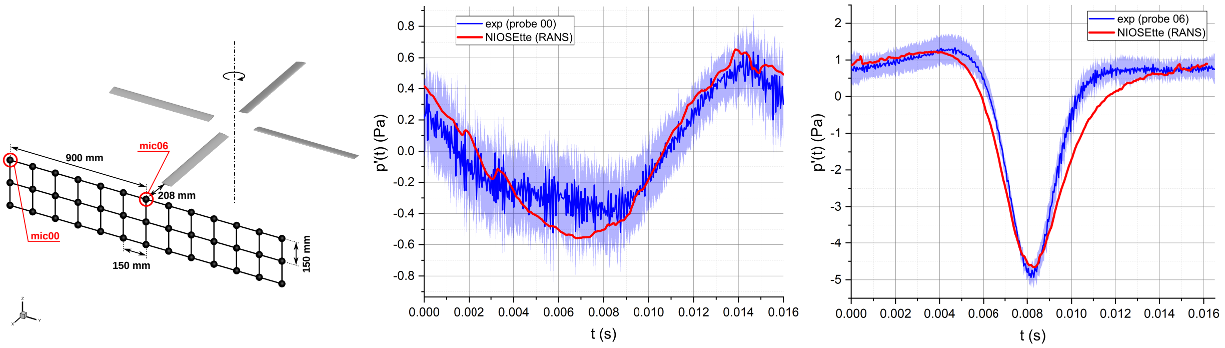


(a) Flow field

(b) Pressure coefficient distribution

(c) Tip vortex evolution

**Figure 10.** Caradonna-Tung rotor simulation results



(a) Probes setup

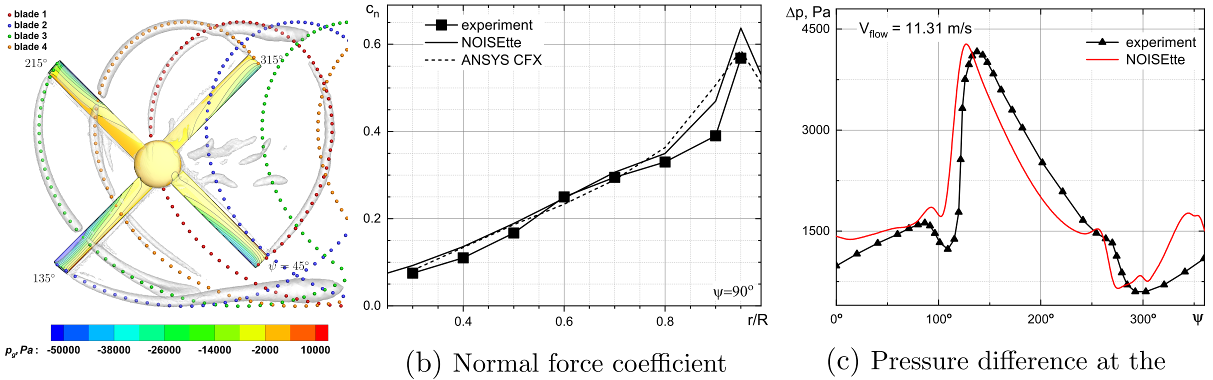
(b) Pressure pulsations, mic00

(c) Pressure pulsations, mic06

**Figure 11.** KAI rotor near-field acoustics simulation results

The normal force coefficient profile along the blade span for azimuth 90° is shown in Fig. 12b. It fits well both with experiment data and numerical simulation performed using commercial CFD package.

The comparison of the pressure difference measured at two characteristic points on the leading edge of the blade were considered. It shows that the values predicted in the numerical experiment are rather accurate (see Fig. 12c).



