DOI: 10.14529/jsfi220407

Hybrid Dynamic Mesh Redistribution – Immersed Boundary Method for Acoustic Simulation of Flow Around a Propeller[*]

Vladimir G. Bobkov[1], *Tatiana K. Kozubskaya*[1],
Liudmila N. Kudryavtseva[1,2], *Valeriia O. Tsvetkova*[1]

A novel hybrid dynamic mesh redistribution – immersed boundary method for simulation of turbulent flows around rotating obstacles of complex geometry and analysis of tonal acoustics is proposed. The feasibility of the approach is demonstrated by considering a drone propeller problem. The results of three-dimensional Reynolds-averaged Navier–Stokes simulations using the proposed approach are compared to the results of body-fitted unstructured simulations in non-inertial reference frame. The dynamic mesh redistribution method allows the reposition of mesh points taking into account the shape of the moving body while retaining the mesh topology. The cell size and quality of the dynamically redistributed mesh strongly depend on the curvature of the body surface. The position and shape of the moving obstacle is prescribed by a distance function defined on an adaptive octree. The results of simulations using the proposed method are in good agreement with both the results of body-fitted simulations and the experimental data.

Keywords: moving adaptive mesh, immersed boundary method, drone rotor, rotor acoustics, unstructured mesh, turbulent flow.

Introduction

The problem of flow simulation around moving obstacles or body parts is of high importance in computational aerodynamics and aeroacoustics. There is a number of different modeling techniques and mesh changing methods that are currently being developed to approach the above-described problem. Most popular approaches preserve the classical body-fitted (BF) meshes, requiring boundary nodes to coincide with the obstacle surface. For instance, mesh deformation method, based on quasi one-dimensional elastic media [6], is the least computationally expensive body-fitted approach, which allows efficient deformation of the computational mesh governed by the body motion. The main disadvantage of the mesh deformation method is its inability to handle large obstacle displacements leading to mesh deterioration and/or entanglement. An alternative approaches based on mesh recomputation [7] or adaptive mesh refinement [8], regardless of their computational efficiency, introduce interpolation errors and significantly complicate the process of parallelization, since the topology and mesh size can change over time. The Chimera methods [9] utilizing two or more superimposed computational meshes also require interpolation and data exchange between the meshes.

This paper describes a novel hybrid dynamic mesh redistribution – immersed boundary method (DMR-IBM), capable of simulating compressible flows around arbitrary number of obstacles, either moving according to their own laws or displaced under the action of aerodynamic forces. The feasibility of combining the immersed boundary method with the dynamic mesh redistribution is evaluated for the acoustic simulations of a compressible flow around an isolated drone rotor. The immersed boundary approach is based on the Brinkman penalization method [1–5], where the obstacle is modeled by introducing additional source terms into the

[*] One of the selected papers of The Ninth Russian Conference "Computational Experiment in Aeroacoustics and Aerodynamics" (CEAA2022)

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

[2] Dorodnicyn Computing Center FRC CSC RAS, Moscow, Russian Federation

governing equations describing the evolution of compressible viscous flow. These sources define the body as a continuous porous medium with low permeability. The source terms are non-zero inside the moving obstacle and zero – outside. Thus, the use of the immersed boundary method makes it possible to define the external flow problem in a simply connected domain, which opens up the possibility of using the dynamic mesh redistribution method to position mesh points in the vicinity of the surface of the moving obstacle, while maintaining the mesh topology. In this paper, we consider the problem of aeroacoustics simulation of a drone propeller taken as an example of a rotating body of a very complex geometry. The hybrid DMR-IBM was previously formulated for two-dimensional flows and tested for the rotating projection of the propeller [5]. In this paper, the DMR-IBM is generalized to three dimensions and is verified for the problem of tonal acoustics of a propeller by comparing the results of the simulations against the data obtained using the classical body-fitted approach in non-inertial frame of reference. The DMR-IBM implemented for a propeller located inside of a puck-shaped domain with the mesh points dynamically redistributed to follow the motion of the propeller and to resolve the boundary layer around it. Such an approach would allow, in future, to use DMR-IBM for simulation of flows around multiple rotating objects located within its own puck-shaped subdomian belonging to a larger computational domain, while approximating fuselage using a body-fitted mesh approach. An example of such a problem would be a flow around vehicle with several rotors, e.g., a quad-copter.

A complex geometry of the propeller is characterized by sharp corners, very thin parts of blades, and high curvature regions. To provide a sufficiently high resolution mesh near the rotating propeller surface by means of dynamic mesh redistribution method is quite challenging problem. To overcome this challenge, a number of novel methods has been developed. To start, the initial mesh with high node density in the region of the propeller rotation is constructed. Then the control mesh adaptation metric is developed. This metric depends on the distance function and its gradient as well as on parameters of the immersed body (i.e. propeller) and its surface, such as curvature and distances to the internal and external medial axes.

The article is organized as follows. Section 1 gives the information on the problem formulation. In Section 2 we discuss the choice of the mathematical model for two approaches. Section 3 contains information about the computational set-up, including details of the numerical method, computational meshes, mesh adaptation algorithm and its parallel implementation in corresponding subsections. Section 4 is devoted to the numerical results of aerodynamics and acoustics. Conclusion summarizes the study and points directions for further work.

1. Problem Formulation

The problem considered in this paper is similar to the APC Slow Flyer 10x4.7 problem, namely, the small-scaled UAV rotor problem studied in [10]. Similar to Ref. [11], the drone problem with the rotor and hub radii of $R = 0.127\ m$ and $r = 0.0127\ m$, respectively, is studied. The blade of the rotor is based on the Eppler E63 airfoil (blade inner part) and Clark-Y airfoil (near blade tip) with non-linear twist and chord spanwise distribution (Fig. 1). The propeller is designed for 2 000–20 000 rotations per minute (RPM), while the regime under consideration is 4 000 RPM, which corresponds to the tip velocity $V_{tip} = 53.2\ m/s$. In the computational setup, the size of the propeller is normalized by the maximum length of the chord of the blade $b = 0.0287\ m$. The Reynolds number, defined using the blade tip velocity and the blade maximum chord, is $Re = \rho_0 V_{tip} b/\mu_0 \approx 10^5$, where $\rho_0 = 1.204\ kg/m^3$ and $\mu_0 = 1.815 \times 10^{-5}\ Pa \cdot s$

corresponding to the parameters of the air at temperature of $T_0 = 293.15K$. The tip Mach number is $M = V_{tip}/\sqrt{\frac{\gamma \mathcal{R} T_0}{\mathcal{M}}} = 0.156$, where γ is the adiabatic index, $\mathcal{R}$ is the gas constant and $\mathcal{M}$ is the molar mass of the gas.

