14th World Congress in Computational Mechanics (WCCM)
ECCOMAS Congress 2020
Virtual Congress: 11 – 15 January 2021
F. Chinesta, R. Abgrall, O. Allix and M. Kaliske (Eds)

UNSTRUCTURED MESH ADAPTATION FOR MOVING BODIES IN IMMERESED BOUNDARY METHODS

TATIANA KOZUBSKAYA[1], LIUDMILA KUDRYAVTSEVA[1,2] AND VALERIIA TSVETKOVA[1]

[1] Keldysh Institute of Applied Mathematics, 4, Miusskaya Sq., Moscow, 125047, Russia
[2] Dorodnicyn Computing Center FRC CSC RAS, 40, Vavilov St., Moscow, 119333, Russia
lera.tsvetkova@gmail.com

Key words: Adaptive Mesh, Unstructured Mesh, Parallel Meshing Algorithm, Moving Body, Immersed Boundary Method, Fluid-Structure Interaction.

Abstract. The paper presents a method for simulating the flow of a viscous compressible gas around a moving obstacle, specified by the immersed boundary method on a simply connected unstructured mesh, and the features of its application for bodies of different configuration. An advantage of the developed moving-mesh method is that it keeps the topology of an initial mesh and provides an anisotropic mesh refinement to the “immersed” surface of a streamlined body. The position and shape of the body are determined by the interpolation grid, which stores the distance and normal to the body surface. The paper also describes a special technique to improve the efficiency of adaptation. The technique is based on the prediction of the body location in some automatically chosen time. This trick allows to find the coordinates of mesh vertices during the predicted time interval by interpolation. The integral strategy combining the immersed boundary method with the developed moving-mesh adaptation is tested on several model cases in two- and three-dimensional formulations. However, the paper does not consider the numerical results of the cases, it is mostly targeted at the meshing details.

1 INTRODUCTION

Nowadays the simulation of turbulent flows near moving or deformable bodies of different shapes presents a high interest, and methods for solving such problems are rapidly developing. For instance, “Chimera” type meshes [1] are widely used to simulate cases with moving objects. This method uses two overlapping meshes: one includes all the computational domain and the second one covers the area around the moving body. The main problem of this approach is the loss of the accuracy while exchanging data between the meshes. Besides, this method is not well suitable for multiprocessor systems with distributed memory. Mesh deformation is used to simulate small body movements [2] but is not appropriate for the cases with arbitrary motion of a body. It is also possible to account body movement by local regeneration of the mesh, however this way requires high computational costs.
The immersed boundary method (IBM) allows us to work in a simply connected domain and thereby to deal with arbitrary motion of a body. According to IBM, the mesh nodes are categorized as solid or fluid points. The governing gasdynamic equations are complemented by the extra source terms which act only in solid nodes and model the impact of the immersed boundaries on the flow. In the paper we use the Brinkman penalization method [3,4] which is one of the most developed immersed boundary methods. The usage of simply-connected domains allows to build the method of adaptation that controls not only the dynamic refinement and coarsening but also the redistribution of mesh vertices. The last option allows to keep the original mesh topology and offers an easy use of mesh decomposition techniques for parallel

implementation. The method of node redistribution is based on the minimization of hyperelastic quasi-isometric functional and is formulated both for 2D and 3D cases.
One of the important features of our approach is the use of implicitly given geometries. In the general case, the shape, position, and trajectory of the body cannot be described analytically. We define the shape and location through the level-set octree-type grid which carries the values and gradients of the signed distance function. The level-set grid moves with the body and thereby delivers the information about the distance to the body surface and the normal to it to the main computational mesh through the interpolation.
The quality of the numerically predicted turbulent flow near the solid obstacle is directly connected with the quality of mesh resolution in the vicinity of the body. To improve the mesh resolution in boundary layers, we combine the IBM with the dynamic unstructured-mesh adaptation to immersed body surfaces. We develop the techniques to preserve a smooth changing of mesh-element size throughout the domain and to control the mesh quality near highly curved regions of the geometry. Such mesh adaptation strongly contributes to a better modelling of the immersed boundary conditions. Another advantage of the mesh adaptation is that it offers us a possibility to keep the cloud of refined mesh only around the moving body and thereby to significantly reduce the mesh size.
As 2D model problems for verification and validation of the presented technique we consider cylinder, airfoil and plane projection of a drone rotor. The cylinder case offers the analytically given geometry. The geometry of airfoil is carried on the specially build level-set grid. The feature is the presence of two extremum vertices – at the leading and trailing edges. Adaptation to the shape of rotors has several challenging aspects. Firstly, as in the case of airfoil, the bounding contour of a blade has areas of high curvature and non-smooth regions. To provide the boundary approximation of high quality, we need to smartly distribute anisotropic cells along near-boundary region. Secondly, in the drone case we keep in mind that real drone configuration involves several closely located rotors. To treat multi-rotor configuration, we prohibit the adaptation outside the specified domain which is a circle in the 2D case of rotor projection. This solution allows to resolve conflicts between several bodies moving over the mesh nodes, however, it may complicate the needed refinement of the mesh near the body surface. To alleviate this problem, the adaptation procedure for a rotating propeller is started not from a uniform mesh, but from the specially prepared mesh with the uniformly added vertices along the circle boundary. As a 3D model problem, we consider the case of induced oscillations of a sphere.
To improve the adaptation efficiency, we develop a special algorithm. Solving the variational problem underlying the adaptation method on each time step is costly and often unnecessary. We show that for most cases it is possible to predict rather accurately the body position and therefore the mesh configuration in some time relatively long in comparison with the time step. We name this time interval as time step of adaptation. Then, at each computational time step inside the time step of adaptation time, we use the cheap interpolation procedure, to update the mesh. The efficiency of the developed technique is certainly dependent on the case. In the paper, the results are demonstrated on the considered cases.

