

Article

Modeling Object Motion on Arbitrary Unstructured Grids Using an Invariant Principle of Computational Domain Topology: Key Features

Aleksey Sarazov ^{1,2}, Andrey Kozelkov ^{1,2,3} , Dmitriy Strelets ^{3,*}  and Roman Zhuchkov ^{1,2}

¹ Russian Federal Nuclear Center, All-Russian Research Institute of Experimental Physics, Nizhny Novgorod Region, Sarov 607188, Russia; askozelkov@mail.ru (A.K.)

² Department of Applied Mathematics, Nizhny Novgorod State Technical University n.a. R.E. Alekseev, 24 ul. Minina, Nizhny Novgorod 603155, Russia

³ Aircraft Design and Certification Department, Moscow Aviation Institute, Volokolamskoe Shosse, 4, Moscow 125993, Russia

* Correspondence: maksmai33@gmail.com

Abstract: This paper uses a finite volume algorithm to address the numerical modeling of fluid flow around moving bodies. The Navier–Stokes equations, which describe the flow of viscous compressible gas, along with key boundary conditions and discretization schemes, are presented. As the motion of boundaries typically leads to changes in the control volumes, the basic discretization schemes need to be adapted. This paper provides a detailed discussion on the adaptation of the initial system to deforming boundaries while preserving communication topology. The method for calculating the boundary velocity is a crucial element of the numerical scheme. The paper proposes an approach to reconstruct the boundary velocity vector using deformation analysis and the condition of geometric conservation. This approach ensures correct simulation results for arbitrary unstructured computational grids. A comparison of two approaches to reconstructing the boundary velocity vector for characteristic aviation problems in the direct formulation is presented. It is shown that the proposed approach allows for more accurate modeling of object motion on arbitrary grids using the “invariant” principle of the computational domain topology.

Keywords: computational fluid dynamic; unstructured grids; deformation; geometric conservation law



Citation: Sarazov, A.; Kozelkov, A.; Strelets, D.; Zhuchkov, R. Modeling Object Motion on Arbitrary Unstructured Grids Using an Invariant Principle of Computational Domain Topology: Key Features. *Symmetry* **2023**, *15*, 2081. <https://doi.org/10.3390/sym15112081>

Academic Editor: Toshio Tagawa

Received: 22 August 2023

Revised: 13 October 2023

Accepted: 17 October 2023

Published: 17 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Problems in hydro-aerodynamics that involve the motion or shape alteration of boundaries require special attention to formulate correct boundary conditions for numerical simulations [1]. Accurately dividing the boundaries into classes based on motion is a key step that affects the speed and accuracy of the calculation. The selection of moving boundaries is determined solely by the physical formulation of the problem [2–4]. However, in practice, attention should also be paid to boundaries that remain “fixed” throughout the calculation. Fixed boundaries include model surfaces (computational grids), for which invariance is guaranteed in both the numerical experiment and physical model. In addition, often the open boundaries of the computational grid should remain stationary in order to simplify the problem definition.

In some cases, it may be necessary to define open boundaries (boundaries of the computational domain or subarea) as movable [5]. This is particularly practical when the computational model (grid) includes one or more boundaries that move as a unit without varying the distance between any two points on the surface being examined. With this configuration, the entire computational grid can be moved according to the set of moving boundaries without using deformation algorithms, resulting in a significant reduction in simulation time and required computational power.

However, this approach becomes inapplicable if at least one of the boundaries is fixed or moves independently. In cases where the relative motion of the boundaries is insignificant, the widely used method involves grid deformation with the preservation of the communication topology [6]. This approach is suitable for modeling the “direct” motion of an object. Unlike the conventional method of determining aerodynamic characteristics, where the flow runs over a stationary body, this method assumes that the body is moving in a stationary medium. Such problems arise during the design stage of aircraft when determining the aerodynamic characteristics (ADC) for complex flight modes. These problems are currently solved using finite volume methods with arbitrary unstructured grids [7]. Such grids are selected in view of the fact that unstructured grids are usually used in engineering practice at the present time because of their higher flexibility and a possibility to build more complex models [8]. This choice, in turn, requires adapted numerical schemes implemented in the code to ensure that the results obtained are correct. So, this fact gave an impetus to the research in this area that allowed improving various methods for computations on unstructured grids [9].

The methodology for modeling the interaction between a moving body and a viscous compressible gas is based on the Navier–Stokes simultaneous equations. However, the movement of nodes not only affects the inherent characteristics of the discrete model but also modifies the underlying equations. This paper presents an adaptation of classical equations for solving numerical aerodynamics problems on deformable grids while preserving the communication topology. The key component of the updated numerical scheme is the boundary velocity vector. Although different methods can be used for its reconstruction, for arbitrary deformations of the computational grid, they typically rely on the condition of geometric conservation [10]. However, to obtain accurate simulation results, it is necessary to analyze the deformation of the computational grid together with the condition of geometric conservation, as further demonstrated in this paper.

This paper discusses the adaptation of the Navier–Stokes equations for the numerical modeling of fluid flow around moving bodies on deformable grids while preserving the communication topology. The algorithm for reconstructing the boundary velocity vector is given special attention. The proposed formula, which is based on the deformation analysis of the discrete model and the condition of geometric conservatism, ensures accurate results for regular and unstructured grids.

The efficiency of the developed numerical algorithms is demonstrated on typical problems of fluid flow around bodies, such as the NACA0012 and AGARD 445.6 airfoils [11]. The numerical simulation of these problems is performed in the direct formulation, where the body is moving. The correctness of the developed methods is shown through the analysis of the lift and drag forces, which are accurately simulated by the proposed numerical schemes.

2. Basic Equations, Method of Discretization, and Boundary Conditions

The unsteady three-dimensional turbulent flows of viscous heat-conducting gas are described by the Reynolds-averaged Navier–Stokes equations [12]. These equations, in their conservative form and expressed in Cartesian coordinates, are given below (averaging signs are omitted):

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \\ \frac{\partial (\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \nabla \cdot (\tau_\mu + \tau_t) \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \vec{u} h) = \nabla \cdot [\vec{u} \cdot \tau_v - (\vec{q}_\mu + \vec{q}_t)] \end{cases}, \quad (1)$$

The simultaneous equations (1) use generally accepted designations: ρ —density; $\vec{u}$ —flow velocity vector with components u , v , and w ; p —pressure; $E = C_v T + 0.5(u^2 + v^2 + w^2)$ —total energy of gas; $h = C_p T + 0.5(u^2 + v^2 + w^2)$ —total enthalpy of gas; τ_μ and τ_t —molecular and turbulent components of the tangential tensor of shearing strain, respectively; q_μ and

q_t —molecular and turbulent components of heat flow density, respectively; T —temperature; $C_v = (C_p T - R/m)$ —specific heat capacity at constant volume; C_p —specific heat capacity at constant pressure; R —universal gas constant; m —molar mass of gas; and τ_v —viscous stress tensor.

The molecular component of the tangential strain tensor of a Newtonian fluid obeys Newton's law of viscosity, which relates the viscous strain tensor to the strain rate tensor. The components of the heat flow density vector are related to the local temperature gradient through Fourier's law [10].

$$\tau_{\mu} = 2\mu(T) \left(S - \frac{1}{3} I \nabla \vec{u} \right), \quad S = \frac{1}{2} \left(\nabla \vec{u} + [\nabla \vec{u}]^t \right), \quad q_{\mu} = \lambda(T) \nabla T, \quad (2)$$

The dynamic viscosity coefficient $\mu(T)$ and heat transfer coefficient $\lambda(T)$, which vary with the flow temperature, are determined using the Sutherland formula [13]:

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{0.5} \frac{T_0 + T_s}{T + T_s}, \quad \chi = \chi_0 \left(\frac{T}{T_0} \right)^{0.5} \frac{T_0 + T_s}{T + T_s}, \quad (3)$$

where μ_0 and χ_0 —dynamic viscosity and heat transfer coefficient, respectively, at temperature T_0 ; and T_s —Sutherland coefficient.

The simultaneous equations (1) are incomplete due to the unknown relationship between variables τ_t and q_t of the system and the averaged parameters of the flow. This relationship is established using additional relations, known as turbulence models. In engineering, linear differential turbulence models [14,15] are commonly used, which utilize empirical relations for the turbulent viscosity coefficient μ_t , as well as eddy-resolving models such as Large Eddy Simulation (LES) and Detached Eddy Simulation (DES) [16,17].

To solve this system, boundary conditions of various types should be added to the Navier–Stokes equations. A description of these conditions can be found in works [12,13].