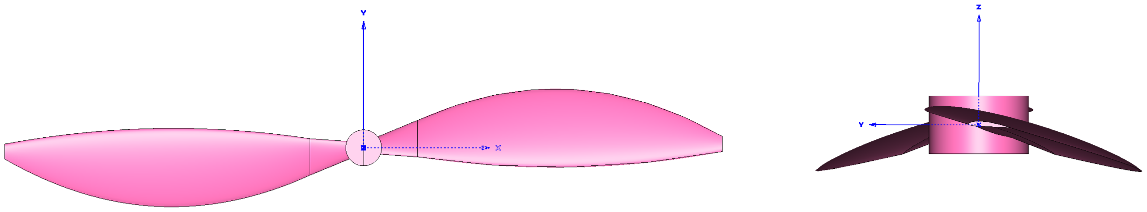

Figure 1. The rotor geometry

2. Mathematical Model

Large Reynolds number, high velocity flows are typical for aviation applications. Direct Numerical Simulation (DNS) of such flows requires high resolution computations both in space and time that are beyond the current realm despite the continuing growth of the performance of supercomputers. There is a number of approaches which can be applied to model turbulent flow near propeller, including RANS, LES and hybrid RANS-LES approaches. It should be mentioned that there is no full understanding yet that correct modeling of aerodynamics using presented technique is feasible. We are aiming to simulate rotor acoustics. So, differences that [12] presents by comparing RANS and RANS-LES for simulation of flow around rotating propeller do not make big differences in terms of our aims. For that reason, the turbulent flow simulation uses compressible Reynolds-averaged Navier–Stokes (RANS) equations with the Spalart–Allmaras turbulence model [13]. The system of RANS equations is written in the form of conservation laws for the vector $\mathbf{Q}$ of conserved variables

$$\mathbf{Q} = (\rho, \rho\mathbf{u}, E, \rho\tilde{\nu})^T,$$

where $\mathbf{u} = (u_1, u_2, u_3)$ is the velocity vector, ρ is the density, $E = \rho\mathbf{u}^2/2 + \rho\varepsilon$ is the total energy, ε is the specific internal energy, p is the pressure defined by ideal gas equation $p = \rho\varepsilon(\gamma - 1)$, $\gamma = 1.4$ is the adiabatic exponent, $\tilde{\nu}$ is the evolutionary variable which is used to determine the turbulent viscosity μ_T according to the Spalart–Allmaras model:

$$\mu_T = \rho\tilde{\nu}\frac{\chi^3}{\chi^3 + 357.911}, \quad \chi = \frac{\rho\tilde{\nu}}{\mu},$$

where μ is the coefficient of dynamic molecular viscosity.

The system of Reynolds-averaged Navier–Stokes equations can be written in the following vector form:

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

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

The convective transport flux vector is given as a function of the physical variables ρ, $\mathbf{u}$, p

$$\mathbf{F}_i^C(\mathbf{Q}) = (\rho u_i, \rho u_i \mathbf{u} + p\mathbf{I}, (E+p)u_i, \rho\tilde{\nu}u_i)^T, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. The diffusion flux vector is defined as a function of physical variables and their gradients as

$$\mathbf{F}_i^D(\mathbf{Q}, \nabla\mathbf{Q}) = \left(0,\ \tau_{i1},\ \tau_{i2},\ \tau_{i3},\ \tau_{ij}u_j + q_i,\ \frac{3}{2}(\mu + \rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_i}\right)^T, \tag{3}$$

where the components of the viscous tensor of viscous stresses and the heat flux vector can be written as follows:

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

where δ_{ij} is the Kronecker symbol, μ is the molecular viscosity coefficient, $\mathrm{Pr} = 0.72$ and $\mathrm{Pr}_T = 1$ are the molecular and turbulent Prandtl numbers, respectively.

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

$$\mathbf{S}(\mathbf{Q}, \nabla\mathbf{Q}) = (0,\ 0,\ 0,\ P_\nu(\mathbf{Q}, \nabla\mathbf{Q}) - Y_\nu(\mathbf{Q}, \nabla\mathbf{Q}) + 0.992\nabla\tilde{\nu}\cdot\nabla\tilde{\nu})^T. \tag{5}$$

The detailed definition of terms $P_\nu(\mathbf{Q}, \nabla\mathbf{Q})$, $Y_\nu(\mathbf{Q}, \nabla\mathbf{Q})$ describing respectively the generation and dispersion of turbulence is given in paper [13].

Let us formulate the immersed boundary condition for system (1). At the boundary between a solid Ω_B and a medium Ω_f there is a no-slip condition:

$$u|_{\partial\Omega_B} = V. \tag{6}$$

Condition (6) is defined by Brinkman penalization method [1]. The Brinkman penalization modifies the right-hand side of the system (1) by adding the extra source terms so that the new vector of source term becomes as:

$$\mathbf{S}^{penal}(\mathbf{Q}, \nabla\mathbf{Q}) = \mathbf{S}(\mathbf{Q}, \nabla\mathbf{Q}) + \left(0,\ \frac{\chi}{\eta}\rho(u_i - u_{Bi}),\ \frac{\chi}{\eta}\rho u_i(u_i - u_{Bi}),\ 0\right), \tag{7}$$

where χ defines the body location at every time moment as follows:

$$\chi(t) = \begin{cases} 1,\ x \in \overline{\Omega}_B(t) \\ 0,\ x \in \Omega_f(t). \end{cases} \tag{8}$$

Parameter η determines the rate of the relaxation of the flow velocity to the velocity of the moving body. In this paper the penalization parameter is equal to 10^{-4}. When we use the standard body-fitted approach to simulate the flow over the rotating propeller we solve the RANS equations in non-inertial frame of reference [14]. Let $\mathbf{V} = (V_1, V_2, V_3)^T = (\boldsymbol{\omega} \times \mathbf{r})$ be the peripheral propeller speed determined by angular velocity vector $\boldsymbol{\omega}$ and radius vector of the point in medium. Following this notation, the system of RANS equations (1) can be rewritten as

$$\frac{\partial\mathbf{Q}}{\partial t} + \nabla\cdot(\mathcal{F}^C(\mathbf{Q}) - \mathcal{F}^R(\mathbf{Q}) - \mathcal{F}^D(\mathbf{Q},\nabla\mathbf{Q})) = \tilde{\mathbf{S}}(\mathbf{Q},\nabla\mathbf{Q}), \tag{9}$$

where $\mathcal{F}^R$ is the rotation transport flux with the components

$$\mathbf{F}_i^R(\mathbf{Q}) = (\rho V_i,\ \rho u_i \mathbf{V},\ E V_i,\ \rho\tilde{\nu} V_i)^T. \tag{10}$$

Source term (5) is now dependent on angular velocity:

$$\tilde{\mathbf{S}}\left(\mathbf{Q}, \nabla\mathbf{Q}\right) = (0,\ \rho\,\boldsymbol{\omega} \times \mathbf{u},\ 0,\ P_\nu\left(\mathbf{Q}, \nabla\mathbf{Q}\right) - Y_\nu\left(\mathbf{Q}, \nabla\mathbf{Q}\right) + 0.992 \nabla\tilde{\nu} \cdot \nabla\tilde{\nu})^T. \tag{11}$$

From the point of view of an observer in the inertial coordinate system, the system of equations (9)–(11) with flux vectors (2) and (3) describes the change in conservative variables due to their convective and diffusive transport in a medium rotating at speed $\mathbf{V}$, the influence of the pressure gradient and the rotation of the velocity vector at the azimuthal angle $\psi(t) = -\left|\boldsymbol{\omega}\right| t$.

3. Computational Set-up

3.1. Numerical Method

The system of RANS equations (1) is solved using the fifth order Edge-based Reconstruction (EBR5) scheme [15–17]. The higher accuracy is achieved through the quasi-one-dimensional reconstructions of variables on the extended stencils oriented along the mesh edges. The approximation is built in such a way that being applied to translationally invariant meshes (i.e., meshes that transform into themselves when translated by an edge vector) the EBR5 scheme provides the fifth order of accuracy.