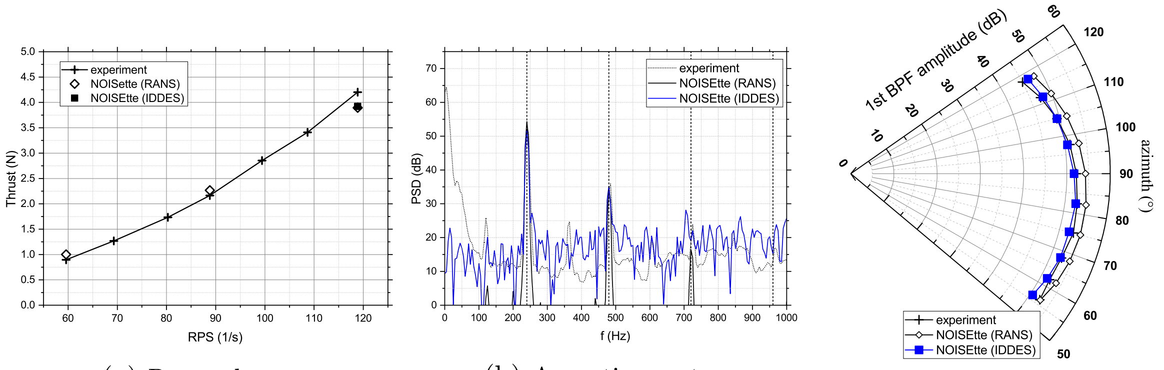
(a) Tip vortices trajectories

(b) Normal force coefficient span distribution

(c) Pressure difference at the blade leading edge

**Figure 12.** Rotor in forward flight numerical simulation results

The fourth case represents simulations of turbulent flow around a two-bladed rotor of a small-scale UAV [18]. Rotor aerodynamics and near-field acoustics were modeled using both RANS and scale-resolving hybrid RANS-LES approaches (see Sec. 1.2). The far-field acoustics were evaluated using the integral Ffowcs-Williams and Hawkings method described in Sec. 3.6. The numerical results were compared with the available experimental data.



(a) Rotor thrust

(b) Acoustic spectra

(c) 1st BPF amplitude azimuthal distribution

**Figure 13.** Small-scaled UAV rotor simulation results

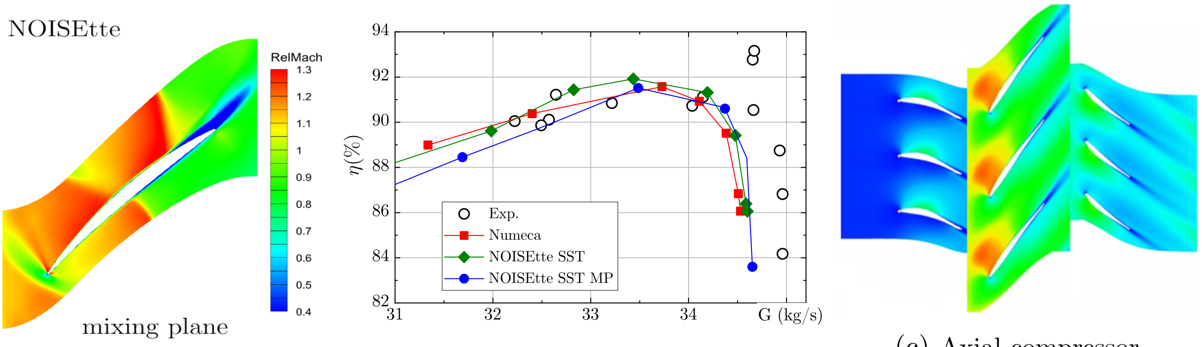
The simulations were carried out using hybrid supercomputer using several dozen GPUs with techniques described in Sec. 2 . The performed computational experiments using the RANS and IDDES methods confirmed that both approaches are capable to predict aerodynamics and tonal noise of an isolated UAV propeller in hover with acceptable accuracy. The good agreement for propeller thrust is well seen in Fig. 13a. Both RANS and IDDES gives good accuracy for first and second tones of blade passing frequency (BPF) (Fig. 13b). At the same time the scale-resolving approach is more accurate in prediction of first BPF tone amplitude for the whole azimuthal directions range (Fig. 13c).