2 MATHEMATICAL MODEL

To model turbulent flow in the test problems considered in the paper, we use the Reynolds averaged Navier-Stokes (RANS) equations closed by the Spalart-Allmaras (SA) turbulence model [5]. On the immersed boundaries of the streamlined bodied we impose the no-slip condition that is satisfied by using the Brinkmann penalization method. Following this method, the special terms acting only in solid vertices of the mesh are added to the source terms of SA RANS equations.

We miss the detailed description of the mathematical model since is not so important in this paper primarily devoted to the meshing techniques.

3 ADAPTATION ALGORITHM

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

To introduce the variational problem for mesh optimization one can assume that $x^n(\xi)$ is a quasi-isometric diffeomorphism. The deformation at time t^{n+1} is found as a solution of the variational problem

$$F(x(\xi,t), x^n(\xi)) = \int_{\Omega_\xi} W(Q(x^n,t)\nabla_\xi x(\xi,t)H(\xi)^{-1}) \det H d\xi \qquad (1)$$

We consider the mapping of a triangle with the vertices coordinates (h_0, h_1, h_2) in Lagrangian domain Ω_ξ to a triangle with Eulerian coordinates of the vertices (p_0, p_1, p_2) in domain Ω_x. The Lagrangian coordinates are basically the coordinates of the initial mesh, the Eulerian coordinates are the Cartesian coordinates in the computational domain. For our problem formulation $\Omega = \Omega_x = \Omega_\xi$. $H = (p_1 - p_0, p_2 - p_0)$. Matrix $G_x(x,t) = Q^T Q$ defines a metric tensor in the Eulerian coordinates.

In functional (1) function $W(C)$ defines polyconvex elastic potential (internal energy), which is a weighted sum of the shape distortion measure and the volume distortion measure:

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

In most cases we set $\theta = 4/5$. The elastic potential is minimal when the deformation is isometric, meaning that only rotation and translation are allowed.

Let zero isosurface of the level-set function $u(x,t)$ define the boundary of the moving body. We assume that $u(x,t)$ is close to the signed distance function near the boundary of the domain. In most cases the calculation of the exact distance function is not needed. Instead, some approximate variations are used.

Metric tensor G_x can be written as $G_x = U\Sigma^2 U^T$, where the columns of U are the eigenvectors of G_x, and Σ^2 is the diagonal matrix consisting of parameters σ_i^2, which are eigen values of G_x. At each point of computational domain one can define the local optimal coordinates by affine mapping with matrix $Q = \Sigma U^T$. We say that a computational mesh is optimal (isometric, unit) one if after the affine transformation to optimal coordinates each mesh cell coincides with the cell of initial mesh, or

$$Q\nabla_{\xi} x H^{-1} = V ,$$

where *V* is an arbitrary orthogonal matrix.
A correct definition of the adaptation metric requires normal and tangential directions at each point *p* (the columns of matrix *U*) which are defined by the isosurface of $u(x,t)$ passing through *p* and stretching coefficients σ_i along these directions. When σ_i is large, the local cell size in direction u_i is small, while the ratio of $\max_{i,j} \sigma_i / \sigma_j - 1$ defines the degree of anisotropy. Function $\sigma_1 = \sigma_{normal}(x,t)$ defines the mesh stretching in the normal direction to the body and $\sigma_2 = \sigma_{tangential}(x,t)$ ($\sigma_{2,3}$ in 3D) defines 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.
One can easily get an inconsistent metric which leads to a sharp growth of the quasi-isometric constant of the deformation and to sharp localized cell deformations and size jumps. The most consistent metric which provides the smallest quasi-isometry constants is not known even for very simple bodies. In this work we have developed a special heuristics where normal stretching σ_1 is a constant in the boundary layer and decreases like a hyperbole away from it, while the tangential stretching is defined by the curvature of the surface and gradually subsides away from the boundary layer. Each body is assigned an active spot with radius *D*. Outside this spot metric G_x is just the unity matrix.
The isotropic adaptation metric is defined by equality

$$G(x,t) = \sigma_1^2 I,$$

where $\sigma_1(x,t) = \phi(u(x,t)))$ is the mesh density. In this case, the anisotropy is equal to zero. 1D function $\phi(\cdot): R^1 \to R^1$ is a hyperbole which defines the mesh density as shown in Figure 1 (left). Figure 1 (right) shows the auxiliary normalized one-dimensional mapping from the Lagrangian to the Eulerian coordinates, which guarantees transition from small mesh cells to large mesh cells with the prescribed rate of growth of mesh cell size. The inverse mapping in the transition zone is the logarithmic one, and its derivative defines the hyperbolic mesh density function.
An anisotropic version of metric tensor is defined as follows:

$$G(x,t) = \sigma_1^2 I + (\sigma_2^2 - \sigma_1^2)\nabla_x u \nabla_x u^T \frac{1}{|\nabla_x u|^2} .$$

On highly curved fragments of the body surface or near sharp boundary vertices we set $\sigma_2 = \sigma_1$, otherwise we set $\sigma_2 = \sigma_1 / K$, where $K > 1$ is a user defined anisotropy ratio. Away from the body, σ_2 converges to σ_1 and both converge to unity.