The control volume method is used to numerically solve the simultaneous equations (1). Arbitrary polyhedrons that cover the computational domain without gaps or overlaps are used as control volumes (grid cells). Each polyhedron is bounded by an arbitrary number of faces with vertices corresponding to the grid nodes. Figure 1 depicts the general configuration of a cell.

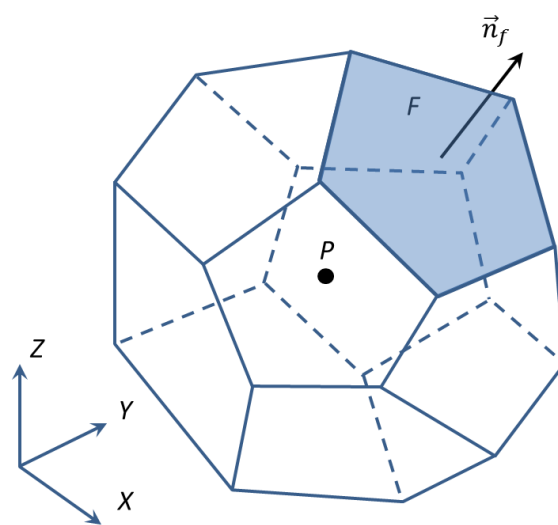


Figure 1. General view of a grid cell.

Let us clarify Figure 1. Here, P —center of a cell; F —one of the cells faces highlighted in color; and $\vec{n}_f$ —normal to the face F .

To apply the finite volume method for numerical solution, the Navier–Stokes equations for the three-dimensional flow of a heat-conducting viscous gas can be expressed in vector form:

$$\frac{d}{dt} \int_{\Delta V} W dV + \oint_{\Delta \Sigma_P} (F - G) dS = \int_{\Delta V} H dV, \quad (4)$$

where vector W —vector of conservative variables; F and G —vectors of convective and diffusive flows; and H —source term.

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad F = \begin{pmatrix} \rho u_n \\ \rho u u_n + p n_x \\ \rho v u_n + p n_y \\ \rho w u_n + p n_z \\ \rho h u_n + p u_n \end{pmatrix}, \quad G = \begin{pmatrix} 0 \\ \tau_{nx} \\ \tau_{ny} \\ \tau_{nz} \\ \left(\vec{\tau}_n, \vec{u} \right) + q \end{pmatrix} \quad (5)$$

where u_n —normal component of velocity; q —heat flow; and τ_{nj} —normal components of the viscous stress tensor, $\vec{\tau}_n = (\tau_{nx}, \tau_{ny}, \tau_{nz})$.

The mean value theorem is used for the integration. The value in the center of the cell is taken as the mean value over the volume, whereas the value in the center of the face is taken as a mean value over the face. Hence, Equation (4) can be expressed as follows:

$$\left(\frac{\partial W}{\partial t} \right)_P \Delta V_P + \sum_{f=1}^n [F(W) - G(W, \nabla W)]_f \Delta S_f - (H(W, \nabla W))_P \Delta V_P = 0, \quad (6)$$

or in the operator form

$$\left(\frac{\partial W}{\partial t} \right)_P \Delta V_P + L(W) = 0, \quad (7)$$

where

$$L(W) = \sum_{f=1}^n [F(W) - G(W, \nabla W)]_f \Delta S_f - (H(W, \nabla W))_P \Delta V_P, \quad (8)$$

The difference operator $L(W)$ includes approximations of convective and diffusive flows, as well as sources, and is formed on the basis of the implicit approximations of these flows. One-sided differences are typically used to approximate convective flows. Therefore, the approximation of a differential operator by a difference operator is of first order with respect to the spatial variable. As the equations are nonlinear, iterations over nonlinearity can be used to determine the solution at a new time layer. Let us use γ index to denote the parameters found at the previous iteration. Consequently, the value of operator (8) can be represented in the following form:

$$L(W^{\gamma+1}) = L(W^{\gamma+1}) - L(W^\gamma) + L(W^\gamma) = \Delta L_1(W^{\gamma+1}, W^\gamma) + L_2(W^\gamma), \quad (9)$$

where

$$L_2(W^\gamma) = \sum_{f=1}^n [F(W^\gamma) - G(W^\gamma, \nabla W^\gamma)]_f \Delta S_f - (H(W^\gamma, \nabla W^\gamma))_P \Delta V_P, \quad (10)$$

$$\Delta L_1(W^{\gamma+1}, W^\gamma) = \sum_{f=1}^n \left[\frac{\partial F}{\partial W} - \frac{\partial G}{\partial W} \right]_f^\gamma (W^{\gamma+1} - W^\gamma)_f \Delta S_f - \frac{\partial H}{\partial W}_P^\gamma (W^{\gamma+1} - W^\gamma)_P, \quad (11)$$

The resulting delta form of the difference equations is as follows

$$\frac{\Delta V_P}{\Delta t} \Delta W_P + \sum_{f=1}^n \left[\frac{\partial F}{\partial W} - \frac{\partial G}{\partial W} \right]_f^\gamma \Delta W_f \Delta S_f - \frac{\partial H}{\partial W}_P \Delta W_P \Delta V_P = R(W), \quad (12)$$

The right-hand side of the equations represents the residual of the balance equations

$$R(W) = -\frac{W_p^\gamma - W_p^n}{\Delta t} \Delta V_p - \sum_{f=1}^n [F - G]_f^\gamma \Delta S_f + H_p^\gamma \Delta V_p, \quad (13)$$

and the widely used Godunov-type schemes for computing convective flows are the Roe [18] and AUSMPW [19] schemes that have a clear physical interpretation. In these schemes, the solution is first reconstructed, which results in the calculation of the primary flow parameters to the left and right of the cell face. To calculate the diffusion flows using the explicit scheme, the parameters and derivatives defined on the inner faces are obtained through interpolation from their values in the cells.

To solve practical problems involving movable boundaries, the simultaneous equations (6) must be supplemented with boundary conditions that account for the motion of the boundary faces in the discrete model, in addition to those generally accepted [12].

The boundary condition for a “solid” wall undergoes a crucial modification to account for boundary mobility. When the boundary faces are fixed, implementing the “solid” wall boundary condition practically involves introducing a “fictitious” velocity (in the fictitious cell) that is related to the velocity vector in the real cell by a given equation [12,13]:

$$\vec{V}' = \vec{V} - 2\vec{n} \left(\vec{n}, \vec{V} \right), \quad (14)$$

Geometrically, Equation (14) creates a mirrored copy of the velocity vector $\vec{V}$ with respect to the plane of the face, which results in zero convective flows (Figure 2).

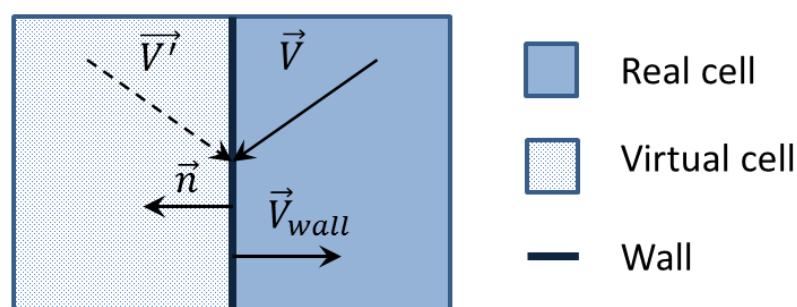


Figure 2. Processing of boundary condition “wall”.

The situation becomes more complicated when the face in the boundary condition “solid wall” is mobile. In such cases, it is necessary to consider not only impermeability but also the contribution to the momentum of the cell due to its motion, in accordance with Newton’s third law. In this scenario, the expression for velocity in the fictitious cell $\vec{V}'$ will take the following form:

$$\vec{V}' = \vec{V} - 2\vec{n} \left(\vec{n}, \vec{V} \right) + 2\vec{V}_{wall}, \quad (15)$$

where $\vec{V}_{wall}$ —velocity of the face for the boundary condition “solid wall.”

Equations (14) and (15) are applicable to walls with both adhesion and sliding.

Boundary division into subsets based on deformation is typically determined by the problem formulation. Special attention should be given to both movable and fixed boundaries since they influence the deformation process of the computational grid. Although open boundaries are often fixed to simplify problem formulation, this is only a recommendation. It is important to note that there may be multiple correct methods to subdivide grid boundaries into subsets for a given physical process. Ultimately, the decision to subdivide boundaries into classes is left to the discretion of the researcher.