## 4.5. Turbomachinery

The turbomachinery cases considered while testing the realization of the MP technology [26] include single rotor (e.g., Rotor67), axial turbines, axial and centrifugal compressors. The results of simulations are well-compared both with reference data (mostly integral characteristics) and the results obtained using commercial turbomachinery-oriented CFD software packages, such as Cadence (former Numeca) FINE/Turbo.

Figures 14a and 14b demonstrate the results of computation of the Rotor67 test case. Head-capacity characteristics (dependence of efficiency $\eta$ on mass flow rate $G$) for this configuration are presented in Fig. 14b. It is seen that the results with usage of the MP (upstream and downstream the rotor) correlate well with the experimental data, results of simulations using the Cadence FINE/Turbo commercial CFD software, as well as with the results obtained without rotor-stator interfaces.

Figure 14c presents the relative Mach number field for 1.5 stage of the axial compressor case. The NOISEtte results are well compared with those provided using the Cadence Fine/Turbo (see [26] for details).



(a) Rotor67 flow field

(b) Rotor67 efficiency

(c) Axial compressor flow field

**Figure 14.** Turbomachinery test cases: Rotor67 flow visualization (a) and head-capacity characteristics (b); visualization of flow in an axial compressor (c)

## Conclusion

The paper presents quite a complete overview of the CFD/CAA supercomputer code NOISEtte. It provides a brief description of the mathematical models, numerical methods and computational technologies implemented in the code. Notable qualities of the code include the use of lower-cost higher-accuracy numerical methods on unstructured meshes (i), a focus on high-fidelity simulations of complex non-stationary turbulent flows (ii), the developed techniques to retrieve and analyze far acoustic fields (iii), and an effective heterogeneous parallel model that allows to run computations with high efficiency on modern high-performance computing systems with different architectures (iv). The paper specially contains many references to the authors' publications, which provide an opportunity to get a deeper insight into the implemented algorithms.

## Acknowledgements

The development of dedicated turbomachinery techniques (see Section 3.2) was carried out within the project 21-71-10100 funded by the Russian Science Foundation. Section 4.3 is prepared in the implementation of the program for the creation and development of the World-Class Research Center “Supersonic” for 2020–2050 funded by the Ministry of Science and Higher Education of the Russian Federation (Grant agreement as of 25.04.2022 № 075-15-2022-330). The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [82], the equipment of Shared Resource Center of KIAM RAS (http://ckp.kiam.ru). The authors thankfully acknowledge these institutions.

## References

1. 4th AIAA CFD High Lift Prediction Workshop – Test Cases, `https://hiliftpw.larc.nasa.gov/Workshop4/testcases.html`

2. Abalakin, I., Bakhvalov, P., Kozubskaya, T.: Edge-based reconstruction schemes for unstructured tetrahedral meshes. International Journal for Numerical Methods in Fluids 81,

331–356 (2016). https://doi.org/10.1002/fld.4187

3. Abalakin, I., Kozubskaya, T., Vasilyev, O., Zhdanova, N.: Implementation of charcteristic based volume penalization method on unstructured meshes (01 2021). https://doi.org/10.23967/wccm-eccomas.2020.176

4. Abalakin, I.V., Anikin, V.A., Bakhvalov, P.A., *et al.*: Numerical investigation of the aerodynamic and acoustical properties of a shrouded rotor. Fluid Dyn. 51(3), 419–433 (Nov 2016). https://doi.org/10.1134/S0015462816030145

5. Abalakin, I.V., Bakhvalov, P.A., Bobkov, V.G., Gorobets, A.V.: Parallel algorithm for flow simulation in rotor–stator systems based on edge-based schemes. Math. Models Comput. Simul. 13, 172–180 (2021). https://doi.org/10.1134/S2070048221010026