The standard finite element discretization is applied to functional (1). Metric tensor G_x is evaluated in the mesh vertices which is crucial to the stability of very thin and highly compressed mesh layers. To solve optimization problem on each time step, we apply the preconditioned gradient descent technique 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 [6].

4 EFFICIENCY IMPROVMENTS. PREDICTOR-INTERPOLATION SCHEME FOR DEFORMING MESHES WITH GUARANTEED QUALITY

Suppose that metric $G(x,t)$ is defined by signed distance function $d(x,t)$. The boundary surface at time moment *t* is just zero isosurface of *d*, i.e. $\partial\Omega(t)=\{x : d(x,t)=0\}$ computed at time $t=t_n$. It finds deformation direction p_n, such that $x(t_n)+sp_n$ corresponds to the nondegenerate mesh for all $0<s<1$. The value 1 corresponds to large time increment ΔT_n so that $x(t_n+\Delta T_n)=x(t_n)+p_n$. Now for all time $t_{k+1}\le t_n+\Delta T_n$, for each $k=n,n+1,n+2,...$ one can compute

$$x(t_{k+1})=x(t_k)+\frac{\Delta t_k}{\Delta T_n}p_n .$$

To compute time step $\Delta \mathrm{T_n}$ for predictor, we evaluate the minimal normal spatial mesh step size at the boundary of the surface. It is defined by $h_1=\frac{h_0}{\sigma_{\max}}$, where h_0 is uniform mesh size, while $\sigma_{\max}$ is the maximal target stretch in the normal direction. $\sigma_{\max}$ can be computed as the square root of the maximal eigenvalue of metric tensor G at the surface of the moving body. Then

$$\Delta T_n=\frac{h_1}{\max V_n},$$

where V_n is an absolute value of the normal velocity of the body at the surface. The maximum is taken over all the boundary points. We assume that time step Δt_n does not exceed predicted time step ΔT_n.

One should take into account the following two additional criteria which can signal that the mesh changes are too fast for a successful application of the interpolation.

1. The accumulated boundary displacement becomes above h_1:

$$\int_{t_n}^{t_{k+1}} V_n(t)dt>h_1$$

2. $\max\left|d_k\left(x^i\left(t_k\right),t_n+\Delta T_n\right)-d_{k+1}\left(x^i\left(t_k\right),t_n+\Delta T_n\right)\right|>0.25h_1$, where $d_k\left(x^i\left(t_k\right),t_n+\Delta T_n\right)$ is the predicted distance to the body in *i*-th mesh vertex, which is calculated after step k is finished. At this moment, all coordinate $x(t_k)$ are defined. $d_{k+1}\left(x^i\left(t_k\right),t_n+\Delta T_n\right)$ is the predicted distance to the body in *i*-th mesh vertex, calculated before starting step $k+1$. These values can be different if the character of the body movement has changed significantly. The maximum is taken over all the mesh points.

In both these cases, one should recompute the predictor by solving the new mesh deformation problem.

5 NUMERICAL RESULTS

5.1 Oscillating cylinder

Two-dimensional cylinder with diameter $D=1$ makes harmonic oscillations ruled by the equation $y=A\sin(2\pi ft)$, where amplitude $A=0.25$, and frequency $f=0.15915$. There is the upstream flow of Mach number $M=0.1$. Reynolds number $\mathrm{Re}=20000$. To simulate the cylinder motion, the mesh of size of 250 thousand nodes is used. The adapted version of the mesh is shown in Figure 7. The compression degree in the normal direction reaches 30. The first picture in Figure 8 demonstrates the effect of predictor-interpolation scheme. The black line shows the relation of the time spent on the adaptation process to the total computional time at each time step. The black zones correspond to the periods when the interpolation is used between the steps when the predictors are recomputed. The adaptation cost changes rapidly in these zones from high (up to 40% of the total step) to very low (less than 2% of the total step) values. The accumulated adaptation cost in relation to the total computational time is shown by the red line, which tends to ~23%. It is easy to notice that there is a periodicity for both lines. Indeed, the interpolation technique depends on the motion law. That is why when the velocity is low the interpolation is called more often, while in the regions of high body speed the variational problem of adaptation is solved at each time step.

5.2 Plunging of NACA0012 airfoil

Another 2D test case we consider is the problem of the NACA0012 airfoil plunging [10]. The movement law is basically the same as it is for the cylinder (section 5.1), as well as the upstream flow parameters. The mesh of 220 thousand nodes is used for the simulation. The adapted version of the mesh shown in Figure 7 (right). The geometry has both the high curvature regions and sharp corners at the leading and trailing edges of the airfoil. Both these special parts are approximated by mostly isotropic cells. The adaptation technique for approximating the complicated shapes is described in more detail in section 6.3
A similar comparison on the adaptation efficiency is performed for this geometry too. The second picture in Figure 8 shows that the resulting cost of the adaptation is about 40%, but during the time advancing it might jump from 1% to 65%.

