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, ρ is the density, $E = \rho\boldsymbol{u}^2/2 + \rho\varepsilon$ is the total energy, ε is the specific internal energy, $p = \rho\varepsilon\,(\gamma - 1)$ is the pressure defined by the ideal perfect gas state of equation, the constant γ 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 ρ, $\boldsymbol{u}$, 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 δ_{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 τ_{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 μ 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 γ [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, σ 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., σ [58] or a S3PQR model [78]). A few enhanced dynamic sub-grid scale models are implemented: $\tilde{\Delta}_\omega$, Δ_{SLA} [56, 69]; Δ_{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.

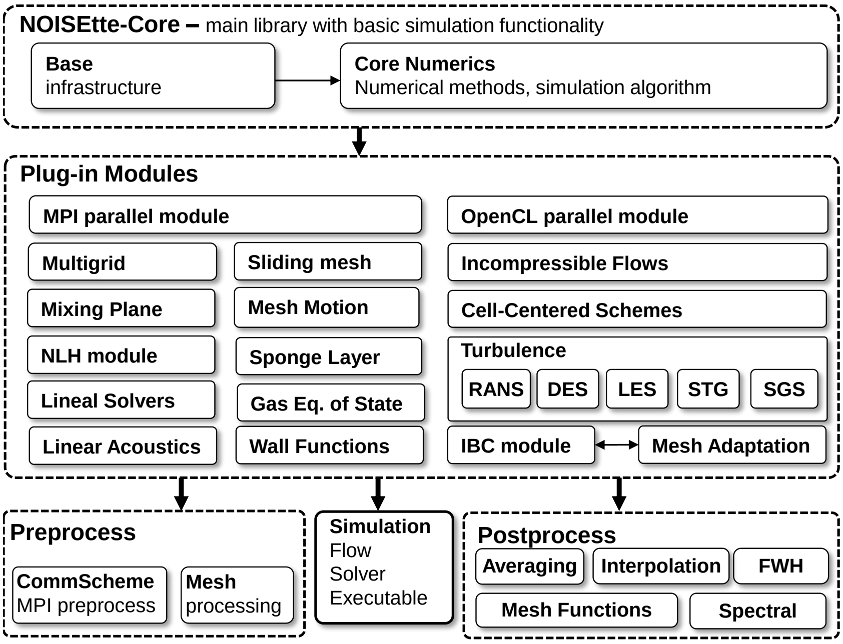


Figure 1. Code structure diagram

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.

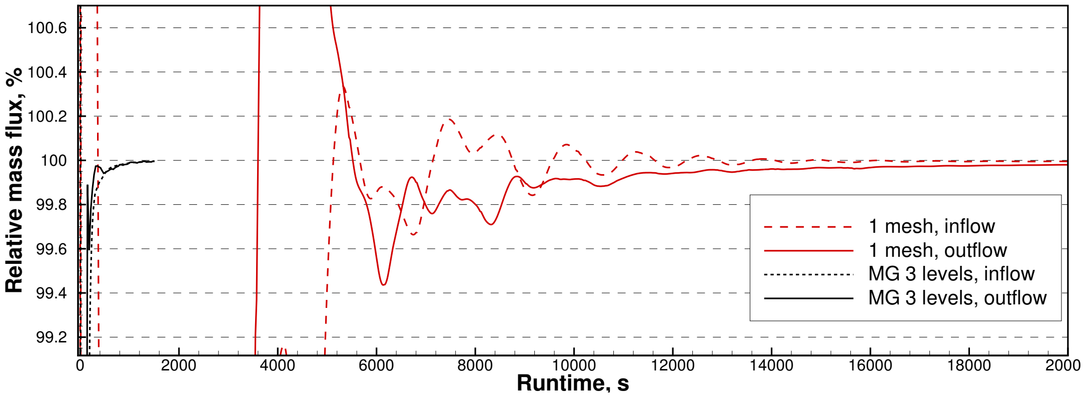


Figure 2. Multigrid convergence acceleration example: input and output fluxes (in relative values compared to the reference value) of a multistage axial compressor on a mesh of about 20 million nodes