6. Abalakin, I.V., Bobkov, V.G., Kozubskaya, T.K., *et al.*: Numerical simulation of flow around rigid rotor in forward flight. Fluid Dynamics 55(4), 534–544 (Jul 2020). https://doi.org/10.1134/s0015462820040011

7. Abalakin, I.V., Vasilyev, O.V., Zhdanova, N.S., Kozubskaya, T.K.: Characteristic based volume penalization method for numerical simulation of compressible flows on unstructured meshes. Comput. Math. and Math. Phys. 61(8), 1315–1329 (2021). https://doi.org/10.1134/S0965542521080029

8. Abalakin, I.V., Zhdanova, N.S., Kozubskaya, T.K.: Immersed boundary method for numerical simulation of inviscid compressible flow. Comput. Math. and Math. Phys. 58(9), 1411–1419 (2018). https://doi.org/10.1134/S0965542518090026

9. Aksenov, A., Dyadkin, A., Pokhilko, V.: Overcoming of barrier between CAD and CFD by modified finite volume method. In: ASME Pressure Vessels and Piping Division Conference Proc. vol. 377-1. ASME PVP, San Diego (1998)

10. Angot, P., Bruneau, C.H., Fabrie, P.: A penalization method to take into account obstacles in viscous flows. Numerische Mathematik 81, 497–520 (1999). https://doi.org/10.1007/s002110050401

11. Ansys inc.: ANSYS, https://www.ansys.com

12. Bakhvalov, P.A., Bobkov, V.G., Kozubskaya, T.K.: Technology to predict acoustic far-field disturbances in the case of calculations in a rotating reference frame. Math. Models and Comp. Simul. 9, 717–727 (2017). https://doi.org/https://doi.org/10.1134/S2070048217060035

13. Bakhvalov, P.A., Kozubskaya, T.K.: EBR-WENO scheme for solving gas dynamics problems with discontinuities on unstructured meshes. Computers and Fluids 157, 312–324 (2017). https://doi.org/10.1016/j.compfluid.2017.09.004

14. Bakhvalov, P.A., Kozubskaya, T.K., Rodionov, P.V.: EBR schemes with curvilinear reconstructions for hybrid meshes. Computers and Fluids 239 (2022). https://doi.org/10.1016/j.compfluid.2022.105352

15. Bakhvalov, P.A., Surnachev, M.D.: Method of averaged element splittings for diffusion terms discretization in vertex-centered framework. Journal of Computational Physics 450 (2022). https://doi.org/10.1016/j.jcp.2021.110819

16. Bakhvalov, P.A., Vershkov, V.A.: Edge-based schemes on moving hybrid meshes in the NOISEtte code. Keldysh Inst. Appl. Math. Preprint 127 (2018). https://doi.org/10.20948/prepr-2018-127

17. Bobkov, V.G., Vershkov, V.A., Kozubskaya, T.K., Tsvetkova, V.O.: Deformation technique of unstructured mesh deformation to find the aerodynamic characteristics of bodies at small displacements. Mathematical Models and Computer Simulations 13(6), 986–1001 (Nov 2021). https://doi.org/10.1134/s2070048221060028

18. Bobkov, V., Gorobets, A., Kozubskaya, T., Zhang, X., Zhong, S.: Supercomputer simulation of turbulent flow around isolated UAV rotor and associated acoustic fields. Communications in Computer and Information Science 1510, 256–269 (2021). https://doi.org/10.1007/978-3-030-92864-3_20

19. Brentner, K.S., Farassat, F.: Modeling aerodynamically generated sound of helicopter rotors. Progress in Aerospace Sciences 39(2), 83–120 (2003). https://doi.org/10.1016/S0376-0421(02)00068-4

20. Brown-Dymkoski, E., Kasimov, N., Vasilyev, O.V.: A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows. J. Comp. Phys. 262, 344–357 (2014). https://doi.org/10.1063/1.4825260