In addition to stationary and moving boundaries in a discrete model, some boundaries can or should be modified during computation without affecting the simulation result or the physics of the process. The nature of the modification of the shape and position of such boundaries is usually undetermined in advance. Such boundaries are referred to as free boundaries, which adjust to variations in the computational grid. Free boundaries can be used to simplify problem formulation, as well as open boundaries for problems with the predetermined motion of one or a set of boundaries, as discussed earlier. Free boundaries should be used when the moving surface is coupled to a boundary such as a symmetry plane. The fixation of the symmetry plane when moving with the coupled surface will inevitably lead to the degradation in the quality of the computational grid due to the extensive formation of reversed, overlapped cells in the contact area of the surfaces.

It is important to note that a surface such as the symmetry plane must accurately reflect the physical meaning of such a boundary condition, which is only possible when the surface is planar. Therefore, such boundary conditions must be defined as free boundaries with a shape restriction, which prohibits the normal component of the increment in the coordinates of the surface nodes. In other words, the nodes belonging to the symmetry plane can only move within that plane.

3. Discretization of Basic Equations with Moving Boundaries While Preserving Coupling Topology of a Computational Domain

The discretization method described in the previous paragraph is only valid if the computational grid remains stationary. To solve aerodynamic problems that involve moving structural elements, it is necessary to employ specific algorithms in addition to the presented method. These algorithms are based on deformable grid approaches that preserve communication topology and use overlapping grids technology to handle solid boundaries belonging to different regions that intersect or contact. The implementation of these methods should meet the quality requirements of the deformed computational grid, as well as ensure the convergence and stability of the numerical algorithm when using interpolation methods for overlapping grid problems.

In this document, the terms ‘motion’ and ‘deformation’ refer to changes in the coordinates of one or more surfaces during the calculation. These changes can include the translational motion or rotation of the entire computational grid, as well as motion or alteration of individual boundary conditions resulting in compression, stretching, bending, or other forms of deformation. Whenever there is potential ambiguity in the use of these terms, additional clarifications will be provided.

The discretization of the simultaneous equations (1) is applicable only for stationary computational grids. To solve aerodynamic problems involving moving structural elements, particularly when the boundaries of the control volume are in motion, it is recommended that the presented method be supplemented by specific algorithms.

The variations in the control volumes should be taken into account when constructing numerical calculation schemes to accommodate the motion of the boundaries. The basic law can be modified using the differential identity [1]:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} W dV = \int_{\Omega(t)} \frac{\partial W}{\partial t} dV + \int_{\Omega(t)} W \left(\vec{\dot{x}} \cdot \vec{n} \right) dS, \quad (16)$$

where $\vec{\dot{x}}$ —velocity of the face.

The left-hand side of Equation (16) indicates that the control volume can vary with time, as shown by the second term on the right-hand side. When the boundaries move, the nodes of the computational grid move, resulting in a transformation of the control volume. By substituting Equation (16) into (4), the discrete equations take the following form:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} W dV + \int_{\partial\Omega(t)} \left(F(W) - \vec{\dot{x}} W \right) \cdot \vec{n} dS - \int_{\partial\Omega(t)} (G(W)) \cdot \vec{n} dS = \int_{\Omega(t)} H dV, \quad (17)$$

The first term in the equation above represents both the time variation in conservative variables and the rate of variation in the control volume. In the case of convective flows, the face velocity must also be considered. If all boundaries are stationary, the basic Navier–Stokes equation can be obtained from Equation (17).

However, to ensure the accuracy of numerical schemes in calculating Equation (17), further restrictions are imposed on the face velocity. In the simulations of flows on moving grids, errors in discretizing time derivatives and flows that depend on the velocity of cell faces can lead to a reduction in the order of accuracy of the difference scheme.

The paper [1] showed that in order to achieve at least first-order accuracy in time, it is necessary to satisfy the Geometric Conservation Law (GCL) condition. This condition was extensively discussed in works such as [10,20,21]. The GCL requires that the discretization scheme satisfy geometric conservation when the vector $W = \text{const}$ is an exact solution for equations like Equation (17).

The velocity of the boundaries of the control volume is a key factor in constructing a discretization scheme for equations such as Equation (17). However, accurately determining this velocity can be challenging. One approach is to use GCL condition [18,19].

To simplify the forthcoming calculations, let us consider Equation (17) without including the sources on the right-hand side. Let us analyze how the expression will change if it is assumed that the computed fields are constant at every point in space, or $W = \text{const}$. By substituting $W = \text{const}$ into Equation (17), we obtain the following expression:

$$W \frac{dV}{dt} + \int_{\partial\Omega(t)} \left(F(W) - \vec{x} W \right) dS = 0 \quad (18)$$

In Equation (18), no diffusion term is present since no viscous (shear) stresses are observed in an unperturbed medium, and, therefore, it can be assumed that $G(W) \equiv 0$.

On the other hand, the condition $W = \text{const}$ indicates that the total mass, momentum, and energy flows through the faces of the control volume are equal to zero. Therefore, Equation (18) can be simplified as follows:

$$\frac{dV}{dt} - R^* \left(x(t), \vec{x}(t) \right) = 0, \quad (19)$$

or in a discrete form by applying the Euler scheme:

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \sum_f \vec{x}_f \Delta S_f, \quad (20)$$

From Equation (20), it can be concluded that the variation in the volume is the sum of the “swept” volumes of each face, which can be expressed as:

$$V_i^{n+1} - V_i^n = \sum_f \Delta V_f, \quad (21)$$

On the other hand, the “swept” volume can be expressed as:

$$\Delta V_f = \int_{t_n}^{t_{n+1}} \int_f \vec{x} \vec{n} dS dt, \quad (22)$$

Given Equation (20), the limitation imposed on the face velocity can be written as:

$$\vec{x} \vec{n} = \frac{\Delta V_f}{\Delta t \Delta S_f}, \quad (23)$$

It is important to note that Equation (23) allows only the amplitude of the face velocity to be calculated, rather than its direction, as it is impossible to accurately reconstruct one of the vectors from the scalar product. The accurate estimation of the direction of the face velocity vector will be discussed later. Equation (23) expresses the velocity magnitude of a given face as the volume swept by it over time $\Delta t = t_{n+1} - t_n$.

The major challenge in using Equation (23) is determining the swept volume of a moving face, which can undergo arbitrary deformations. Here, “deformation” stands for the change in the area and shape of the face, rather than in the edge–node topology. This is due to the nodes of the face moving in arbitrary ways, making it difficult to apply standard methods for calculating the volume of the prism. However, the swept volume can be determined using the method described below [21].

Let an arbitrary cell of the control volume (Figure 3) vary over time.

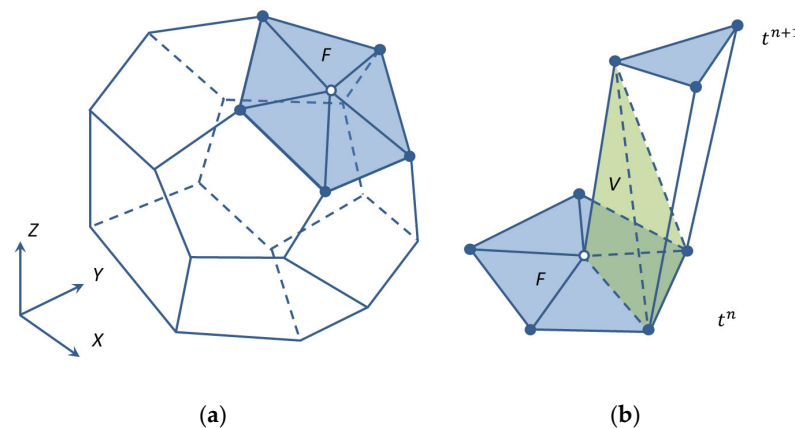


Figure 3. Face motion. (a) General view of cell and face and (b) face splitting and deformation.

Let us attempt to find the swept volume of the face selected in Figure 3 using the algorithm based on the approach of average normal reconstruction, which involves the following steps:

- Calculating the new node position and face center;
- Splitting the face into triangles with a common vertex at the center of the initial face;
- Calculating and summing the volume of each triangular prism.

Subsequently, let us illustrate the variations in the position of a single prism (Figure 3b) over time (Figure 4).

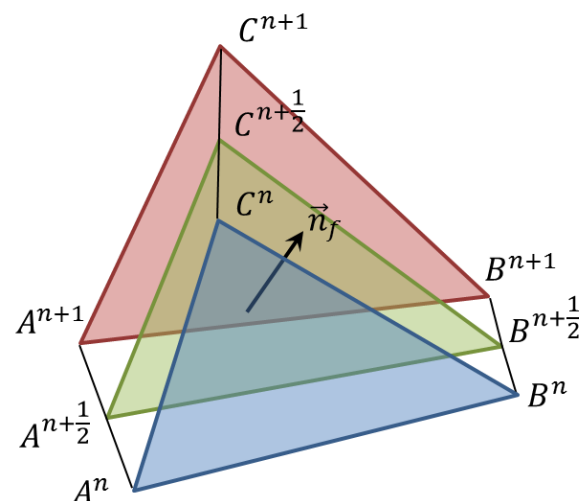


Figure 4. Fixed prism.