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

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

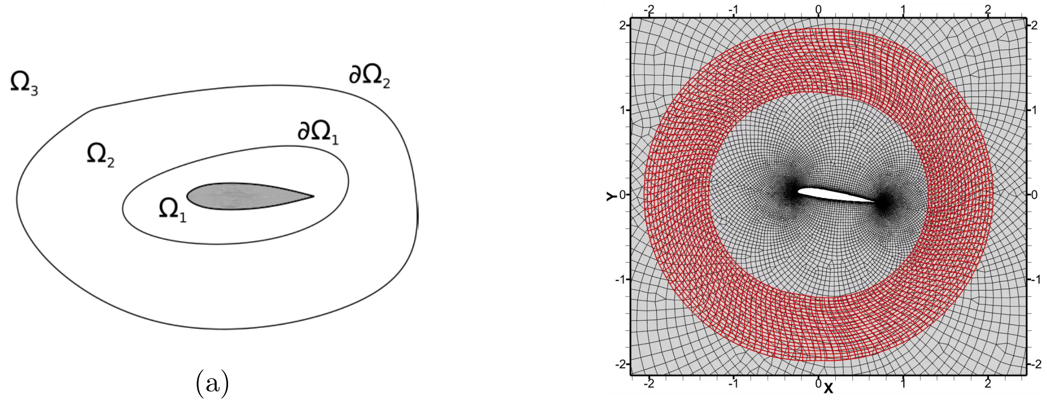
where $\partial C_i(t)$ is the boundary of cell, n is the unit external normal to the boundary $\partial C_i(t)$, $\boldsymbol{V}_c$ is velocity of the boundary of the moving cell.

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 Ω_1, Ω_2, and Ω_3 (Fig. 3a).

Domain Ω_1 contains a moving object, the mesh nodes included in Ω_1 move with this solid body, and the mesh elements do not deform. Domain Ω_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 Ω_3 contains all nodes located far from the surface of the body that must remain motionless. Domain Ω_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 Ω_1 and Ω_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].

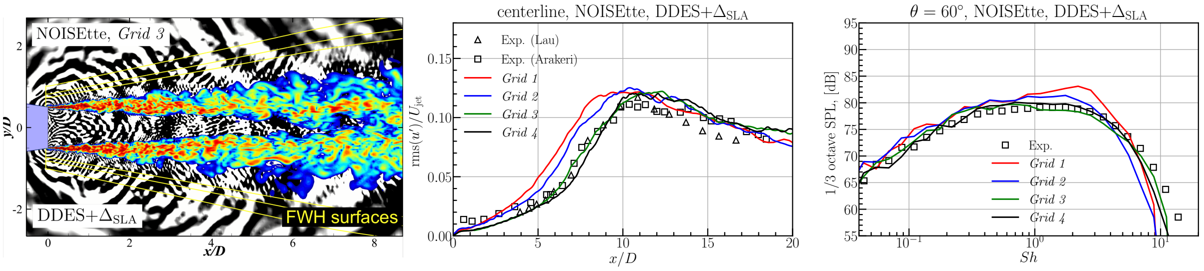
The proposed method allows to simulate turbulent flow near moving solid bodies and applicable for modelling a wide range of problems such as oscillating airfoils and wings, hinged rotor blades movement with pitch control, etc.

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+Δ_{SLA}.



(a)

(b)

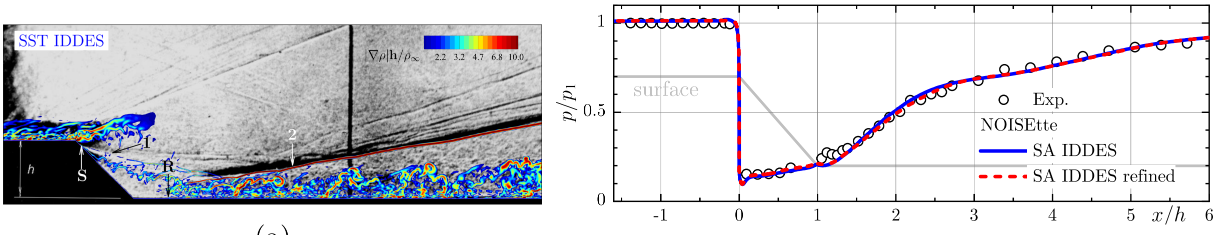
(c)

Figure 4. Subsonic turbulent jet: flow visualization (a) and mesh convergence results for the DDES+Δ_{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).