The EBR scheme for the Euler equations presents a vertex-centered method with physical and conservative variables defined at the mesh vertices around which the computational cells of the dual mesh are built. Thus, a finite-volume approximation of convective flows is constructed for dual mesh cells acting as finite volumes. The viscous terms of the RANS equations are approximated using the finite-element method with the linear basis functions.

For the time integration, the implicit second-order scheme with Newton iterations is used to solve the nonlinear system of algebraic equations resulting from the space discretization. At each Newtonian iteration, the corresponding system of linear equations is solved using the stabilized biconjugate gradient method.

3.2. Dynamic Mesh Redistribution Method

To build a dynamic mesh redistribution method, we consider a computational mesh as an elastic material which is deformed when zones of mesh compression follow the boundaries of moving obstacles. The time-dependent elastic deformation is defined by a special mapping $\mathbf{x}(\boldsymbol{\xi}, t) : R^d \times R \to R^d$. Let C define the Jacobian matrix of mapping $\mathbf{x}(\boldsymbol{\xi}, .)$, where $c_{ij} = \dfrac{\partial x_i}{\partial \xi_j}$. Let $\mathbf{x}^n(\boldsymbol{\xi})$ denote the mapping of the initial mesh onto the mesh at time t^n.

We consider the mapping of a regular tetrahedron T_i with the vertices coordinates $\mathbf{h_0}$, $\mathbf{h_1}$, $\mathbf{h_2}$, $\mathbf{h_3}$ in Lagrangian domain $\Omega_{\boldsymbol{\xi}}$ to a tetrahedron with Eulerian coordinates of the vertices $\mathbf{p_0}$, $\mathbf{p_1}$, $\mathbf{p_2}$, $\mathbf{p_3}$ in domain $\Omega_{\mathbf{x}}$. The Eulerian coordinates are the desired coordinates in the computational domain. For our problem formulation $\Omega_{\boldsymbol{\xi}}$ is some speculative domain that consists of regular tetrahedrons and $vol(\Omega_{\boldsymbol{\xi}}) = vol(\Omega_{\mathbf{x}})$. At the time moment t for each vertex of the tetrahedron Jacobian matrix of this mapping can be written as $\nabla_{\boldsymbol{\xi}}\mathbf{x}(\boldsymbol{\xi}, t)\mathbf{H}^{-1}$, where $\nabla_{\boldsymbol{\xi}}\mathbf{x}(\boldsymbol{\xi}, t) = (\mathbf{p_1} - \mathbf{p_0},\ \mathbf{p_2} - \mathbf{p_0},\ \mathbf{p_3} - \mathbf{p_0})$ and $\mathbf{H} = (\mathbf{h_1} - \mathbf{h_0},\ \mathbf{h_2} - \mathbf{h_0},\ \mathbf{h_3} - \mathbf{h_0})$. Here $\boldsymbol{\xi}$ are local barycentric coordinates inside parametric tetrahedron.

To keep track of moving body we consider metric in Eulerian coordinates which depends on body geometry and time. Let $\mathbf{G}(\mathbf{x}, t)$ denote metric tensor and $\mathbf{Q} = \mathbf{Q}(\mathbf{x}, t)$ be an arbitrary matrix factorization of metric tensor $\mathbf{G}(\mathbf{x}, t)$, defined by

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{G}(\mathbf{x}, t), \ \det \mathbf{Q} > 0.$$

To introduce the variational problem for mesh optimization one can assume that $\mathbf{x}^n(\boldsymbol{\xi})$ is a quasi-isometric piecewise affine one-to-one mapping. We find the mapping of the initial mesh onto the mesh at time $t = t^{n+1}$ as minimizer of this semi-discrete functional

$$F(\mathbf{x}(\boldsymbol{\xi}, t), \mathbf{x}^n(\boldsymbol{\xi})) = \sum\nolimits_i \int\limits_{T_i} W(\mathbf{Q}(\mathbf{x}^n(\boldsymbol{\xi}), t)\nabla_{\boldsymbol{\xi}}\mathbf{x}(\boldsymbol{\xi}, t)\mathbf{H}^{-1}) \det \mathbf{H} d\boldsymbol{\xi}, \tag{12}$$

where $\mathbf{C} = \mathbf{Q}\nabla_{\boldsymbol{\xi}}\mathbf{x}(\boldsymbol{\xi}, t)\mathbf{H}^{-1}$ is the Jacobian matrix of mapping from regular tetrahedron T_i to i-th tetrahedron in metric space.

In functional (12) function $W(\mathbf{C})$ defines polyconvex elastic potential (internal energy), which is a weighted sum of the shape distortion measure and the volume distortion measure:

$$W(\mathbf{C}) = (1-\theta)\frac{\frac{1}{d}\operatorname{tr}(\mathbf{C}^T\mathbf{C})}{\det \mathbf{C}^{2/d}} + \frac{1}{2}\theta\left(\frac{1}{\det \mathbf{C}} + \det \mathbf{C}\right). \tag{13}$$

In most cases we set $\theta = 4/5$. The elastic potential is minimum when the deformation is isometric, meaning that only rotation and translation are allowed. The details of the current variational approach are written in [18].

To use the mesh vertices effectively, the adaptation should be anisotropic. At the same time, it is necessary to qualitatively capture the surface of the body and its features. The correct definition of the adaptation metric requires normal and tangential directions at each point $\mathbf{p}$ which are defined by the isosurface of signed distance function $u(\mathbf{x}, t)$ passing through $\mathbf{p}$ and stretching coefficients σ_i along these directions. When σ_i is large, the local cell size in correspondent direction is small. Function $\sigma_1 = \sigma_{normal}(\mathbf{x}, t) = \phi(u(\mathbf{x}, t))$ defines the mesh stretching in the normal direction to the body and $\sigma_2 = \sigma_{2,tangential}(\mathbf{x}, t)$ and $\sigma_3 = \sigma_{3,tangential}(\mathbf{x}, t)$ define the spatial distribution of the anisotropy. We require the highest anisotropy in a thin boundary layer near the body, then the anisotropy is partially reduced to zero away from the body. Here 1D function $\phi(\cdot) : R^1 \to R^1$ is a hyperbola, which defines the mesh density in normal direction and guarantees transition from small mesh cells to large mesh cells with the prescribed rate of growth of mesh cell size. The definitions of σ_2 and σ_3 require extra information about the body shape such as principal curvatures, principal normal directions, and distances to the medial axis. All this volumetric data is gained during the preprocessing stage. Parameters σ_2 and σ_3 are used for defining of the control metric, so that mesh cells are close to isotropic near the sharp edges and anisotropic in the tangential surface near the regions close to a plane. The complete procedure of geometry preparation and the algorithm for constructing the anisotropic mesh adaptation metric is described in detail in [19].

We solve our adaptation mesh problem in a small cylinder that encompasses propeller geometry. Vertices on the boundary of this small cylinder are fixed. To achieve a good mesh refinement at all the edges of the propeller geometry and provide continuity of point distribution through the fixed boundary, the initial mesh is specially deformed. The vertices in the small cylinder are slightly redistributed to follow up the features of the body shape: an area near the hub and the region where the tip of the blade moves have higher mesh density as shown in Fig. 2. To account for that nonuniformity, the background metric is introduced. The background metric is a spherical metric where the eigenvalues are defined as cubic root of a ratio of average adjacent cell volume in parametric space to the correspondent volume in physical space. So now there are two metrics: background metric and adaptation metric $\mathbf{G_x}$. For the metric interpolation the log-Euclidian framework [20] is used. The paper [20] proposes introducing the metric logarithm and the commutative logarithm addition.