Here, the time $t_{n+\frac{1}{2}}$ corresponds to the intermediate position of the face, based on which the swept volume is calculated.

Let us introduce the following notations:

$$\begin{aligned} AC &= A - C; \quad BC = B - C \\ \Delta A &= A^{n+1} - A^n; \quad \Delta B = B^{n+1} - B^n; \quad \Delta C = C^{n+1} - C^n; \end{aligned} \quad (24)$$

Then, the swept volume can be expressed as follows:

$$V_f = \frac{1}{3}(\Delta A + \Delta B + \Delta C) \cdot \eta, \quad (25)$$

where

$$\eta = \frac{1}{6} \left([AC^n, BC^n] + [AC^{n+1}, BC^{n+1}] + \frac{1}{2} \left([AC^n, BC^{n+1}] + [AC^{n+1}, BC^n] \right) \right), \quad (26)$$

By using Equations (24)–(26), it is possible to calculate the numerical values of the velocity for each face in the control volume. This enables numerical modeling of hydro-aerodynamics problems involving moving boundaries.

The movement of boundaries necessitates the development of nonstationary computational schemes for equations such as Equation (17). A commonly used approach assumes that the flow is stationary at each moment [22]. As a result, instead of solving equations such as Equation (17), modified equations are considered:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} W dV + \frac{\partial}{\partial \tau} \int_{\Omega(t)} W dV + \oint_{\partial\Omega(t)} (F - \dot{x}W - G) dS = \int_{\Omega(t)} H dV, \quad (27)$$

where τ —pseudo time.

In Equation (27), the first two terms can be formally discretized. The first-order accurate scheme for time can be expressed as follows:

$$\begin{aligned} \left[\frac{V^{n+1}}{\Delta \tau} + \frac{2V^{n+1} - V^n}{\Delta t} \right] \Delta W^{k+1} + \oint_{\partial\Omega(t)} [F - \dot{x}W - G] dS = \\ = \int_{\Omega(t)} H dV - \frac{1}{\Delta t} \left(W^k (2V^{n+1} - V^n) - W^n V^{n+1} \right), \end{aligned} \quad (28)$$

where V^{n+1} , V^n —cell volume at times t_{n+1} , t_n , respectively.

As mentioned earlier, Equation (23) allows only the amplitude of the face velocity to be calculated, whereas its direction remains uncertain. The incorrect determination of the face velocity vector can result in unphysical values due to the inaccurate computation of momentum contributions [1]. Due to the unique aspects of computations on unstructured grids, such inconsistencies can significantly degrade the solution and stability of the numerical method.

The direction of face motion can be determined by introducing additional characteristics of the face, such as its previous state, which can help to monitor the movement of the entire face system.

Determining the velocity vector of a face becomes more complicated if its motion involves processes like rotation or compression/tension, especially on unstructured grids. If the face motion is close to plane-parallel shear, the velocity vector can be easily determined. However, due to the use of arbitrary polygons as faces in unstructured grids, ensuring their planarity is generally impossible. To calculate the swept volume, the faces are triangulated as described above, and the value is subsequently calculated using Equation (26). Additionally, it should be noted that grid deformation increases the probability of grid model degradation, which further complicates the determination of the face velocity vector. The

following experimental algorithm is proposed to determine the velocity vector of a face for an arbitrary unstructured grid:

- Calculating the swept volume, node displacements, and displacements of the face center based on the algorithm (24)–(26);
- Averaging the calculated displacements and normalize the averaged displacement vector;
- Scaling the normalized averaged displacement vector using Equation (23).

The introduction of an averaging step for the calculated movements of the face nodes facilitates determining the key information about the direction of the face velocity for arbitrary unstructured grids. This mechanism, along with the scaling defined by Equation (23), helps to preserve the validity of the numerical scheme, crucial for the accuracy and stability of the solution.

As a result, the modified formula for determining the velocity vector takes the form:

$$\vec{\dot{x}} = \frac{\Delta V_f}{\Delta t \Delta S_f} \cdot \frac{\sum_{k=1}^k \vec{dR}_k + \vec{dR}_c}{\left\| \sum_{k=1}^k \vec{dR}_k + \vec{dR}_c \right\|}, \quad (29)$$

where $\vec{dR}_c$, $\vec{dR}_k$ —displacement of the face center and nodes.

Equation (29) defines a vector with a magnitude of $\frac{\Delta V_f}{\Delta t \Delta S_f}$, which satisfies the condition of geometric conservatism and ensures the appropriate field distribution by accurately determining the direction of the examined vector.

The above equations are applicable for any type of boundary conditions where faces are movable. It is worth noting that these assumptions have no effect on the number of geometric entities in the discrete model or the ratio of cells or nodes through a face, i.e., the topology of the grid, enabling efficient solutions of aerodynamics problems with movable boundaries while using computational power and memory. The algorithm can be illustrated using a flow chart shown in Figure 5.

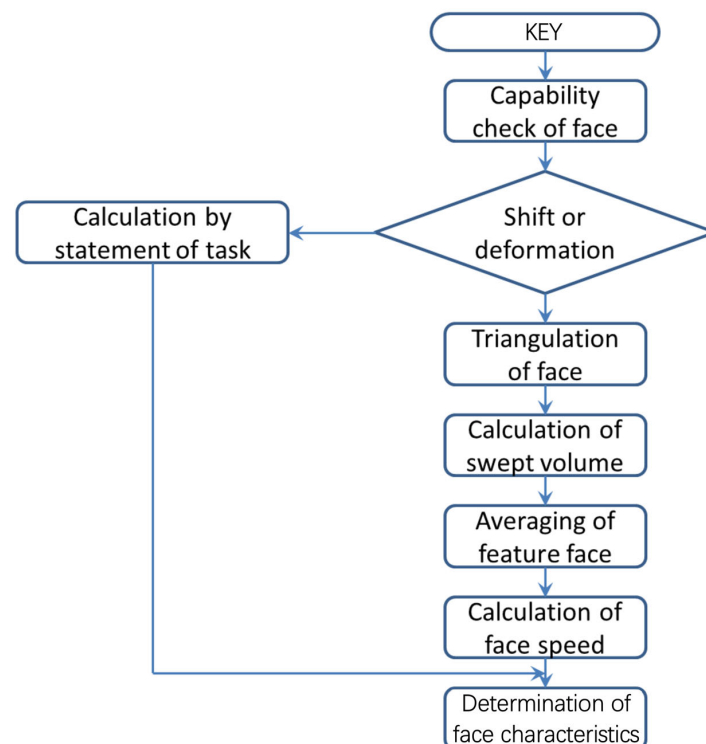


Figure 5. Extended flow chart for calculating face velocity.

As shown in Figure 5, the calculation of the face velocity commences with the “Determination of face characteristics” block. This block performs the initial processing of information about the face based on the problem statement. When the entire computational domain is shifted, determining the face velocity is a typical problem. In the case of deformation, the volume of the observed face is calculated and used to scale the averaged direction of the face motion.

The discretization methods discussed above enable calculations with moving boundaries, independent of the grid deformation algorithms. However, it is important to note that the accuracy of the obtained results depends directly on the quality of the deformed grid. Hence, another critical aspect of the methodology for solving problems involving time-varying boundaries is the grid deformation algorithm.

4. Numerical Experiments

The methods and algorithms described above are implemented in the Russian computer-aided engineering (CAE) software package LOGOS version number 5.3.21, which is designed to solve a wide range of industrial problems [23–25]. All calculations presented in this article were performed using the LOGOS software package.

4.1. Direct Flow around NACA0012 Airfoil Profile

The numerical experiment in this study examines the problem of unsteady flow around the NACA0012 airfoil profile using a viscous compressible gas flow. The analytical relation defines the shape of the airfoil profile:

$$\frac{y}{c} = \pm \frac{0.21}{0.2} \left(0.2969 \sqrt{\frac{x}{c}} - 0.1260 \frac{x}{c} - 0.3516 \left(\frac{x}{c} \right)^2 + 0.2854 \left(\frac{x}{c} \right)^3 - 0.1015 \left(\frac{x}{c} \right)^4 \right), \quad (30)$$

where $y(x)$ —profile surface coordinate; and c —chord airfoil length.

The problem is solved in two stages. In the first stage, the ADC in the stationary formulation is determined to obtain appropriate quantitative characteristics, which will be used for comparison with various algorithms for calculating the geometric characteristics of the “solid wall” boundary condition in the direct non-stationary calculation (where the body moves in the medium).