21. Caradonna, F.X., Tung, C.Y.: Experimental and analytical studies of a model helicopter rotor in hover (1980), https://api.semanticscholar.org/CorpusID:106679612

22. Dankov, B.N., Duben, A.P., Kozubskaya, T.K.: Analysis of Self-Oscillation Processes in a Cavity with a Flow of OpenType on the Basis of the Data of Vortex-Resolving Calculations. Fluid Dynamics 58(4), 659–669 (Aug 2023). https://doi.org/10.1134/S0015462823600517

23. Deck, S., Renard, N.: Towards an enhanced protection of attached boundary layers in hybrid RANS/LES methods. Journal of Computational Physics 400, 108970 (Jan 2020). https://doi.org/10.1016/j.jcp.2019.108970

24. Denton, J.D.: An improved time-marching method for turbomachinery flow calculation. Journal of Engineering for Power 105(3), 514–521 (Jul 1983). https://doi.org/10.1115/1.3227444

25. Duben, A.P., Abalakin, I.V., Tsvetkova, V.O.: On boundary conditions on solid walls in viscous flow problems. Math. Models and Comput. Simul. 13(4), 591–603 (2021). https://doi.org/10.1134/S2070048221040128

26. Duben, A., Gorobets, A., Soukov, S., *et al.*: Supercomputer Simulations of Turbomachinery Problems with Higher Accuracy on Unstructured Meshes, pp. 356–367. Springer International Publishing (2022). https://doi.org/10.1007/978-3-031-22941-1_26

27. Duben, A., Kozubskaya, T., Bosnyakov, S.: Two Cases Calling for Scale-Resolving Simulation, pp. 77–87. Springer International Publishing (2022). https://doi.org/10.1007/978-3-031-12019-0_6

28. Duben, A.P., Kozubskaya, T.K.: Evaluation of Quasi-One-Dimensional Unstructured Method for Jet Noise Prediction. AIAA Journal 57(12), 5142–5155 (dec 2019). https://doi.org/10.2514/1.j058162

29. Duben, A.P., Ruano, J., Gorobets, A.V., *et al.*: Evaluation of enhanced gray area mitigation approaches based on jet aeroacoustics. AIAA Journal 61(2), 612–625 (Feb 2023). https://doi.org/10.2514/1.j062116

30. Evans, A., Lacy, D., Smith, I., Rivers, M.: Test Summary of the NASA High-Lift Common Research Model Half-Span at QinetiQ 5-Metre Pressurized Low-Speed Wind Tunnel. In: AIAA Paper 2020-2770. American Institute of Aeronautics and Astronautics Inc, AIAA (2020). https://doi.org/10.2514/6.2020-2770

31. Farassat, F.: Linear acoustic formulas for calculation of rotating blade noise. AIAA Journal 19(9), 1122–1130 (1981). https://doi.org/10.2514/3.60051

32. Ferris, R., Sacks, M., Cerizza, D., *et al.*: Aeroacoustic Computations of a Generic Low Boom Concept in Landing Configuration: Part 1-Aerodynamic Simulations. AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021 (2021). https://doi.org/10.2514/6.2021-2195

33. Ffowcs Williams, J., Hawkings, D.: Sound generated by turbulence and surfaces in unsteady motion. Philosophical Transactions of the Royal Society A. Mathematical, Physical and Engineering Sciences 264, 321–342 (1969). https://doi.org/10.1098/rsta.1969.0031

34. Filimonov, S.A., Gavrilov, A., Dekterev, A.A., Litvintsev, K.Y.: Mathematical modeling of the interaction of a thermal convective flow and a moving body. Computational Continuum Mechanics 16(1), 89–100 (Apr 2023). https://doi.org/10.7242/1999-6691/2023.16.1.7