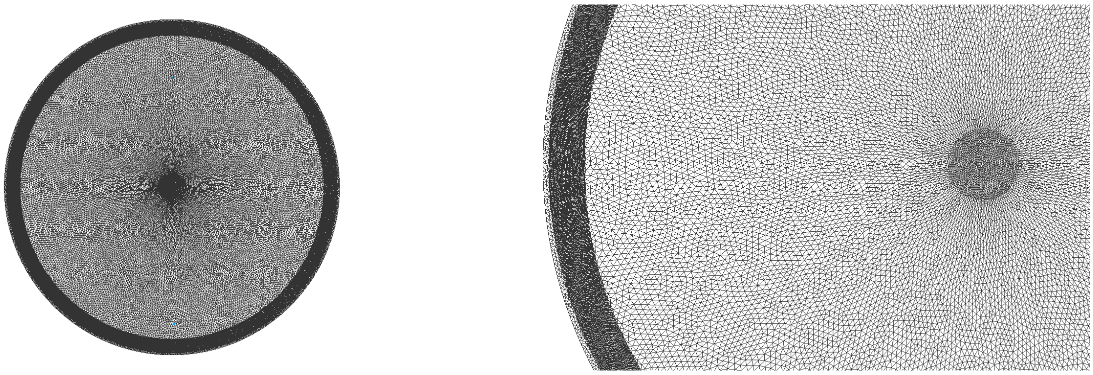

Figure 2. The initial mesh in a puck-shaped domain to start the adaptation to the propeller surface

Metric tensor $\mathbf{G_x}$ is evaluated in the mesh vertices which is crucial to the stability of very thin and highly compressed mesh layers. To solve the optimization problem on the each time step, we apply the preconditioned gradient descent technique [21] where the minimization direction is computed via an approximate solution of the linear system with the reduced Hessian matrix of the functional. The final mesh increment along the minimization direction is computed via the 1D search technique. The full description of the variational approach of the adaptation algorithm is presented in [18].

3.3. Computational Meshes

3.3.1. Mesh for body-fitted approach

The computational mesh for the body-fitted approach is a cylinder with radius $10R$ and height $30R$ (Fig. 3). R is the propeller radius. Propeller was placed in the center of the computational domain. The mesh near the rotor surface is filled with prismatic elements to resolve the boundary layer. A height of the near-surface element is chosen to meet $y^+ < 1$ criteria in the CFD simulation, meaning the viscous sublayer is resolved. y^+ is the dimensionless wall distance defined as $y^+ = \dfrac{yu_\tau}{\nu}$, where $u_\tau = \sqrt{\dfrac{\tau_\omega}{\rho}}$ is friction velocity, τ_ω is the wall shear stress, y is the absolute distance and ν is the kinematic viscosity.

The rest space between prismatic mesh on the rotor surface and the outer domain boundaries is filled with tetrahedrons. As a result, the unstructured mixed-element mesh is build with 2.6 million nodes and 9.9 million elements.

3.3.2. Adaptive mesh for immersed boundary method

To facilitate the mesh adaptation with the metric depending on the distance function the triangulated surface of the propeller is prepossessed and a k-d tree and an octree are built for

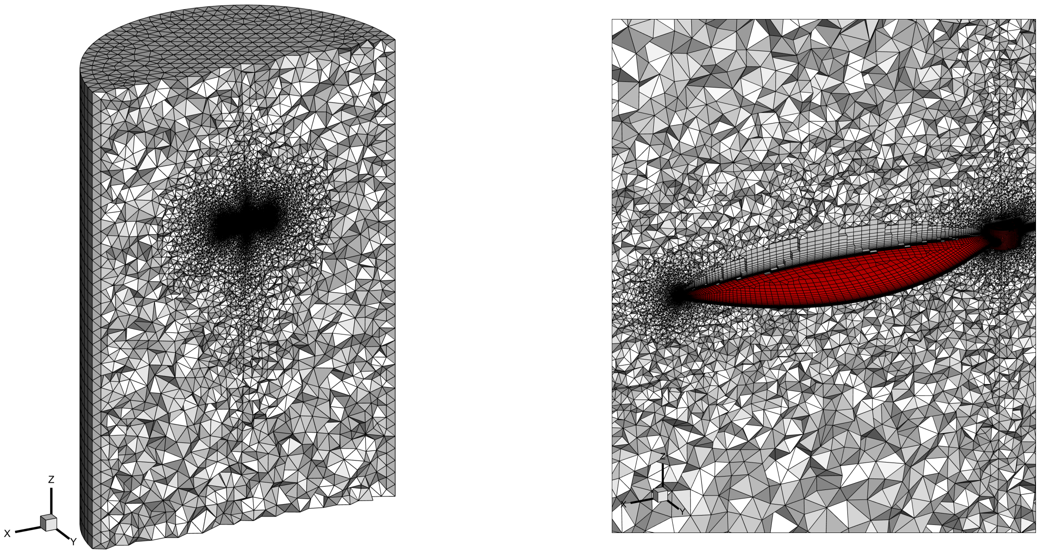

Figure 3. The mesh for the BF approach

fast calculations of the exact and approximate signed distance values, respectively. When possible approximate distance is used, which is calculated by interpolating exact values stored in the octree nodes [22]. In addition to interpolation of the signed distance function the octree is also used to access some shape parameters of the propeller such as principal curvatures, principal normal directions, and distances to internal (Fig. 4 and Fig. 6) and external (Fig. 5 and Fig. 7) medial axes, which are used to define the adaptation control metric. These parameters are found at the vertices of the triangulated surface and then they are extrapolated to the nodes of the interpolation grid (octree) based on the nearest distance. The algorithm for the curvature calculation is described in [23]. The medial axes are approximated using the PowerCrust algorithm [24].