In the second stage, a direct non-stationary calculation is carried out using two approaches to determine the face velocity:

- Approach 1, which assumes collinearity of the face velocity vectors and the normal;
- Approach 2, which involves reconstructing the face direction.

The purpose of this numerical experiment is to demonstrate the significant deviation of ADC from the stationary solution, even in the first few time steps, when an incorrect algorithm is used to calculate the face velocity.

A discrete model containing approximately 160,000 cells was used for the calculations (Figure 6).

In order to solve the steady-state problem, the flow of a viscous compressible gas with the following parameters was considered: $M_\infty = 0.184893$ —Mach number; $\alpha = 4^\circ$ —incidence angle; $T_\infty = 245.35$; K—total temperature; $P_\infty = 95,610$ Pa—static pressure; and $V_\infty = 58.1$ m/s—inlet velocity.

The results of the steady-state calculation are as follows. Figure 7 illustrates the distributions of turbulent viscosity and velocity amplitude of the convergent solution of the stationary problem, respectively.

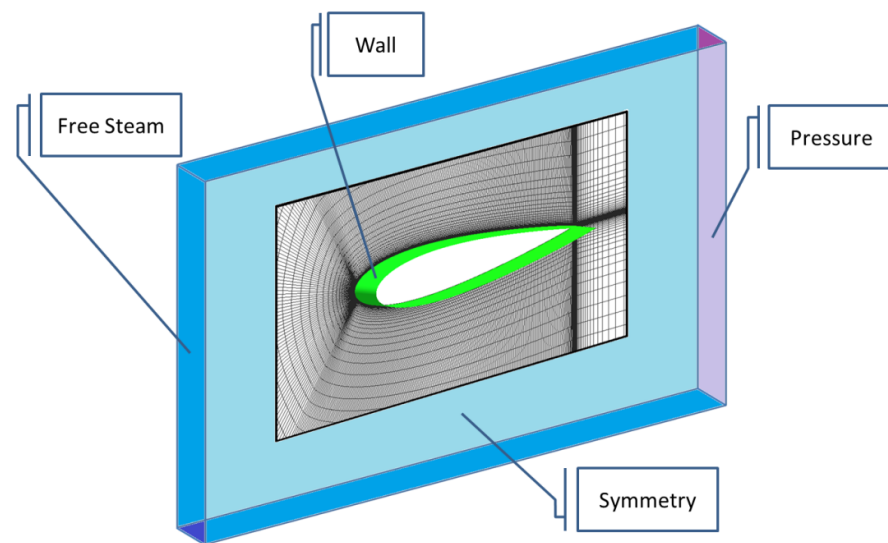


Figure 6. Schematic of a calculation model for NACA0012.

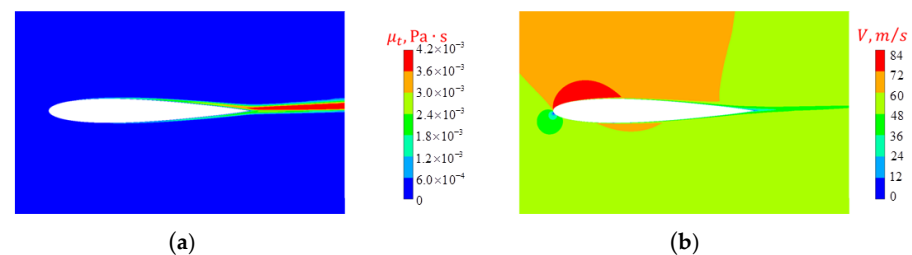


Figure 7. Stationary solution. (a) Turbulent viscosity and (b) velocity amplitude (V).

The presented figures illustrate the turbulent wake formation and the characteristic distribution of the velocity field for this flow regime. In the steady-state calculation, the solid wall was kept fixed throughout the simulation.

In order to ensure the resumption (restart) of the iterative process in the stationary calculation corresponding to the dual process of airfoil motion in an unperturbed medium, the velocity field and open boundary conditions were modified. At each point in space, the resulting velocity field was calculated using the following formula:

$$\vec{V}_{new} = \vec{V}_{old} - \vec{V}_{far_field}, \quad (31)$$

where $\vec{V}_{new}$ —new velocity vector for direct calculation of airfoil motion; $\vec{V}_{old}$ —velocity vector at the position obtained in the stationary calculation; and $\vec{V}_{far_field}$ —velocity vector at the open boundary.

In addition to varying the velocity field, it is necessary to modify the parameters of the open boundary conditions by setting the velocity of the incoming flow to 0 m/s.

In order to ensure consistent calculation results, the grid deformation approach with preserved communication topology was used. Accordingly, all open boundaries were fixed, and a stationary velocity was imposed on the NACA0012 airfoil profile boundary. The symmetry plane neighboring the open boundaries and the moving wall was treated as a free boundary, and its nodes were moved using the inverse weighted interpolation approach [6].

Figure 8 shows the discrete model at the final moment of the simulation.

During the non-stationary calculation, the direct problem is solved, i.e., the ADC of the airfoil profile is calculated during its motion. In this case, the “solid wall” boundary

condition is characterized by the velocity, which is calculated using two algorithms (23) and (29) for each calculation case.

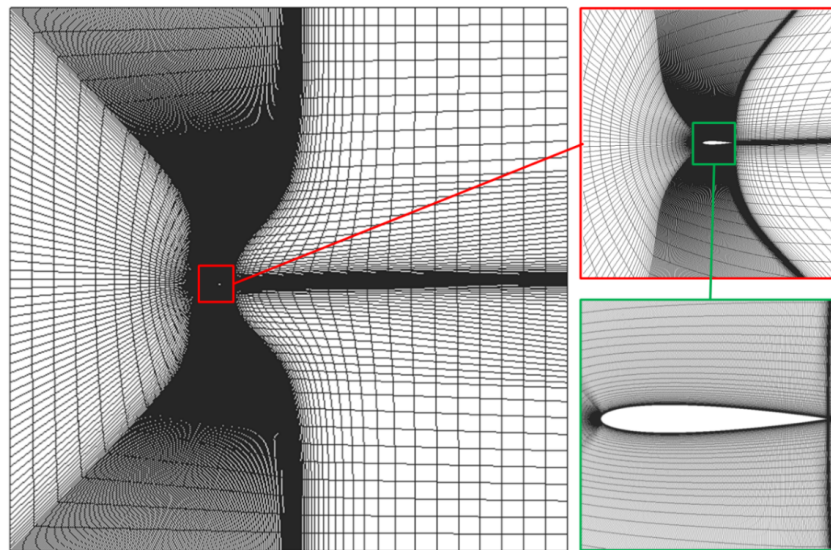


Figure 8. Calculation grid at the final time of calculation (profile travels from right to left).

Figures 9–12 show a comparison of turbulent viscosity distributions when using two different approaches to calculate the face velocity at different times.

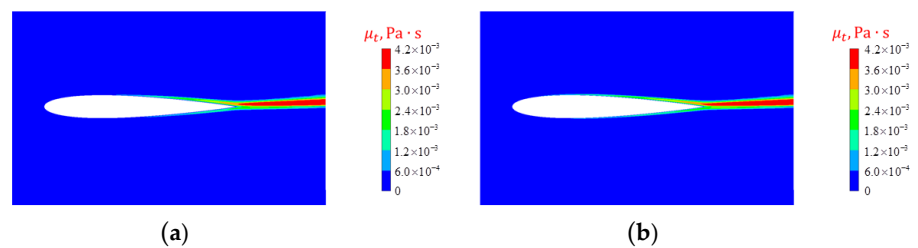


Figure 9. Turbulent viscosity at a time of 0.005 s. (a) Method 1 and (b) Method 2.

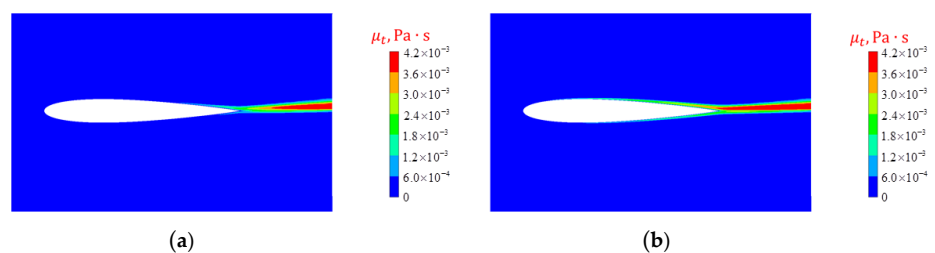


Figure 10. Turbulent viscosity at a time of 0.01 s. (a) Method 1 and (b) Method 2.