35. FlowVision: FlowVision CFD package, https://flowvisioncfd.com

36. FSUE RFNC - VNIIEF: LOGOS CFD package, https://logos-support.ru

37. Gorobets, A.V., Duben, A.P., Kozubskaya, T.K., Rodionov, P.V.: Approaches to the numerical simulation of the acoustic field generated by a multi-element aircraft wing in high-lift configuration. Mathematical Models and Computer Simulations 15(1), 92–108 (feb 2023). https://doi.org/10.1134/S2070048223010088

38. Gorobets, A., Bakhvalov, P.: Heterogeneous CPU+GPU parallelization for high-accuracy scale-resolving simulations of compressible turbulent flows on hybrid supercomputers. Computer Physics Communications 271, 108231 (Dec 2022). https://doi.org/10.1016/j.cpc.2021.108231

39. Gorobets, A.V., Duben, A.P.: Technology for supercomputer simulation of turbulent flows in the good new days of exascale computing. Supercomputing Frontiers and Innovations 8(4), 4–10 (Dec 2021). https://doi.org/10.14529/jsfi210401

40. Gorobets, A.: An Approach to the Implementation of the Multigrid Method with Full Approximation for CFD Problems. Computational Mathematics and Mathematical Physics 63, 2150–2161 (2023). https://doi.org/10.1134/S0965542523110106

41. Gorobets, A., Magomedov, A., Soukov, S.: Heterogeneous parallel implementation of a multigrid method with full approximation in the NOISEtte code. Matematicheskoye modelirovaniye 36, in press (2024)

42. Guillard, H., Viozat, C.: On the behaviour of upwind schemes in the low Mach number limit. Computers and Fluids 28, 63–86 (1999). https://doi.org/10.1016/S0045-7930(98)00017-6

43. Guseva, E.K., Garbaruk, A.V., Strelets, M.K.: Assessment of Delayed DES and Improved Delayed DES Combined with a Shear-Layer-Adapted Subgrid Length-Scale in Separated Flows. Flow, Turbulence and Combustion 98, 481–502 (2017). https://doi.org/10.1007/s10494-016-9769-7

44. Guseva, E.K., Garbaruk, A.V., Strelets, M.K.: An automatic hybrid numerical scheme for global RANS-LES approaches. Journal of Physics: Conference Series 929, 012099 (nov 2017). https://doi.org/10.1088/1742-6596/929/1/012099

45. INM RAS: INMOST CFD package, http://inmost.ru

46. Khalighi, Y., Ham, F., Nichols, J., Lele, S., Moin, P.: Unstructured large eddy simulation for prediction of noise issued from turbulent jets in various configurations. In: AIAA Paper 2011-2886 (2011). https://doi.org/10.2514/6.2011-2886

47. Khorrami, M.R., Shea, P.R., Winski, C.S., *et al.*: Aeroacoustic Computations of a Generic Low Boom Concept in Landing Configuration: Part 3-Aerodynamic Validation and Noise Source Identification. AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021 (2021). https://doi.org/10.2514/6.2021-2197

48. Kolesnik, E., Smirnov, E., Smirnovsky, A.: RANS-based numerical simulation of shock wave/turbulent boundary layer interaction induced by a blunted fin normal to a flat plate. Computers & Fluids 247, 105622 (oct 2022). https://doi.org/10.1016/j.compfluid.2022.105622

49. Langtry, R.B., Menter, F.R.: Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes. AIAA Journal 47(12), 2894–2906 (dec 2009). https://doi.org/10.2514/1.42362

50. LeVeque, R.J.: Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics, Cambridge University Press (2002). https://doi.org/10.1017/CBO9780511791253

51. Marakueva, O., Duben, A.: Accuracy of flow simulation in a low-pressure turbine using a laminar-turbulent transition model. St. Petersburg State Polytechnical University Journal: Physics and Mathematics (2023). https://doi.org/10.18721/JPM.161.141

52. Menter, F.R.: Two-equation eddy-viscosity turbulence models for engineering applications. AIAA Journal 32(8), 1598–1605 (Aug 1994). https://doi.org/10.2514/3.12149