5.3 Rotating propeller

As the most complex 2D case, we consider the simulation of the flow near a rotating propeller. As an original geometry of the case, we consider one of the small-scale propellers studied in [3]. The two-dimentional shape is built by the projection of the 3D geometry on the plane $z = 0$. Figure 2 presents the original shape and its projection.
According to [3], the drone rotor of $R = 0.254\, m$ with the hub of size 0. 0127 m is studied. For the 2D plane projecton, the sizes of the blade remains the same. The propeller makes 3000 revolutions per minute meaning the tip velocity is $U_{BL} = 79.8\, m/s$. The upstream flow velocity is $U_0 = 10\ m/s$. In the numerical setup, the size of the propeller is normalized to be equal to 2 ($\tilde{R} = 1$). The Reynolds number defined using the blade tip velocity $\mathrm{Re} = \rho_0 U_{\max} b / \mu_0 = 1.3 \times 10^6$, where $\rho_0 = 1.293\, kg/m^3$ and $\mu_0 = 1.717 \times 10^{-5}\ N \cdot s/m^2$ corresponding to parameters of the air at temperature of $20°C$. There is an upstream flow with Mach number 0.029. To simulate this problem, the RANS equations with SA turbulent model are used. In this paper, we do not present the results of the simulation and consentrate the attention on the aspects of the adaptation techniques. However, the numerical tests have been sucsessfully performed and will be published in our next papers.
The adaptation techniques to resolve the shapes of the objects take into the account that real drones have normally several rotors located rather closely one to another. The adaptation control near two separate bodies may cause conflicts over the mesh vertices in the middle and most likely unpredictable performance. That is why the adaptation is isolated for every rotor meaning that each separate geometry uses only the assigned vertices inside the circle of radius a little larger than the propeller radial size. In three-dimensional formulation, an analog of the

circle is a flat cylinder. In this work, we suppose that the shortest distance between the propellers is about 20 percent of the radius. The mesh adaptation uses only the vertices inside the circle of 1.1 blade length. If the initial mesh has uniform vertex density, the number of nodes will not be enough to reach the needed high refinement of the mesh near the blade surface. Considering this fact, it is more convenient to use a mesh with a nonuniform vertices distribution where the additional nodes are placed near the tip of the blade and in the area of the hub (Figure 3). The areas of high density of vertices relate to the location of the geometrical features that require more nodes for a proper approximation. The minimal size of the mesh elements that is reached in this case is 7×10^{-3}.

It should be noted that the approximation of the curved boundaries in the vicinity of the blade edges using highly anisotropic cells might be very inaccurate. To reach a proper boundary approximation, the level of anisotropy of the cells near the body should depend on the boundary curvature. At the current stage of our research, the features of the body surface are defined manually. Basically, for each feature there is a circle which corresponds to it. Inside these circles the cells of the mesh have more isotropic shape and slowly increase the anisotropy level as going into the flat regions of the boundary. Figure 4 shows the circles associated with the geometry features and the corresponding adapted mesh. The blade edges are approximated by almost isotropic triangles.

The illustration of the process of the control of the adaptation is show in Figure 5. Parameter σ_1 which controls the mesh stretching in the normal direction is uniform and maximal along the whole boundary and slowly decreases in more distant nodes. Parameter σ_2 reaches its maximum near the corners of the blade boundary. The lower picture presents the distribution of the anisotropy level of the cells of the adapted mesh, which is the relation of the cell size compression in the normal direction to the compression in the tangential one. The picture shows that the compression of 30 times in the normal direction produces the anisotropy level up to 92 on the flat regions of the boundary.

Considering the problem of adaptation to the shape of the blades it is important to maintain an adequate mesh size while still reaching a high compression degree. For the drone rotor case, we use the mesh of 87 thousand nodes. The maximum compression degree we reach is 30. A small size of the initial mesh will hopefully allow us to transfer to the full three-dimensional formulation without significant changes of the total algorithm.

Figure 6 presents a fragment of the initial mesh and the same fragment after 10 full revolutions. After the first rotation the mesh slightly changes as compared to the initially adapted mesh, however no further degradation is observed. During all the other rotations the mesh preserves the structure close to the one shown in Figure 6 (right). It is important to mention that the size of the first boundary cell in the normal direction remains approximately the same during all the revolutions.

The third picture in Figure 8 shows that the relation between the interpolation and prediction steps is more uniform during the rotation. It can be easily explained by the fact that the rotation velocity is constant. The total cost of the adaptation takes about 30% of the total computational time.

5.4 Oscillating sphere

To illustrate the work of the adaptationn method in 3D formulation, we chose the case with the oscillating sphere. The sphere of diameter $D=1$ makes harmonic oscillations ruled by the law $y = A\sin(2\pi ft)$, where amplitude $A=0.2$, and frequency $f=0.15$. There is no upstream flow. The Reynolds number is defined using the maximum of the body velocity $\mathrm{Re} = \rho_0 U_{\max} b / \mu_0 = 318.3$. To simulate the sphere motion, the mesh of 1 million nodes is used.