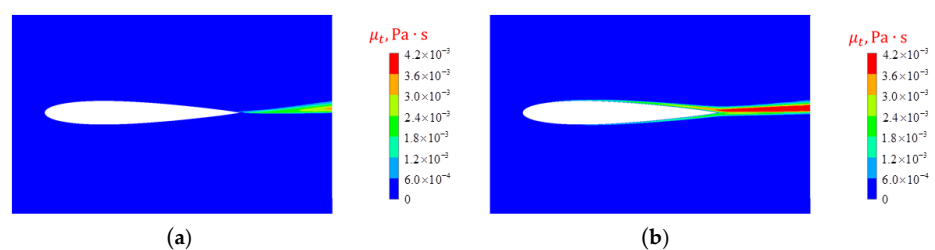


Figure 11. Turbulent viscosity at a time of 0.015 s. (a) Method 1 and (b) Method 2.

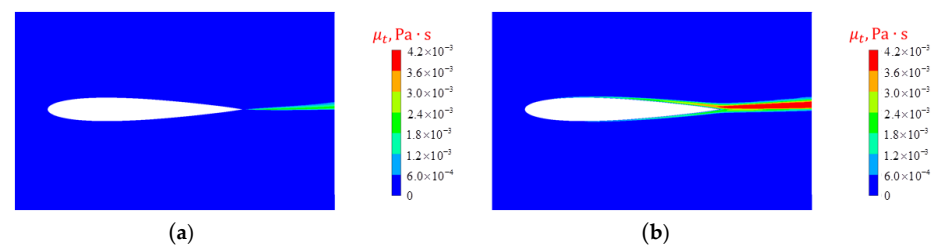


Figure 12. Turbulent viscosity at a time of 0.02 s. (a) Method 1 and (b) Method 2.

Figure analysis reveals that approach 1 (based on the collinearity assumption) leads to a detachment of the turbulent viscosity, whereas the modified approach 2 avoids such detachment. Using the modified approach to calculate face velocity results in a qualitatively and quantitatively accurate preservation of turbulent viscosity, correctly describing the underlying physics of the process. This is evidenced by comparing velocity distribution in the direct formulation at the same time (Figure 13).

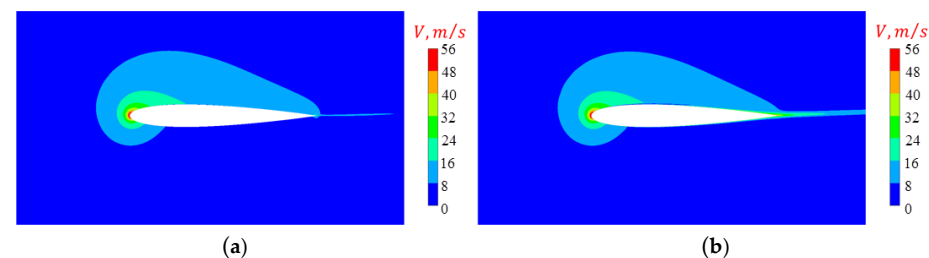


Figure 13. Velocity distribution at a time of 0.02 s. (a) Method 1 and (b) Method 2.

As shown in the figure, the modified approach preserves the qualitative distribution of the velocity field at the trailing edge of the wing.

On the other hand, the detachment of turbulent viscosity affects the ADC of the airfoil profile, specifically the drag and lift forces, as shown in Figure 14.

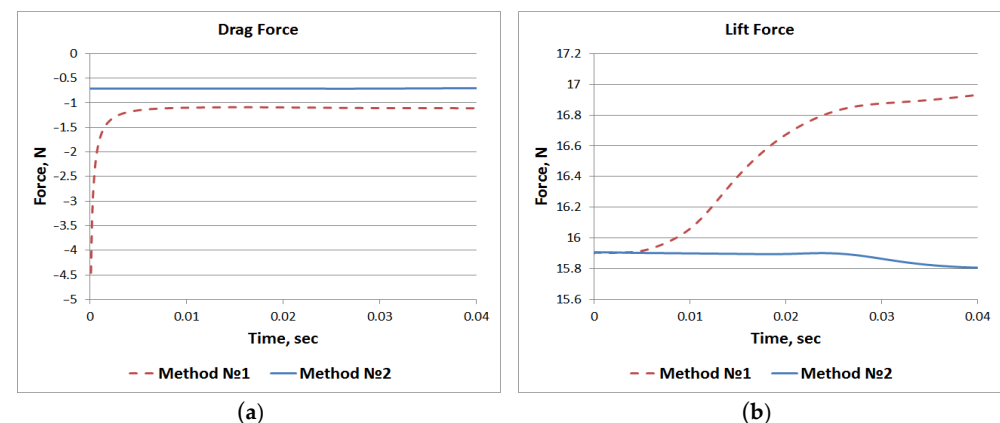


Figure 14. Comparison of aerodynamic loads for two approaches in time. (a) Drag force and (b) lift force.

It should be noted that, unlike the modified approach, the uncorrected calculation exhibits a discontinuity in drag force at the restart point (Figure 15).

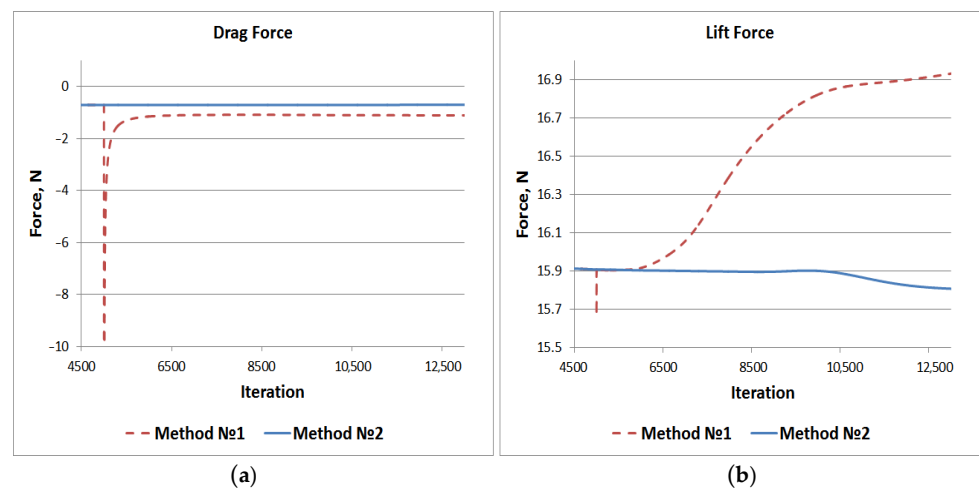


Figure 15. Comparison of aerodynamic loads for two approaches by iterations. (a) Drag force and (b) lift force.

The above graphs show the distribution of aerodynamic forces for two simulation stages:

- Stationary calculation of up to 5000 iterations, corresponding to the profile bleeding;
- Nonstationary calculation, corresponding to the direct problem.

It is evident from the graphs that both curves corresponding to Method 1 exhibit failure at the restart. However since the dual-step approach is employed to solve non-stationary problems, the numerical value of the lift force at the first step converges to a stationary solution, unlike the drag force, which remains incorrect throughout the calculation.

4.2. Direct Flow around AGARD 445.6 Airfoil Profile

The simulation of the AGARD 445.6 airfoil [11], whose geometry is shown in Figure 16, is considered.

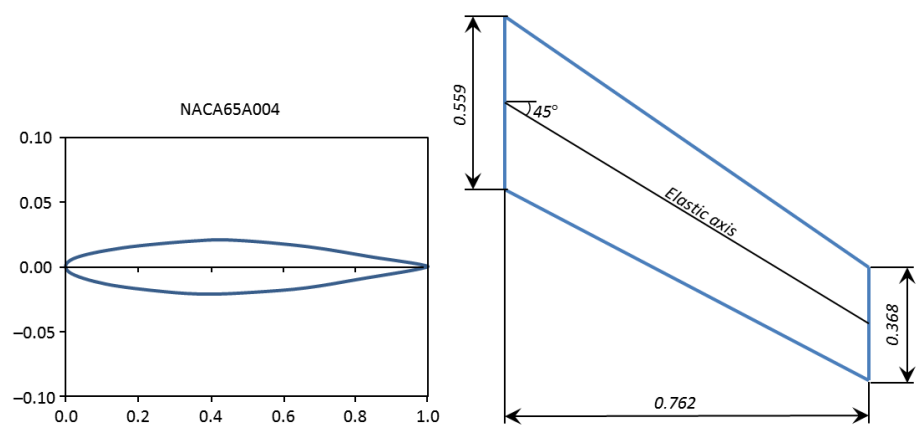


Figure 16. Airfoil geometry.