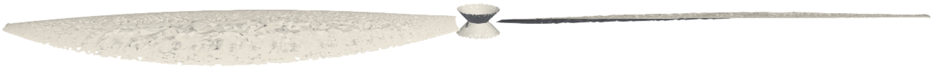

Figure 4. The approximate internal medial axes

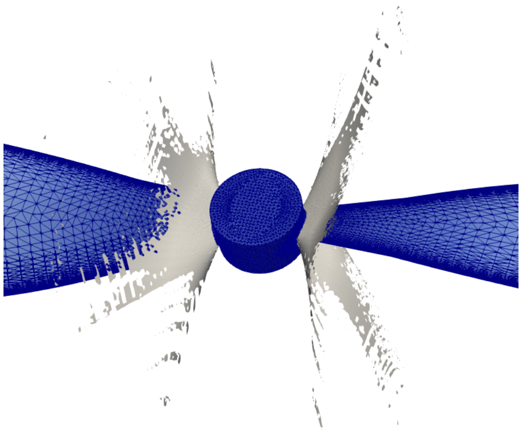

Figure 5. The approximate external medial axes

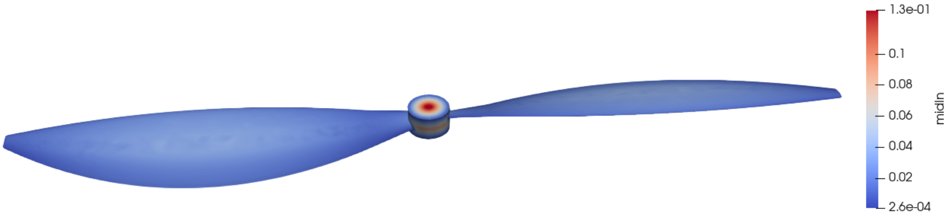

Figure 6. The distances from the surface vertices to the internal medial axes

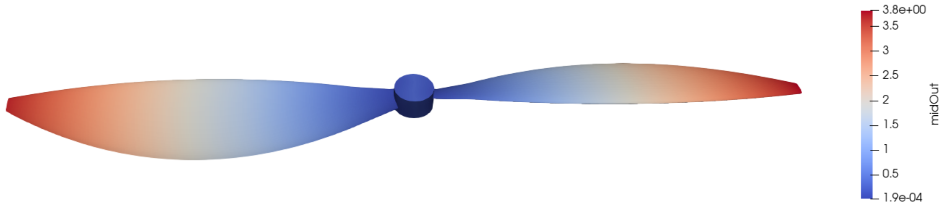

Figure 7. The distances from the surface vertices to the external medial axes

The initial tetrahedral mesh is built in a cylinder-shaped domain with same sizes as in the BF approach with large number of vertices placed near the propeller surface. The total mesh size is 4.4 million vertices with the majority of vertices concentrated in a puck-shaped domain covering the propeller motion region.

The propeller geometry is virtually impossible to reproduce correctly on a simply connected mesh without grid adaptation, since the resolution of thin blade edges requires a significant mesh refinement in the vicinity of the propeller surface.

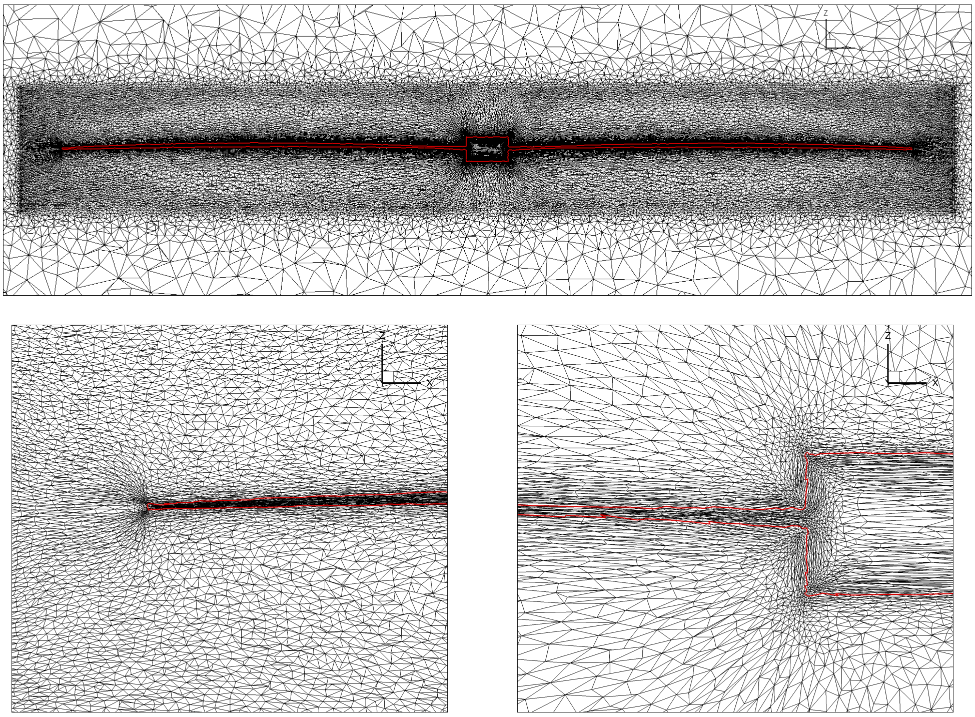

Figure 8. XZ plane section of the DMR-IBM simulation mesh near the propeller (top) and fragments of the adapted mesh near the hub and near the blade tip (bottom)

The mesh redistribution method is then applied to refine the grid to approximately one tenth of the initial mesh size. For the resulting mesh $y^+ \sim 200$, which corresponds to the log-law region. Figure 8 shows a fragment of the adaptive mesh in XZ section and a more detailed image of the mesh near geometry features: the hub and its connection with the blade and the end of the blade. In the presented figures all the points belonging to the body are circled in red. Mesh elements sizes depend on the shape of the body: the mesh is more isotropic near corners and in high curvature regions. Figure 9 presents the body shape in DMR-IBM simulation.

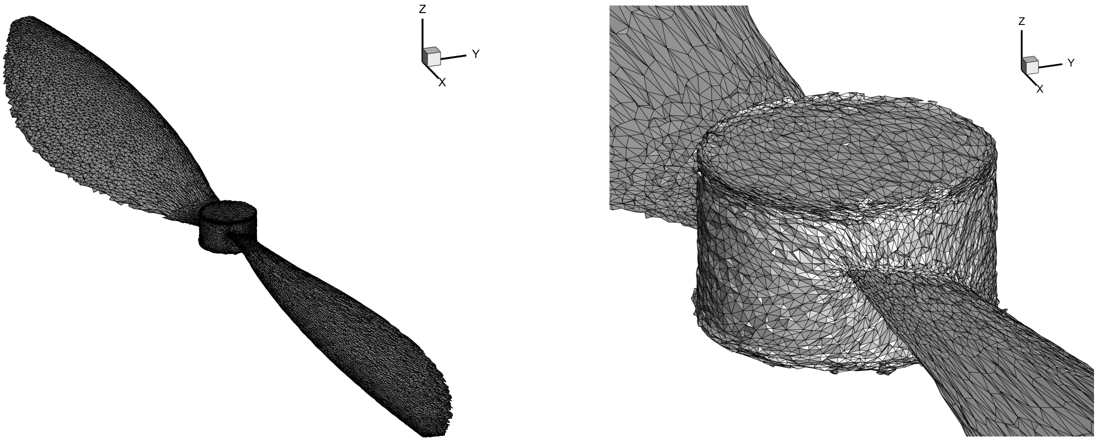

Figure 9. The propeller shape in the DMR-IBM simulation (only the cells with all the vertices inside the body are drawn)

The application of mesh redistribution method is complicated by the presence of thin edges. The distance between the walls of the blade is extremely small and sharp corners produced by these walls are not always well refined using the developed mesh adaptation approach. In the current paper the problem of narrow walls was practically avoided by using a slightly anisotropic initial mesh inside a small cylinder. However, the mesh adaptation algorithm to very narrow parts would need to be further investigated.

3.4. Technology of Acoustics Measurements

To measure and compare the propeller acoustics characteristics in the near field, a set of probes are placed around the rotor. The probes are radially distributed with angular step 10° in plane of rotor rotation (XY) and in XZ plane (Fig. 10). The YZ plane is not considered since, due to the symmetry of the case setup, it is equivalent to the XZ plane.