53. Menter, F.R., Matyushenko, A., Lechner, R., *et al.*: An algebraic LCTM model for laminar–turbulent transition prediction. Flow, Turbulence and Combustion 109(4), 841–869 (Jul 2022). `https://doi.org/10.1007/s10494-022-00336-8`

54. Menter, F.R., Smirnov, P.E., Liu, T., Avancha, R.: A one-equation local correlation-based transition model. Flow, Turbulence and Combustion 95(4), 583–619 (jul 2015). `https://doi.org/10.1007/s10494-015-9622-4`

55. MIPT: FlowModellium CFD package, `https://flowmodellium.ru`

56. Mockett, C., Fuchs, M., Garbaruk, A., *et al.*: Two non-zonal approaches to accelerate RANS to LES transition of free shear layers in DES. In: Progress in Hybrid RANS-LES Modelling, pp. 187–201. Springer International Publishing (2015). `https://doi.org/10.1007/978-3-319-15141-0_15`

57. Murayama, M., Yokokawa, Y., Ura, H., *et al.*: Experimental study of slat noise from 30P30N three-element high-lift airfoil in JAXA kevlar-wall low-speed wind tunnel. In: AIAA Paper 2018-3460. American Institute of Aeronautics and Astronautics Inc, AIAA (2018). `https://doi.org/10.2514/6.2018-3460`

58. Nicoud, F., Toda, H.B., Cabrit, O., Bose, S., Lee, J.: Using singular values to build a subgrid-scale model for large eddy simulations. Physics of Fluids 23(8), 085106 (aug 2011). `https://doi.org/10.1063/1.3623274`

59. Nkonga, B., Guillard, H.: Godunov type method on non-structured meshes for three-dimensional moving boundary problems. Computer Methods in Applied Mechanics and Engineering 113(1-2), 183–204 (1994). `https://doi.org/10.1016/0045-7825(94)90218-6`

60. Pascioni, K.A., Cattafesta, L.N.: Aeroacoustic measurements of leading-edge slat noise. AIAA Paper 2016-2960 (2016). `https://doi.org/10.2514/6.2016-2960`

61. Pascioni, K.A., Cattafesta, L.N., Choudhari, M.M.: An experimental investigation of the 30P30N multi-element high-lift airfoil. In: AIAA Paper 2014-3062. American Institute of Aeronautics and Astronautics Inc. (2014). `https://doi.org/10.2514/6.2014-3062`

62. Pont-Vlchez, A., Duben, A., Gorobets, A., *et al.*: New strategies for mitigating the gray area in delayed-detached eddy simulation models. AIAA Journal 59(9), 1–15 (Sep 2021). `https://doi.org/10.2514/1.j059666`

63. Ribeiro, A.F., Ferris, R., Khorrami, M.R.: Aeroacoustic Computations of a Generic Low Boom Concept in Landing Configuration: Part 2-Airframe Noise Simulations. AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021 (2021). `https://doi.org/10.2514/6.2021-2196`

64. Roe, P.L.: Approximate Riemann solvers, parameter vectors and difference schemes. Journal of Computational Physics 43(2), 357–372 (1981). `https://doi.org/10.1016/0021-9991(81)90128-5`

65. Saad, Y.: Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics, Philadelphia (2003)

66. Shershnev, A.A., Kudryavtsev, A.N., Kashkovsky, A.V., *et al.*: A numerical code for a wide range of compressible flows on hybrid computational architectures. Supercomputing Frontiers and Innovations 9(4), 85–99 (Dec 2022). https://doi.org/10.14529/jsfi220408

67. Shur, M., Strelets, M., Travin, A., *et al.*: Improved Embedded Approaches. Notes on Numerical Fluid Mechanics and Multidisciplinary Design 134, 65–69 (2017). https://doi.org/10.1007/978-3-319-52995-0_3