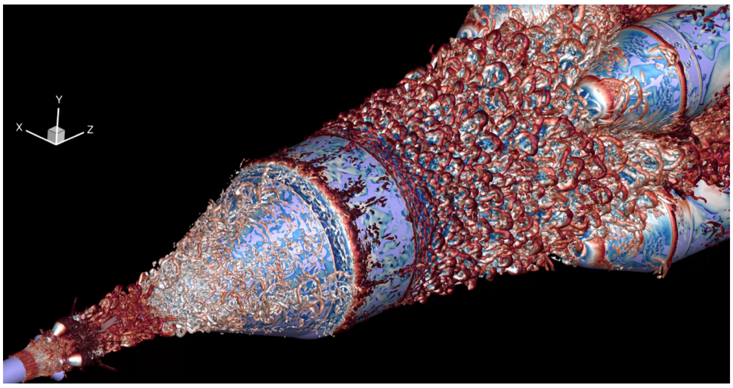
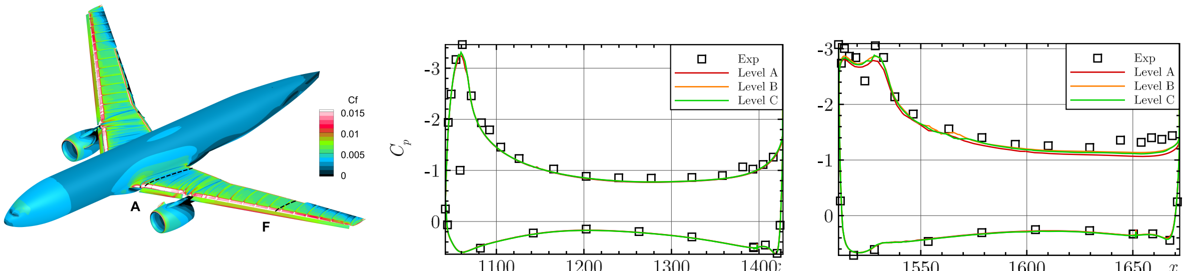


Figure 6. Space rocket: flow visualizations (isosurfaces of Q criterion colored by Mach number levels)

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).



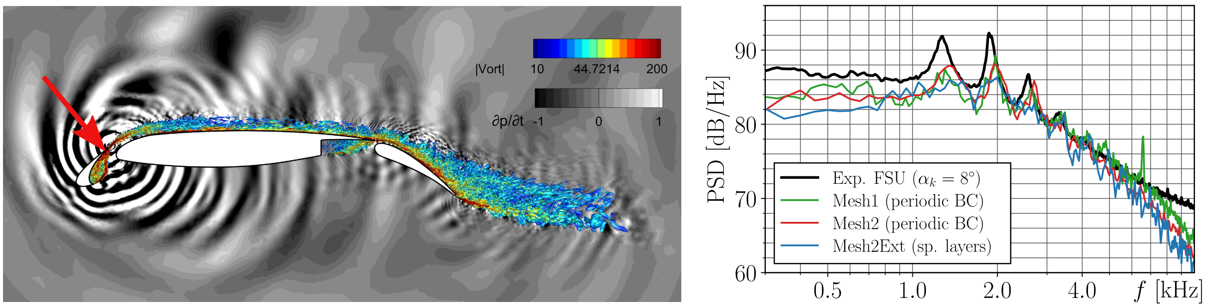
(a) Location of pressure tap rows

(b) Row A

(c) Row F

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

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.

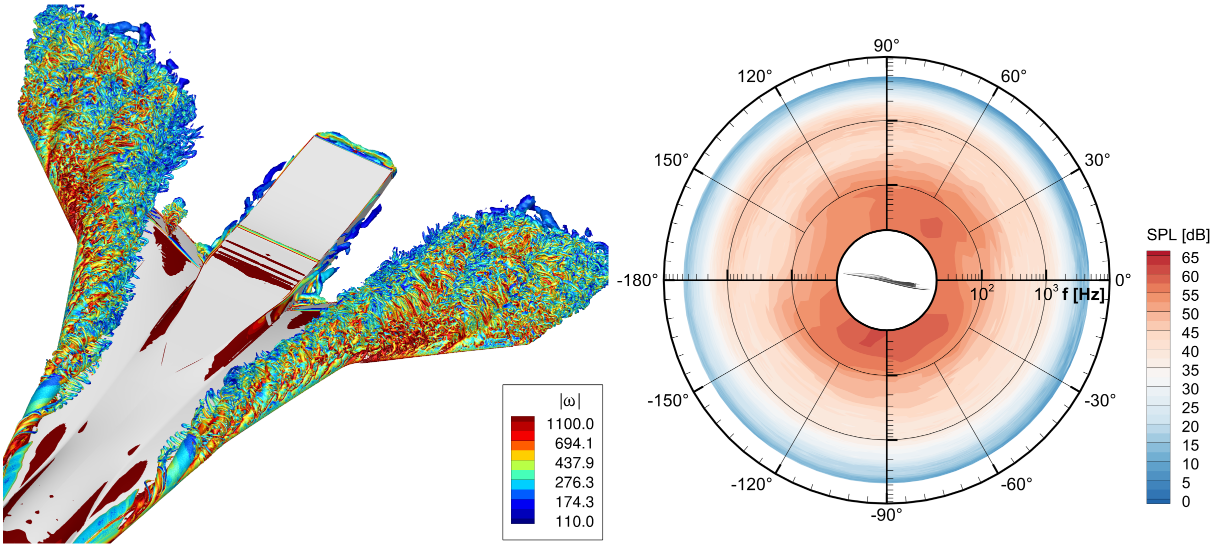


(a) Instantaneous flow field

(b) Spectra at Point 6 (red arrow)