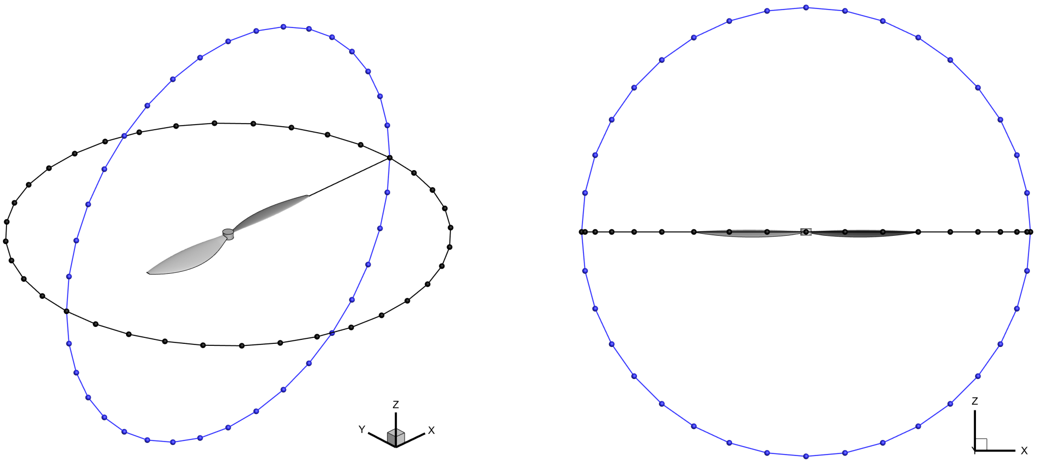

Figure 10. Acoustic probes location: XY plane (black) and XZ plane (blue)

Thus, the following four sets of probes are used: two sets at planes XY, XZ at distance $2R$ from rotor center and two sets at planes XY, XZ at distance $3R$ from rotor center. The azimuthal angle in plane XZ is measured from Ox axis: positive azimuthal angle corresponds to the upper hemisphere and the negative – to the lower (downstream) one.

3.5. Parallel Implementation

All algorithms are implemented in the CFD in-house code NOISEtte. It has MPI+OpenMP parallelization for supercomputers made of multi- and manycore processors. A detailed description of the parallel algorithm is given in [25]. The dynamic mesh adaptation also follows MPI+OpenMP parallelization model. The parallel algorithm uses spacial mesh decomposition. Since degrees of freedom which define mesh deformation are mesh vertices, we build a consistent partitioning of mesh cells and vertices: computational domain is split into connected subdomains consisting of full mesh cells. Mesh vertices belonging to boundaries between subdomains are distributed between subdomains. Parallel BiCGStab-based iterative solver with ILU2 preconditioner is used [21]. Input data for this solver are right hand side partitioned into blocks and sparse matrix partitioned into block rows. Each block precisely corresponds to the partitioning of mesh vertices. The implementation of iterative scheme is based on extended subdomains defining two cell-wide subdomain overlap. We have found that one minimization iteration was enough in order to make mesh precisely follow domain boundaries with prescribed compression. Moreover, it was found that mesh generator can predict mesh deformation with time step exceeding that of RANS solver. In that case on the intermediate time steps mesh is obtained using interpolation technique without any high-load operations. The interpolation sharply improves computational efficiency of mesh deformation solver, which is described in [26].

It should be mentioned that the CFD/RANS simulation of the current problem allows for using the time step exceeding maximal adaptation step. Thus, for the problem under consideration, the adaptation solver was called at each CFD step, while the size of this CFD step was restucted to meet the requirements of the adaptation method. This fact coupled with non-optimal parallel implementation of adaptation solver resulted in rather low efficience of DMR-IBM solution. So far, the current performance of the DMR-IBM method is 8.7 times more expensive than the BF-approach on identically sized meshes with the same time step. The optimized parallel implementation is currently ongoing.

4. Numerical Results

4.1. Validation of BF and DMR-IBM Results on Aerodynamic Characteristics

The flow field in the both DMR-IBM and BF simulations looks similar Fig. 11. There is a jet-type flow downstream the rotor induced by the rotation with the maximum velocity below the blade tips. The tip vortices are well resolved up to 1.5 revolutions and, due to the high mesh resolution in the propeller rotation region in the DMR-IBM case, the tip vortex in the DMR-IBM simulation is resolved better than in the BF case.

In both BF and DMR-IBM simulation the rotor thrust coefficient c_t and power coefficient c_p are compared against the experimental data [11].

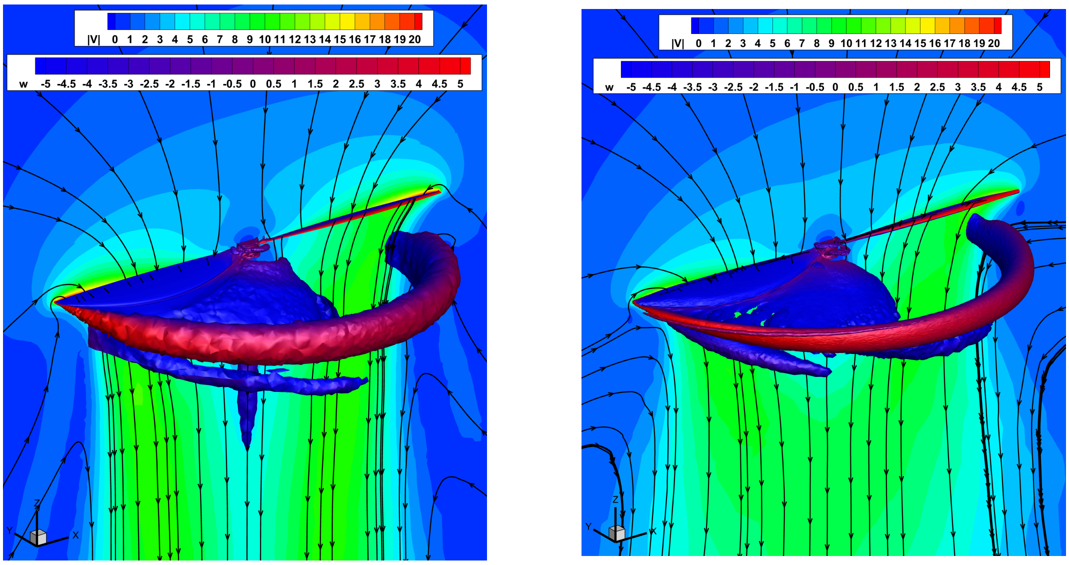

Figure 11. The flow field: velocity magnitude, streamlines in ZX section with Q-criterion isosurfaces for the BF case (left) and DMR-IBM case (right)

Table 1. Aerodynamic coefficients comparison with the reference data

	Reference	BF	δ, %	DRM-IBM	δ, %
Thrust coefficient c_t	0.1158	0.1178	1.7	0.0947	–18.3
Power coefficient c_p	0.0466	0.0465	–0.2	0.0451	–3.2

The result of comparison of aerodynamic coefficients with the experimental data is presented in the Tab. 1 where the relative difference is calculated as $\delta = \left(c_{t|p} - c^{ref}_{t|p}\right)/c^{ref}_{t|p} \cdot 100\%$. As seen from the Tab. 1, the aerodynamics coefficients are noticeably underestimated in the DMR-IBM simulation. It may be a result of a poor mesh resolution of the boundary layer on the rotor blade surface. Another possible reason of this shortcoming could come from not taking into account the Lagrangian convection in the Brinkman penalization term in (7).