The fourth picture in Figure 8 shows that the interpolation is used in most of time steps.

6 CONCLUSIONS

The paper presents the method to simulate the flow near moving bodies of complicated shapes by using IBM in combination with the anisotropic mesh adaptation of unstructured mesh. The IBM allows for considering the problems in the simply connected domains. Our method of mesh adaptation treats the geometrical features of streamlined bodies by controlling the level of anisotropy of the cells in the vicinity of body surface. The paper also introduces the predictor-interpolation scheme to enhance the efficiency of the adaptation process and to reduce its computational cost. It allows to solve the costly variational problem of adaptation only at the time steps when some specific criteria are satisfied. The efficiency of the predictor-interpolation scheme is strongly dependent on the case, the physical time step, and the type of body movement.

All these improvements of the adaptation algorithm are tested on several 2D and 3D model cases. In the problems considered, the adaptation procedure spend from 5% to 40% of the total computational costs. The analysis shows that the developed method can be applicable to the wide variety of problems.

In the near further we plan to improve the adaptation efficiency by implementing the OMP parallel model to the current MPI solution and to test the developed techniques on more complicated three-dimensional cases.

ACKNOWLEDGMENTS

The calculations were performed on the supercomputers installed in the Supercomputer Center of Collective Usage of KIAM RAS.

The work was funded by Russian Science Foundation within Project № 20-41-09018.

PICTURES

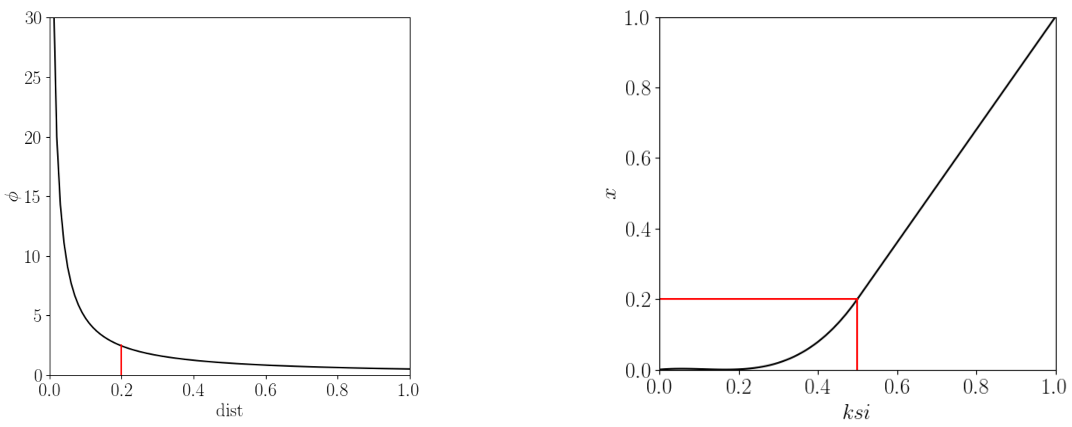


Figure 1: Left: mesh density function, right: normalized 1D mapping defining stretching in the normal direction.

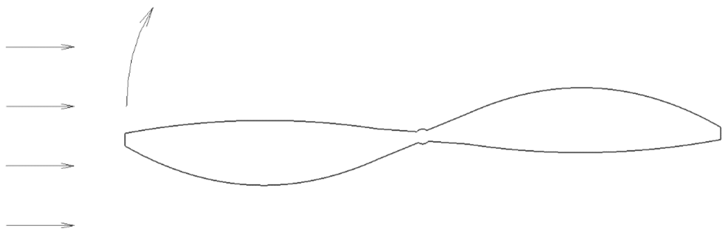


Figure 2: Original rotor geometry and 2D contour.

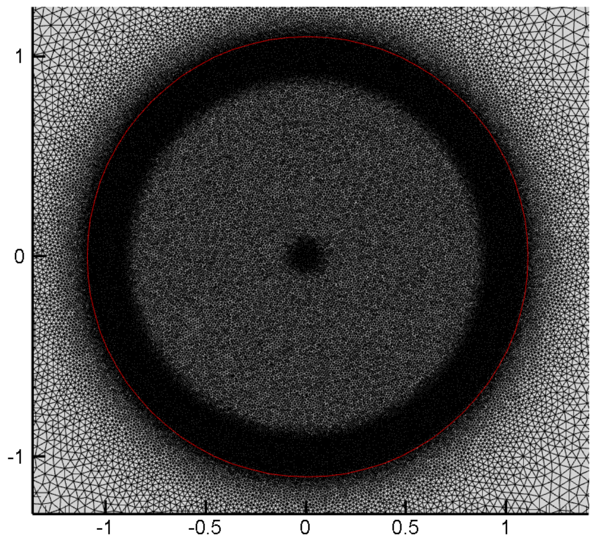


Figure 3: The original nondeformed mesh with additional vertices in the location of the features of the propeller. The vertices outside red circle are not involved in adaptation.