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

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 ($\mathrm{M} = 0.2$, $\mathrm{Re}_{1\mathrm{m}} = 4.6 \times 10^6$, $\alpha = 10°$)

4.4. Helicopter and Drone Rotors

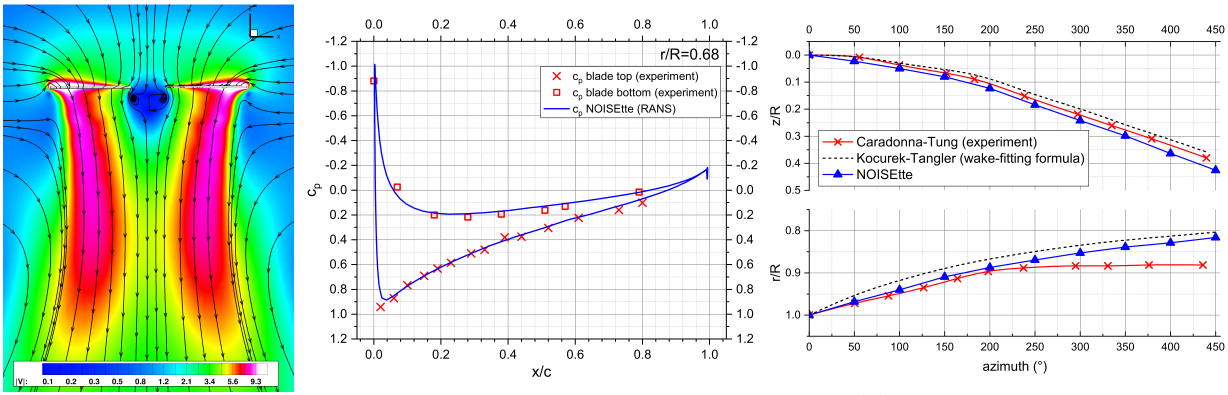
The methods described above were applied for simulating the turbulent flow around rotorcraft rotors and calculating its aerodynamic and acoustic characteristics.

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.