4.2. Comparative Analysis of Acoustic Characteristics

As mentioned above, the primary goal of the work is to study the near-field acoustics generated by the rotating propeller modelled using BF and DMR-IBM approaches. Among the acoustic characteristics, we consider the spectra of pressure pulsation and the directivity diagrams of the pressure pulsation obtained using the above-described probes at the blade passing frequency (BPF). Figure 12 represents the specific spectra of pressure pulsations in two probes – one in the plane of rotation XY with azimuthal position 0° (Fig. 12, left) and the other in the plane XZ with azimuthal position 40°. The presented spectra are built with sampling frequency 6.65 Hz.

As expected, the spectra maxima in both BF and DMR-IBM simulations are reached at the BPF frequency and its multiples. The difference between the BF and DMR-IBM results at first BPF amplitude is about 2.1 dB for the first probe and 1dB for the second one. It is noticeable that the amplitudes of the harmonics that are multiples of the BPF frequencies in the rotation plane obtained in the DMR-IBM simulation are much closer to the results of the BF simulation than in the other azimuthal position.

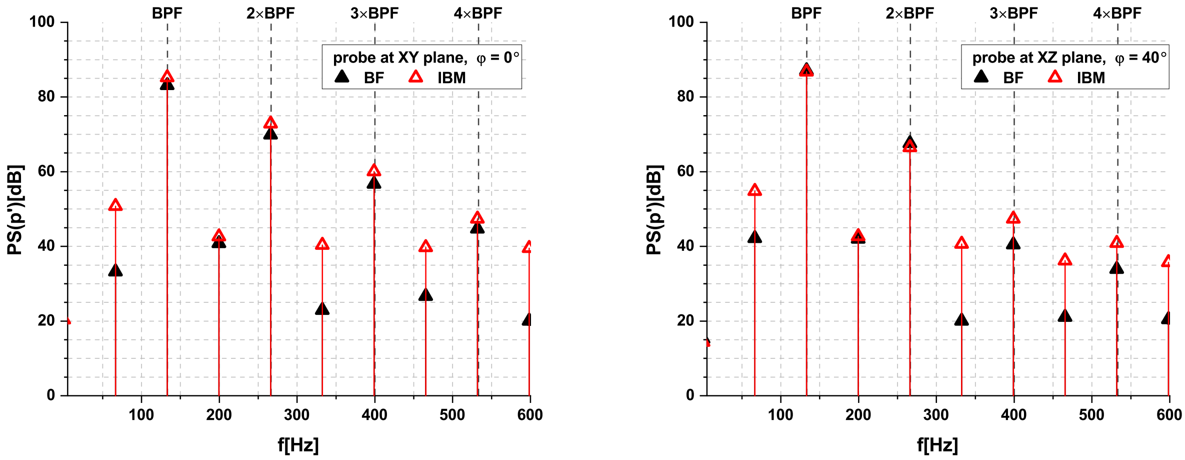

Figure 12. The spectra of pressure pulsation at the probes XY plane azimuth 0°(left), XZ plane azimuth 40° (right)

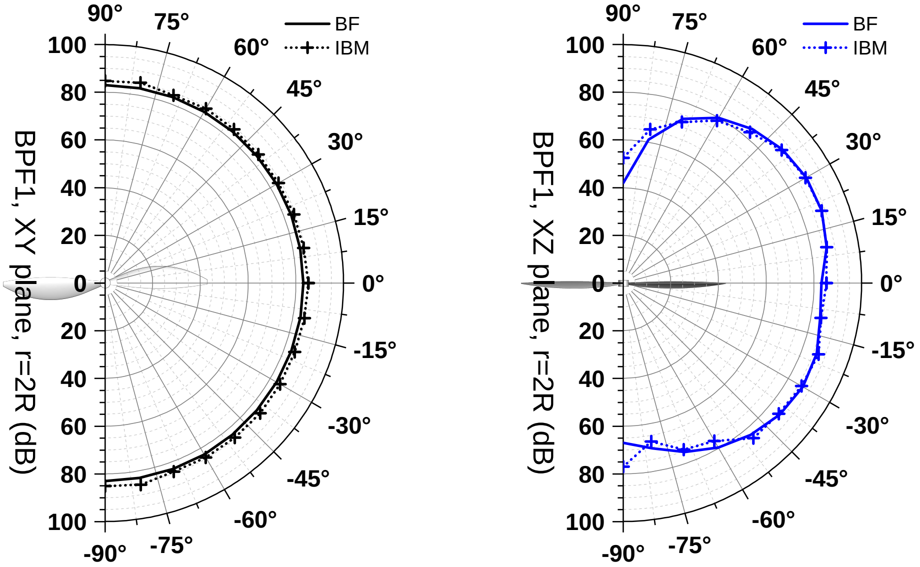

Figure 13. The directivity diagram of the first BPF pressure pulsation at the probes at distance $2R$

In Fig. 13 and Fig. 14 the directivity diagram of the first BPF for the XY and XZ planes are presented for distances $2R$ and $3R$. It is seen that in the plane of rotation the maximum difference between BF and DMR-IBM computation is less than 2.9 dB for $2R$ and 2.2 dB for $3R$. On the other hand, in the XZ plane for azimuthal positions $-75° < \varphi < 75°$ the maximum difference is less than 2 dB for both distances. It should be noted that the pressure pulsation measured at the rotation axes should theoretically be zero. In our case, it is not so (it is about 40–60 dB) solely because of the asymmetric setup. This fact is confirmed by our simulation of a single blade with the periodicity condition in azimuthal direction where it is really zero (see Ref. [14]). Note also that in the DMR-IBM simulation the asymmetric setup is aggravated by the slightly dynamically changing geometry of the blade.

Conclusion

The paper discusses a feasibility of simulating turbulent flow around rotating bodies of complex geometry and the associated acoustics using the developed hybrid dynamic mesh re-

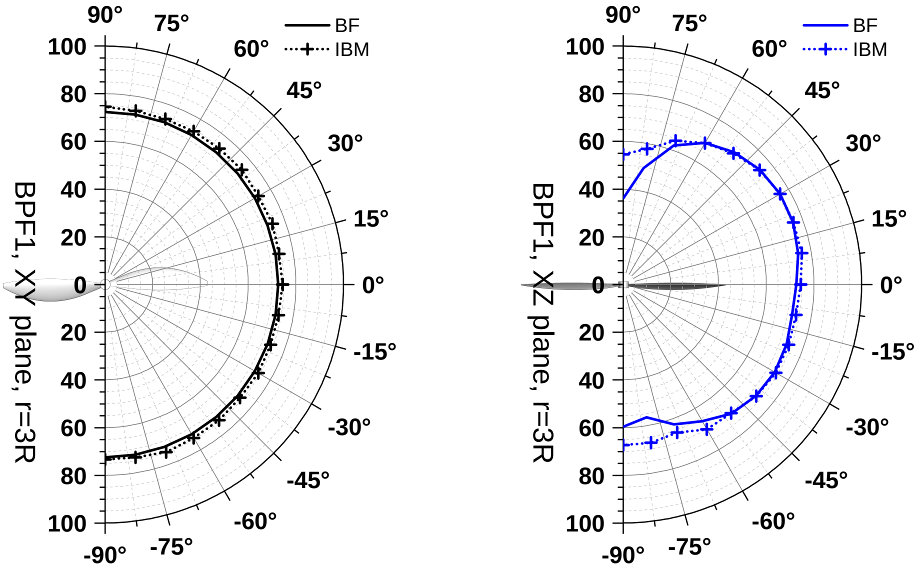

Figure 14. The directivity diagram of the first BPF pressure pulsation at the probes at distance $3R$

distribution – immersed boundary method with mesh metric based on distance function to the obstacle surface. In this paper, we consider the problem of simulation of aeroacoustics of a drone propeller taken as an example of a rotating body of complex geometry. In addition to evaluating the developed method on a representative case, the problem of a single-rotor acoustics has an obvious extension important for a wide range of applications. The current results can be viewed as building block for simulation of the acoustics of multi-rotor machines, the use of which in the daily life is steadily increasing. Although in terms of tonal acoustics the results obtained by the developed hybrid dynamic mesh redistribution – immersed boundary method look promising, the quality of aerodynamic results is not satisfactory and calls for further investigation. One direction that can improve the results of the DMR-IBM simulations to increase the near wall mesh resolution to better represent the immersed solid. Another area of future research is to use wall-functions to decrease the near-wall mesh resolution requirements of the DMR-IBM simulations.