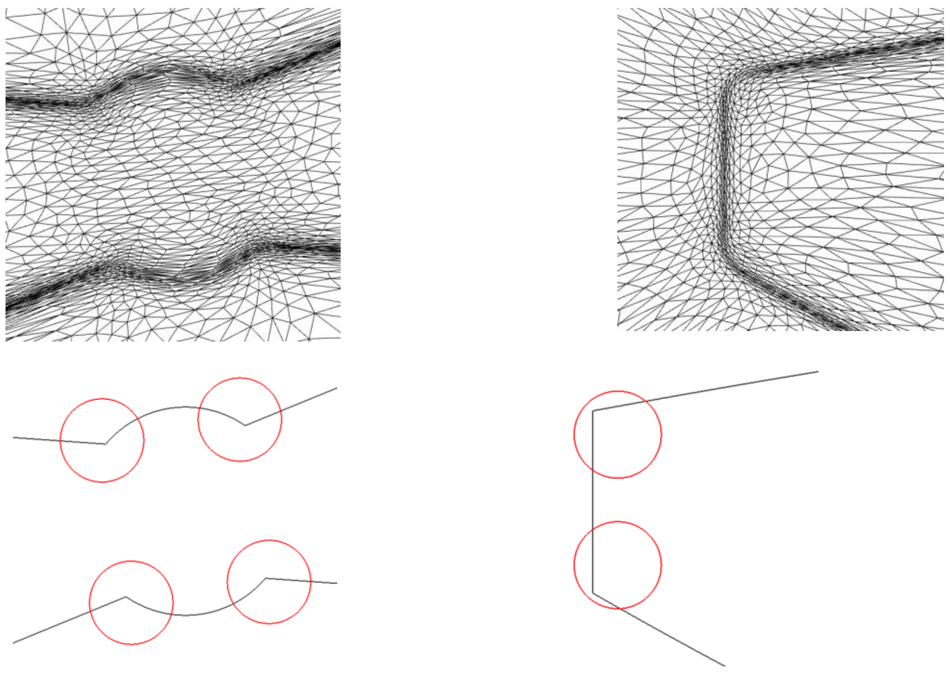


Figure 4: Upper row – fragments of the computational mesh near featured regions. Lower row – corresponding boundary fragments where red circles define regions of isotropic mesh adaptation.

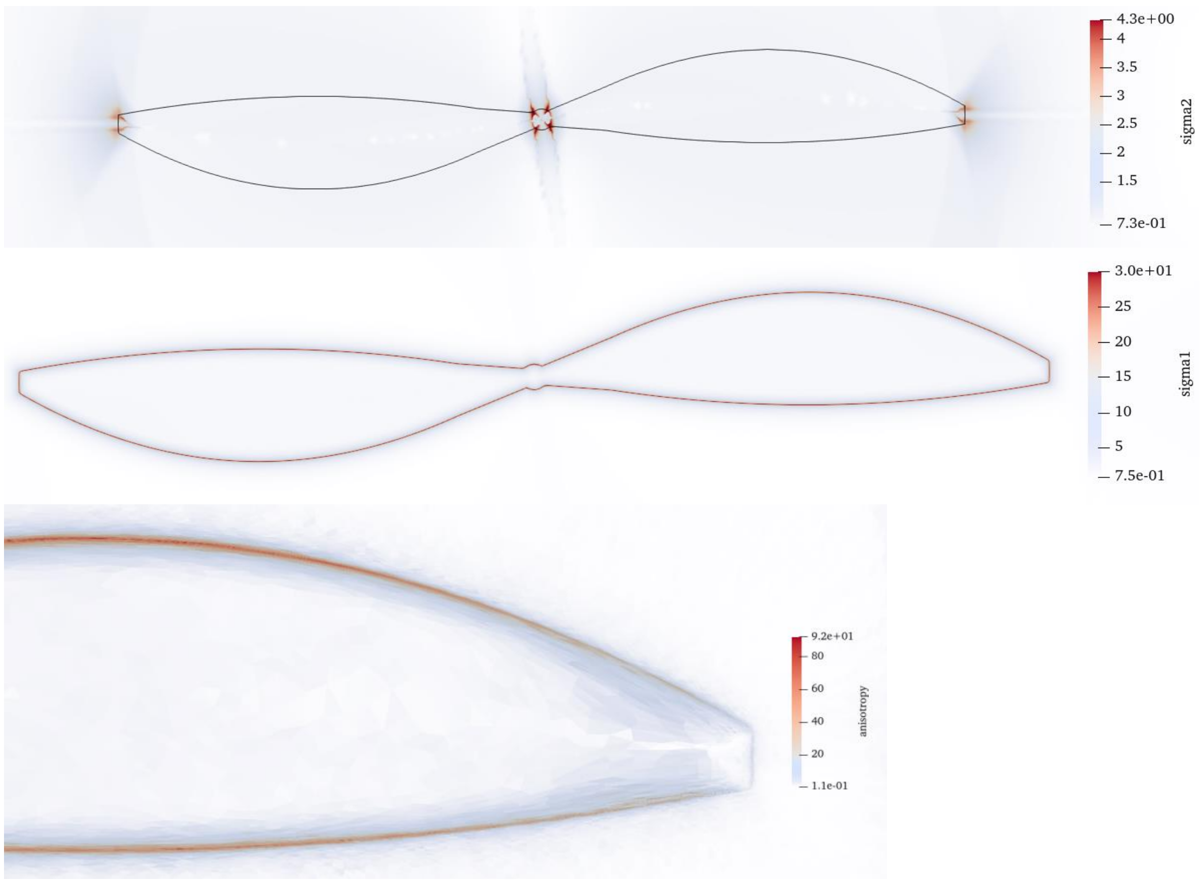


Figure 5: Upper picture – distribution of σ_2. Middle picture – distribution of σ_1. Lower picture – anisotropy distribution.