A discrete irregular 3D model containing approximately 2.3 million polyhedral cells was used for the calculations (Figure 17).

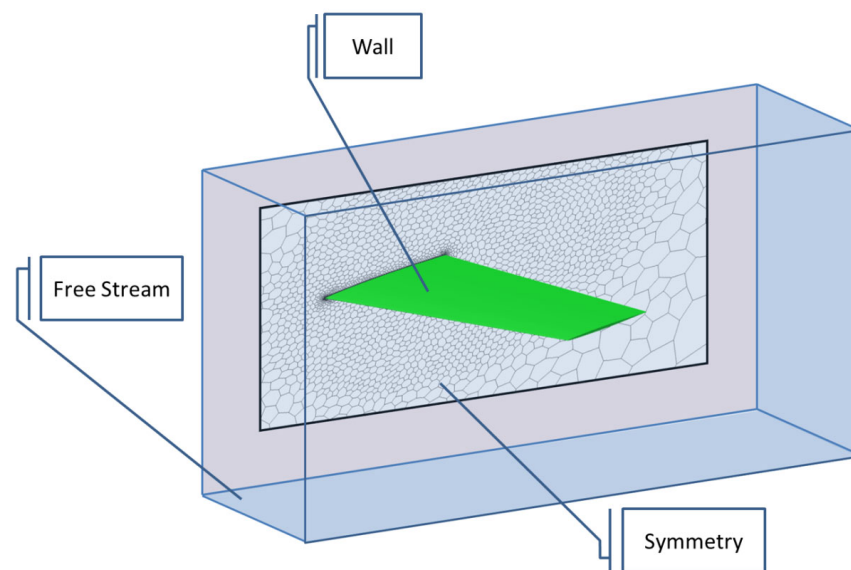


Figure 17. Schematic of a calculation model.

The open boundaries corresponding to the free flow were at a distance of 20 wing chords. The simulation was carried out in two stages, similar to the NACA0012 airfoil flow problem. In the first stage, the flow parameters were characterized by a Mach number of $M_\infty = 0.678$, corresponding $V_\infty = 227.54$ m/s, a total temperature of 280.255 K, the static pressure of $P_\infty = 20,781.6$ Pa, and an incidence angle of zero. The second stage of the simulation corresponded to a direct problem, where the initial field distributions are obtained by transforming the results of the first stage calculation. The time step was set to 0.0001 s. Figure 18 depicts a cross-section of the computational grid at the initial and final times.

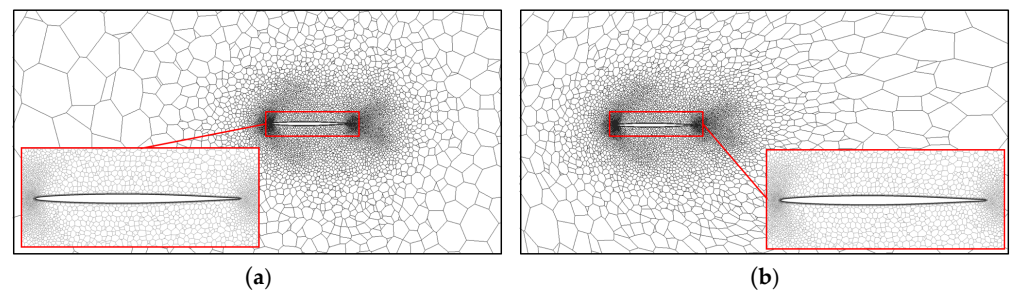


Figure 18. Cross-section of discrete model. (a) Initial time and (b) final time (0.004 s).

Figures 19 and 20 depict a comparison of the distributions of turbulent viscosity obtained using two different approaches for calculating velocity at the initial and final times.

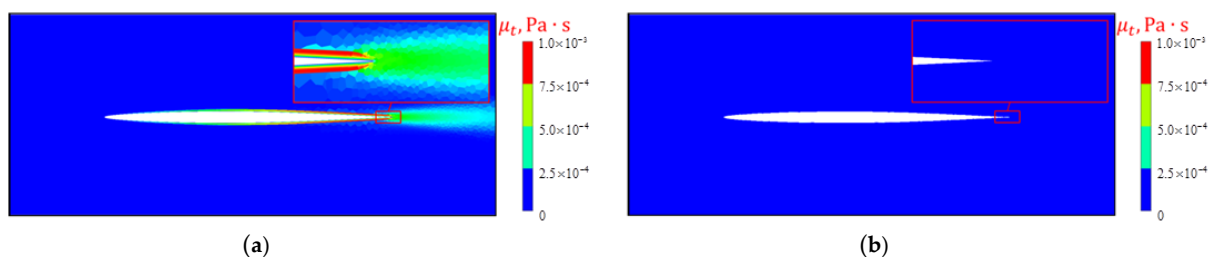


Figure 19. Turbulent viscosity, Method 1. (a) Initial time and (b) final time.

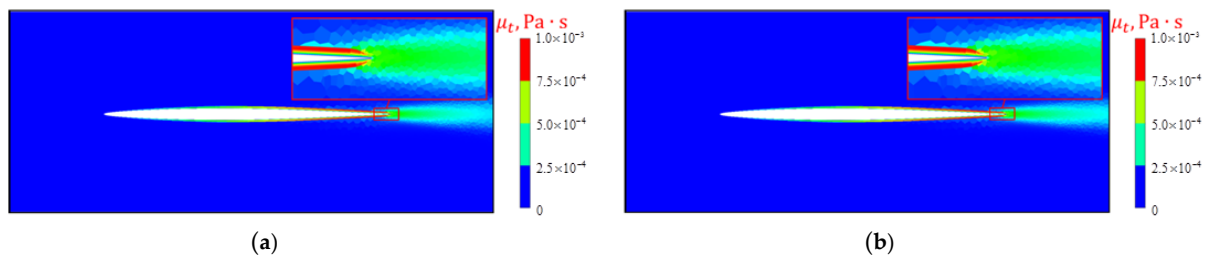


Figure 20. Turbulent viscosity, Method 2. (a) Initial time and (b) final time.

It can be observed from the above figures that using approach 2 preserves the qualitative distribution of turbulent viscosity, unlike Method 1. In order to compare the two approaches quantitatively, a comparison of aerodynamic loads is presented in Figure 21.

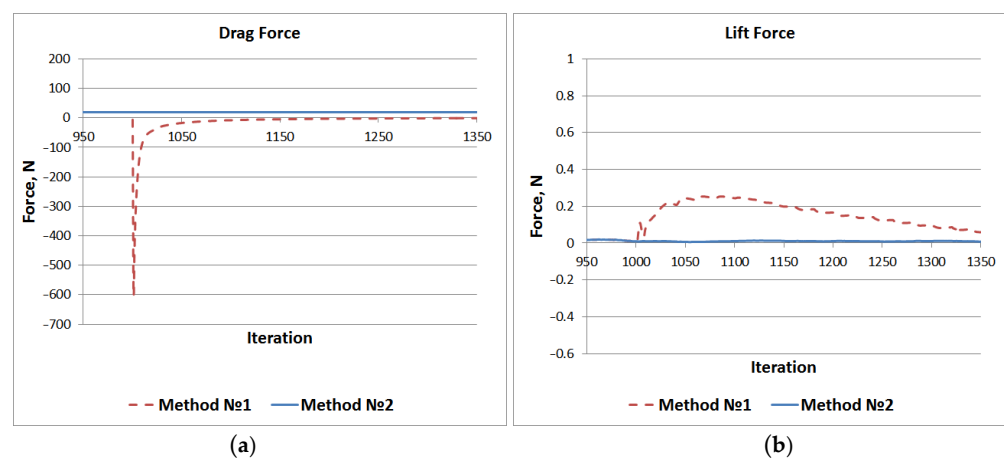


Figure 21. Comparison of aerodynamic loads for two approaches by iterations. (a) Drag force and (b) lift force.

Similar to the NACA0012 problem, a discontinuity in the solution is observed at the restart point of the calculation (1000 iterations).

This highlights the significance of accurately calculating the velocity vector of the face. Merely adhering to the condition of geometric conservatism fails to ensure precise results.

4.3. Analysis of the Results Obtained

The numerical simulation results for typical aerodynamics problems demonstrate a key difference of the considered methods used to calculate the velocity of an arbitrary moving face. Thus, the streamlined body ADCs obtained with Method 1 experience discontinuity at the restart point, whereas Method 2 allows maintaining the load continuity.

The ADC calculation for AGARD 455.6 wing differs from that for the airfoil in view of a significant 3D flow and the use of an unstructured computational grid. The combination of these factors complicates the given problem solution and, as a result, leads to a pronounced discontinuity of the ADC curves at the restart point and invalid results in further calculations.