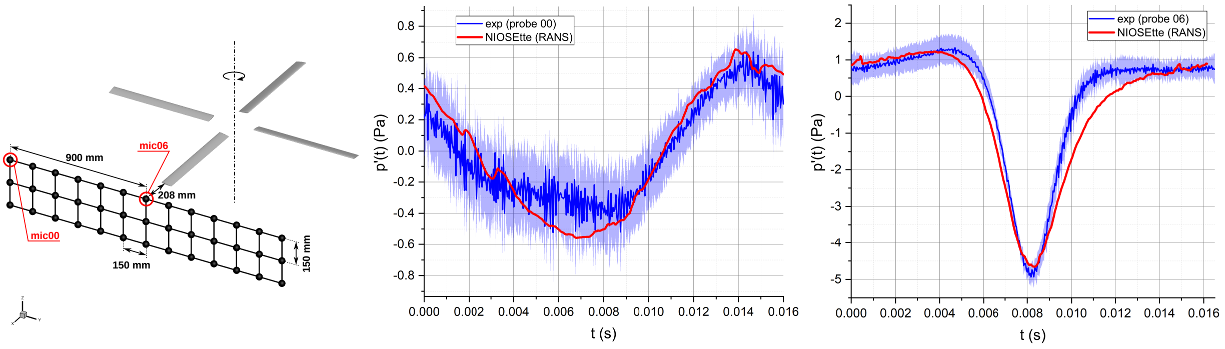


(a) Flow field

(b) Pressure coefficient distribution

(c) Tip vortex evolution

Figure 10. Caradonna-Tung rotor simulation results



(a) Probes setup

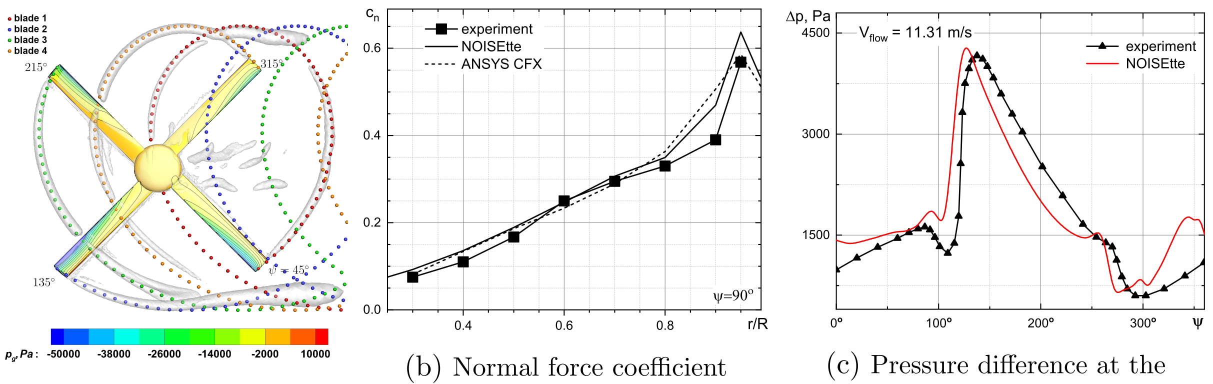
(b) Pressure pulsations, mic00

(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).



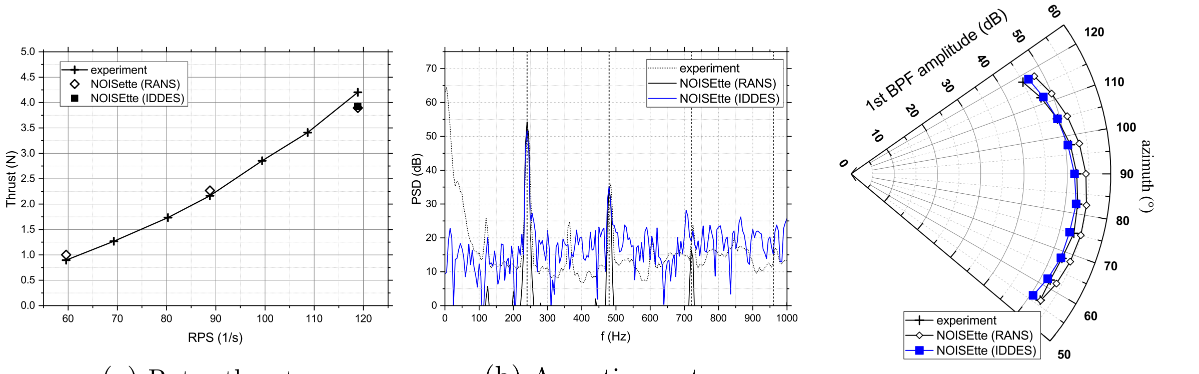
(a) Tip vortices trajectories

(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

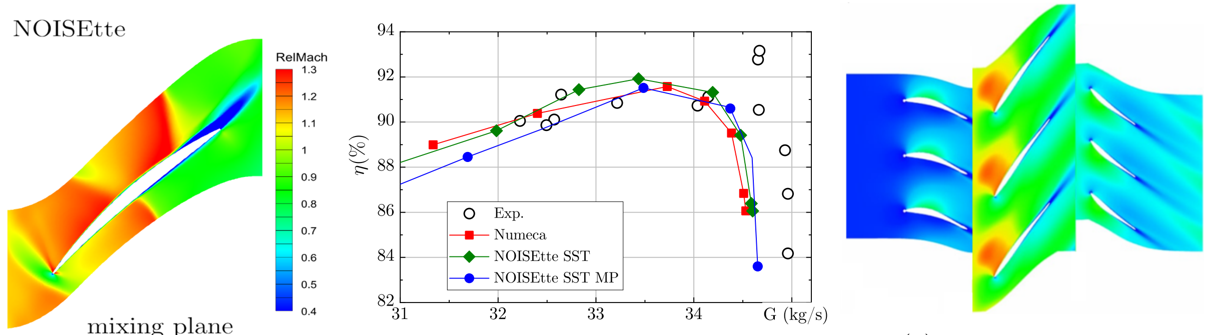
The simulations were carried out using hybrid supercomputer using several dozen GPUs with techniques described in Sec. 2 . The performed computational experiments using the RANS and IDDES methods confirmed that both approaches are capable to predict aerodynamics and tonal noise of an isolated UAV propeller 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 η 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 farfield 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