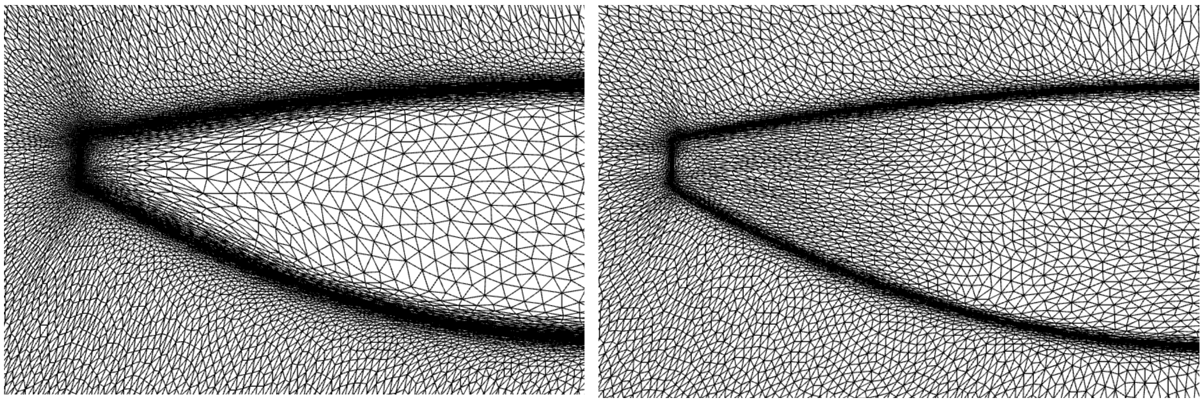


Figure 6: The fragment of the adapted mesh after stationary adaptation before the start of the simulation and after 10 full rotations.

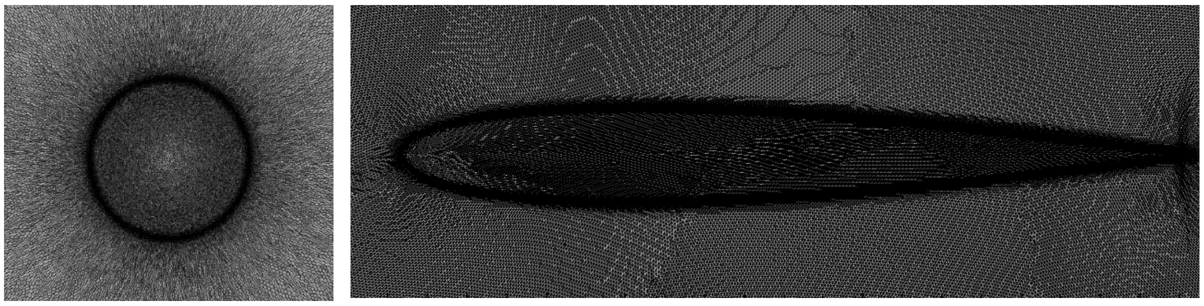
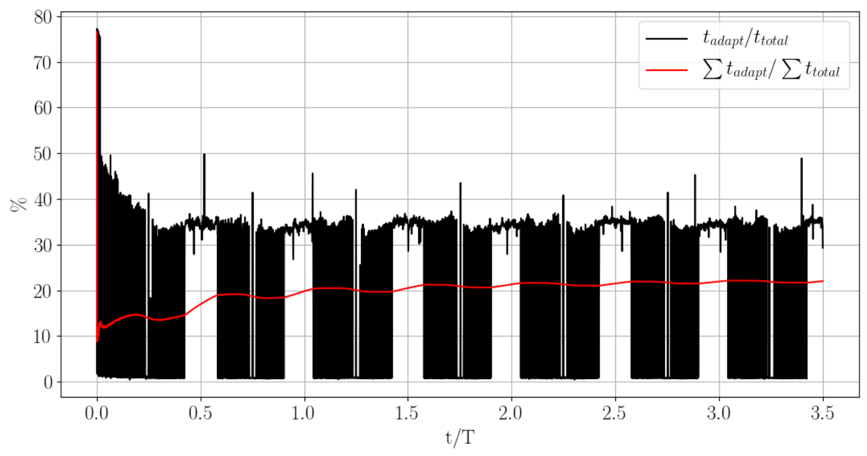
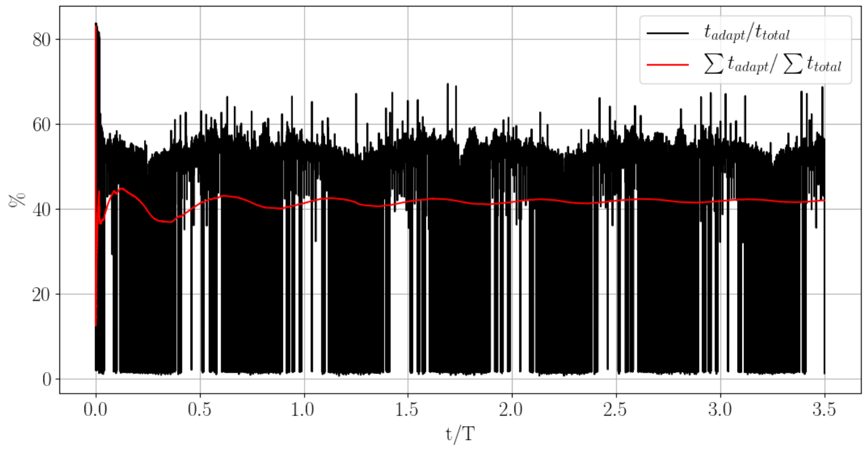
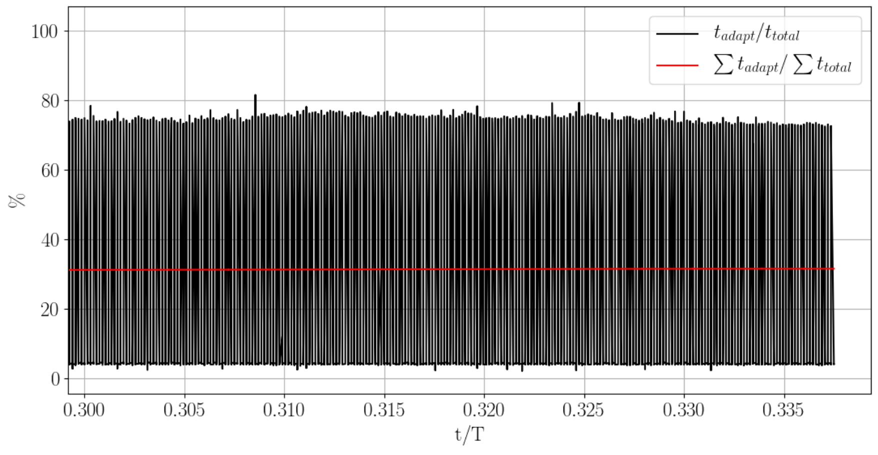