An invalid solution is caused by the significant flow reconstruction near moving walls because of inconsistent vector directions of the face velocity and the velocity calculated from expression (23). The proposed approach to calculating the face velocity in the form of relation (29) allows preserving the boundary layer of the streamlined body generated in the steady computation stage. This advantage provides the ADC constancy for streamlined bodies and turbulent trace in the problem restart process, as shown in Figures 9b, 12b and 20.

This fact clearly demonstrates the importance of the correctly calculated face velocity vector because the requirement of the geometric conservativeness alone does not ensure the result validity.

5. Conclusions

This paper explores the unique aspects of modeling object motion within an arbitrary unstructured grid using the invariant principle of computational domain topology, with a focus on the finite-volume method. The paper describes the key elements of the calculation method for nonstationary problems with moving boundaries and investigates the correctness of a numerical scheme for moving grid boundaries. Based on an analysis of modified conservation laws, an improved scheme for calculating face velocity that ensures the correctness of the numerical scheme is proposed.

The presented approach for calculating the face velocity considers the direction of the face motion while adhering to the condition of geometric conservatism. This approach ensures the accurate calculation of the velocity vector of the face motion for various deformations of the computational grid, leading to high accuracy in simulating the flow around an object with movable boundaries. The proposed algorithm for reconstructing the velocity vector of the face velocity yields qualitative results on both structured and unstructured computational grids, providing a realistic representation of the physics of the flow process of viscous compressible gas around moving bodies.

The validity and efficiency of the proposed method were demonstrated on the problems involving the flow around NACA0012 and AGARD 445.6 airfoil profiles. The results emphasized the importance of correctly calculating the face velocity vector, particularly for three-dimensional irregular grids, as such grids are susceptible to strong solution dissipation.

Author Contributions: Conceptualization, A.K. and A.S.; methodology, A.K. and A.S.; software, A.K.; validation, R.Z. and D.S.; formal analysis, A.K. and D.S.; investigation, A.S.; writing—original draft preparation, A.S. and R.Z.; writing—review and editing, R.Z.; visualization, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work is prepared in the implementation of the program for the creation and development of the World-Class Research Center “Supersonic” for 2020–2025 funded by the Ministry of Science and Higher Education of the Russian Federation (Grant agreement of 20 April 2022 № 075-15-2022-309). The results have been obtained with financial support from the Science and Universities National Project under the Young Scientists Lab Program of the RF Ministry of Education and Science—project identifier No. FSWE-2021-0009 (Research Topic: Development of CFD methods, models and algorithms to simulate liquids and gases in natural and industrial environments under normal and critical conditions on petascale supercomputers) and the Council of the Grants of the President of the Russian Federation for state support of Leading Scientific Schools of the Russian Federation (Grant No. NSH-70.2022.1.5).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Volkov, K.N.; Emelyanov, V.N. *Flow and Heat Transfer in Channels and Rotating Cavities*; Publishing House of Physical and Mathematical Literature: Moscow, Russia, 2003.
2. Meakin, R.L.; Suhs, N.E. Unsteady aerodynamic simulation of multiple bodies in relative motion. In Proceedings of the 9th Computational Fluid Dynamics Conference, Buffalo, NY, USA, 13–15 June 1989; Volume 89, p. 1996-CP.
3. Rejniak, A.A.; Gatto, A. Influence of Rotating Wheels and Moving Ground Use on the Unsteady Wake of a Small-Scale Road Vehicle. *Flow Turbul. Combust.* **2021**, *106*, 109–137. [\[CrossRef\]](#)
4. Giannenas, A.E.; Bempedelis, N.; Schuch, F.N.; Laizet, S. A Cartesian Immersed Boundary Method Based on 1D Flow Reconstructions for High-Fidelity Simulations of Incompressible Turbulent Flows Around Moving Objects. *Flow Turbul. Combust.* **2022**, *109*, 931–959. [\[CrossRef\]](#)
5. Deryugin, Y.N.; Sarazov, A.V.; Zhuchkov, R.N. Features of overset meshes methodology on unstructured grids. *Math. Models Comput. Simul.* **2017**, *5*, 587–597. [\[CrossRef\]](#)
6. Luke, E.; Collins, E.; Blades, E. A fast mesh deformation method using explicit interpolation. *J. Comput. Phys.* **2012**, *231*, 586–601. [\[CrossRef\]](#)
7. Kozelkov, A.; Kurulin, V.; Emelyanov, V.; Tyatyushkina, E.; Volkov, K. Comparison of convective flux discretization schemes in detached-eddy simulation of turbulent flows on unstructured meshes. *J. Sci. Comput.* **2016**, *67*, 176–191. [\[CrossRef\]](#)

8. Mavriplis, D.J. Unstructured grid techniques. *Annu. Rev. Fluid. Mech.* **1997**, *29*, 473–514. [[CrossRef](#)]
9. Hansen, R.P.; Forsythe, J.R. A comparison of structured and unstructured grid solutions for flow over a circular cylinder. In Proceedings of the 2003 User Group Conference, Bellevue, WA, USA, 9–13 June 2003; pp. 104–112. [[CrossRef](#)]
10. Thomas, P.D.; Lombard, C.K. Geometric conservation law and its application to flow computations on moving grids. *AIAA J.* **1979**, *17*, 1030–1037. [[CrossRef](#)]
11. Yates, E.C., Jr. *AGARD Standard Aeroelastic Configurations for Dynamic Response I-Wing 445.6*; Agard Report № 765; Interdisciplinary Research Office NASA Langley Research Center: Hampton, VA, USA, 1988.
12. Fletcher, C.A.J. *Computational Techniques for Fluid Dynamics*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1991; Volume 1.
13. Loitsyansky, L.G. *Mechanics of Fluid and Gas*; Nauka: Moscow, Russia, 1973.
14. Spalart, P.R.; Allmaras, S.R. A one-equation turbulence model for aerodynamic flows. In Proceedings of the 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 6–9 January 1992; p. 0439.
15. Menter, F.R.; Kuntz, M.; Langtry, R. Ten years of industrial experience with SST turbulence model. *Turbul. Heat Mass* **2003**, *4*, 625–632.
16. Spalart, P.R.; Jou, W.H.; Strelets, M.; Allmaras, S.R. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In Proceedings of the First AFOSR International Conference on DND/LES, Ruston, LA, USA, 4–8 August 1997.
17. Travin, A.; Shur, M.; Strelets, M.; Spalart, P.R. Physical and numerical upgrades in the detached-eddy simulation of complex turbulent flows. *Fluid Mechanics and its Application*, 65. In *Advances in LES of Complex Flows, Proceedings of the Euromech Colloquium 412, Munich, Germany 4–6 October 2000*; Friedrich, R., Rodi, W., Eds.; Kluwer: Boston, MA, USA; London, UK, 2002; pp. 239–245.
18. Roe, P.L. Characteristic-based schemes for the Euler equations. *Annu. Rev. Fluid Mech.* **1986**, *18*, 337–365. [[CrossRef](#)]
19. Weiss, J.M.; Smith, W.A. Preconditioning applied to variable and constant density flows. *AIAA* **1995**, *33*, 2050–2057. [[CrossRef](#)]
20. Farhat, C.; Geuzaine, P.; Grandmont, C. The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids. *J. Comput. Phys.* **2001**, *174*, 669–694. [[CrossRef](#)]
21. Lesoinne, M.; Farhat, C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput. Methods Appl. Mech. Eng.* **1996**, *134*, 71–90. [[CrossRef](#)]
22. Blazek, J. *Computational Fluid Dynamics: Principles and Applications*; Elsevier: Amsterdam, The Netherlands, 2001; ISBN 008-0-4300-90.
23. Kozelkov, A.S.; Strelets, D.Y.; Sokuler, M.S.; Arifullin, R.H. Application of Mathematical Modeling to Study Near-Field Pressure Pulsations of a Near-Future Prototype Supersonic Business Aircraft. *J. Aerosp. Eng.* **2022**, *35*, 04021120. [[CrossRef](#)]
24. Kozelkov, A.S.; Struchkov, A.V.; Strelets, D.Y. Two Methods to Improve the Efficiency of Supersonic Flow Simulation on Unstructured Grids. *Fluids* **2022**, *7*, 136. [[CrossRef](#)]
25. Struchkov, A.V.; Kozelkov, A.S.; Volkov, K.; Kurkin, A.A.; Zhuchkov, R.N.; Sarazov, A.V. Numerical simulation of aerodynamic problems based on adaptive mesh refinement method. *Acta Astronaut.* **2020**, *172*, 7–15. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.