Acknowledgements

The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University and the hybrid supercomputer K60 installed in the Supercomputer Centre of Collective Usage of KIAM RAS. The work was funded by the Russian Science Foundation within Project No. 20-41-09018.

References

1. Abalakin, I.V., Zhdanova, N.S., Kozubskaya, T.K.: Immersed boundary method implemented for the simulation of an external flow on unstructured meshes. Math Models Comput Simul. 8,

219–230 (2016). https://doi.org/10.1134/S2070048216030029

2. Zhdanova, N.S., Gorobets, A.V., Abalakin, I.V.: Supercomputer simulations of fluid-structure interaction problems using an immersed boundary method. Supercomputing Frontiers and Innovations 5(4), 78–82 (2018). https://doi.org/10.14529/jsfi180408

3. Abalakin, I.V., Bahvalov, P.A., Doronina, O.A., *et al.*: Simulating Aerodynanics of a Moving Body Specified by Immersed Boundaries on Dynamically Adaptive Unstructured Meshes. Math. Models Comput. Simul. 11(1), 35–45 (2019). https://doi.org/10.1134/S2070048219010034

4. Abalakin, I.V., Duben, A.P., Zhdanova, N.S., *et al.*: Immersed Boundary Method on Deformable Unstructured Meshes for Airfoil Aeroacoustic Simulation. Comput. Math. and Math. Phys. 59(12), 2046–2059 (2019). https://doi.org/10.1134/S0965542519120029

5. Tsvetkova, V.O., Abalakin, I.V., Bobkov, V.G., *et al.*: Simulation of the Flow near a Rotating Propeller on Adaptive Unstructured Meshes Using the Immersed Boundary Method. Math. Models Comput. Simul. 14(2), 224–240 (2022). https://doi.org/10.1134/S2070048222020168

6. 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. Math Models Comput Simul. 13, 986–1001 (2021). https://doi.org/10.1134/S2070048221060028

7. Dervieux, A., Mesri, Y., Alauzet, F., *et al.*: Continuous mesh adaptation models for CFD. Computational fluid dynamics journal 16(4), 346–355 (2008).

8. Tang, J., Cui, P., Li, B., *et al.*: Parallel hybrid mesh adaptation by refinement and coarsening. Graphical Models 111, 101084 (2020). https://doi.org/10.1016/j.gmod.2020.101084

9. Steger, J., Dougherty, F.C., Benek, J.A.: A chimera grid scheme. Advances in Grid Generation 5, 59–69 (1983).

10. Brandt, B.: Small-scale propeller performance at low speed, Master thesis, University of Illinois at Urbana-Champaign, 2005.

11. Brandt, J.B., Selig, M.S.: Propeller performance data at low Reynolds numbers. AIAA Paper 2011-1255 (2011).

12. Bobkov, V., Gorobets, A., Kozubskaya, T., *et al.*: Supercomputer Simulation of Turbulent Flow Around Isolated UAV Rotor and Associated Acoustic Fields. In: Voevodin, V., Sobolev, S. (eds) Supercomputing. RuSCDays 2021. Communications in Computer and Information Science, vol. 1510. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92864-3_20

13. Spalart, P.R., Allmaras, S.R.: A One-Equation Turbulence Model for Aerodynamic Flows. 30th Aerospace Sciences Meeting & Exhibit, January 6–9, 1992, Reno, NV, AIAA Paper 92-0439. https://doi.org/10.2514/6.1992-439

14. 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 (2016). https://doi.org/10.1134/S0015462816030145

15. Abalakin, I., Bakhvalov, P., Kozubskaya, T.: Edge-based reconstruction schemes for unstructured tetrahedral meshes. Int. J. Numer. Meth. Fluids 81(6), 331–356 (2016). https://doi.org/10.1002/fld.4187

16. Bakhvalov, P.A., Kozubskaya, T.K.: Construction of edge-based 1-exact schemes for solving the Euler equations on hybrid unstructured meshes. Comput. Math. Math. Phys. 57(4), 680–697 (2017). https://doi.org/10.1134/S0965542517040030

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

18. Garanzha, V., Kudryavtseva, L.: Hypoelastic Stabilization of Variational Algorithm for Construction of Moving Deforming Meshes. In: Evtushenko Y., Jacimovic M., Khachay M., *et al.* (eds) Optimization and Applications. OPTIMA 2018. Communications in Computer and Information Science, vol. 974, pp. 497–511. Springer (2019). https://doi.org/10.1007/978-3-030-10934-9_35

19. Kozubskaya, T.K., Kudryavtseva, L.N., Tsvetkova, V.O. Anisotropic Adaptation of Moving Unstructured Mesh to Bodies of Complex Shapes Described by an Interpolation Octree. Comput. Math. and Math. Phys. 62, 1590–1601 (2022). https://doi.org/10.1134/S0965542522100074

20. Frazza, L.: 3D anisotropic mesh adaptation for Reynolds Averaged Navier–Stokes simulations. Modeling and Simulation. Sorbonne Université, thesis (2018)

21. Kaporin, I.E., Milyukova, O.Yu.: MPI+OpenMP implementation of the BiCGStab method with explicit preconditioning for the numerical solution of sparse linear systems. Numerical methods and programming 20, 516–527 (2019). https://doi.org/10.26089/NumMet.v20r445

22. Soukov, S.A.: Combined signed distance calculation algorithm for numerical simulation of physical processes and visualization of solid bodies movement. Scientific Visualization 12(5), 86–101 (2020). https://doi.org/10.26583/sv.12.5.08

23. Garimella, R., Swartz, B.: Curvature Estimation for Unstructured Triangulations of Surfaces (2015)

24. Amenta, N., Choi, S., Kolluri, R.K.: The power crust. Proceedings of the sixth ACM symposium on Solid modeling and applications (SMA '01), pp. 249–266 (2001) https://doi.org/10.1145/376957.376986

25. Gorobets, A.V.: Parallel Algorithm of the NOISEtte Code for CFD and CAA Simulations. Lobachevskii Journal of Mathematics 39(4), 524–532 (2018). https://doi.org/10.1134/S1995080218040078

26. Kozubskaya, T., Kudryavtseva, L., Tsvetkova, V.: Unstructured Mesh Adaptation for Moving Bodies in Immersed Boundary Methods. 14th WCCM-ECCOMAS Congress 2020, January 11–15, 2021. https://doi.org/10.23967/wccm-eccomas.2020.353