Figure 7: Fragments of the mesh adapted to the cylinder shape and to the shape of NACA0012 aircraft.







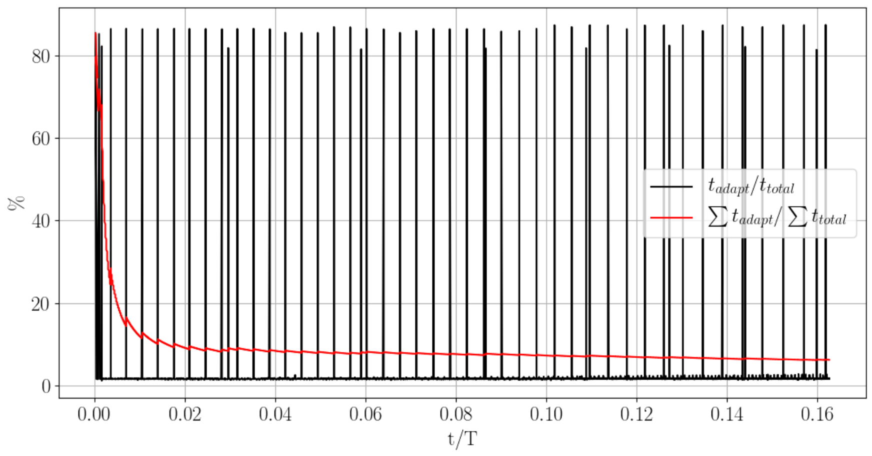


Figure 8: Black: relation of the adaptation time to the total time step at each time iteration, red: relation of the total adaptation time to the total time stepping duration. Pictures correspond to cylinder, NACA0012, propeller, sphere, respectively.

REFERENCES

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

[2] J. Z. Chai, X. L. Han, H. L. Che, A Fast Grid Deformation Algorithm And Its Application, 26th Congress of International Council of the Aeronautical Sciences, 14 - 19 September 2008, Anchorage, Alaska, USA. Paper ICAS2008-P2.4 (2008)

[3] Ph. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows. Numer.Math.81(4) (1999) 497–520.

[4] I.V. Abalakin, T.K. Kozubskaya, N.S. Zhdanova, Immersed boundary penalty method for compressible flows over moving obstacles, ECCM 6/ECFD 7 (2018) 1797.

[5] P. R. Spalart, S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, 30th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings. — AIAA Paper 1992-0439.

[6] V. Garanzha, L. Kudryavtseva, Hypoelastic Stabilization of Variational Algorithm for Construction of Moving Deforming Meshes. In: Evtushenko Y., Jacimovic M., Khachay M., Kochetov Y., Malkova V., Posypkin M. (eds) Optimization and Applications. OPTIMA 2018. Communications in Computer and Information Science, 974 (2019) 497-511.

[7] I.E. Kaporin, O.Yu. Milyukova, MPI+OpenMP implementation of the BiCGStab method with explicit preconditioning for the numerical solution of sparse linear systems. Numerical methods and programming, 20 (2019) 516-527

[8] J. B. Brandt, Small-scale propeller performance at low speeds : дис. – University of Illinois at Urbana-Champaign (2005).

[9] http://airfoiltools.com/airfoil/details?airfoil=n0012-il