68. Shur, M.L., Spalart, P.R., Strelets, M.K., Travin, A.K.: Synthetic Turbulence Generators for RANS-LES Interfaces in Zonal Simulations of Aerodynamic and Aeroacoustic Problems. Flow, Turbulence and Combustion 93, 63–92 (2014). https://doi.org/10.1007/s10494-014-9534-8

69. Shur, M.L., Spalart, P.R., Strelets, M.K., Travin, A.K.: An enhanced version of DES with rapid transition from RANS to LES in separated flows. Flow, Turbulence and Combustion 95(4), 709–737 (2015). https://doi.org/10.1007/s10494-015-9618-0

70. Shur, M., Spalart, P., Strelets, M.: Noise prediction for increasingly complex jets. Part I, Methods and tests; Part II, Applications. International Journal of Aeroacoustics 4, 213–266 (2005). https://doi.org/10.1260/1475472054771376

71. Smagorinsky, J.: General circulation experiments with the primitive equations. Monthly Weather Review 91(3), 99–164 (1963). https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2

72. Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows. In: 30th Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics (Jan 1992). https://doi.org/10.2514/6.1992-439

73. Spalart, P., Shur, M.: Variants of the Ffowcs Williams–Hawkings equation and their coupling with simulations of hot jets. International Journal of Aeroacoustics 8, 477–492 (2009). https://doi.org/10.1260/147547209788549280

74. Stabnikov, A., Garbaruk, A.: An algebraic transition model for simulation of turbulent flows based on a detached eddy simulation approach. St. Petersburg Polytechnic University Journal - Physics and Mathematics 15(1) (2022). https://doi.org/10.18721/JPM.15102

75. T1: CADFlo CFD package, https://cadflo.com

76. Titarev, V.A., Faranosov, G.A., Chernyshev, S.A., Batrakov, A.S.: Numerical modeling of the influence of the relative positions of a propeller and pylon on turboprop aircraft noise. Acoustical Physics 64(6), 760–773 (Nov 2018). https://doi.org/10.1134/s1063771018060118

77. Toro, E.: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer (2009). https://doi.org/10.1007/b79761

78. Trias, F.X., Folch, D., Gorobets, A., Oliva, A.: Building proper invariants for eddy-viscosity subgrid-scale models. Physics of Fluids 27(6), 065103 (jun 2015). https://doi.org/10.1063/1.4921817

79. Trias, F.X., Gorobets, A., Silvis, M.H., *et al.*: A new subgrid characteristic length for turbulence simulations on anisotropic grids. Physics of Fluids 29(11), 115109 (2017). `https://doi.org/10.1063/1.5012546`

80. Vasilyev, O.V., Zhdanova, N.S.: Characteristic-based volume penalization-imposed wall function method for turbulent boundary layer modeling. Computational Mathematics and Mathematical Physics 63(5), 821–836 (2023). `https://doi.org/10.1134/S0965542523050160`

81. Vasilyev, O.V., Zhdanova, N.S.: Generalization of the penalized wall function method for modeling of turbulent flows with adverse pressure gradient. Computational Mathematics and Mathematical Physics 63(12), 2384–2401 (2023). `https://doi.org/10.1134/S0965542523120199`

82. Voevodin, V., Antonov, A., Nikitenko, D., *et al.*: Supercomputer Lomonosov-2: Large scale, deep monitoring and fine analytics for the user community. Supercomput. Front. Innov. 6(2), 4–11 (2019). `https://doi.org/10.14529/jsfi190201`

83. Zhdanova, N.S., Abalakin, I.V., Vasilyev, O.V.: Generalized Brinkman volume penalization method for compressible flows around moving obstacles. Math. Models. and Comput. Simul. 14(5), 716–726 (2022). `https://doi.org/10.1134/S2070048222050180`

84. Zhdanova, N.S., Vasilyev, O.V.: Penalized wall function method for turbulent flow modeling. Supercomputing Frontiers and Innovations 9(4), 55–68 (Dec 2022). `https://doi.org/10.14529/jsfi220406`