

A MESH ADAPTATION METHOD FOR SIMULATION OF UNSTEADY FLOWS

C. H. Zhou*

* Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics,
Nanjing, 210016, China

Keywords: *mesh adaptation, unsteady flow, immersed boundary, moving boundary*

Abstract

In this paper, a mesh adaptation technique for simulation of unsteady flow with moving-boundaries is presented. In each adaptation period, the mesh is refined in the regions where the phenomena evolve and is coarsened in the regions where the phenomena deviate since the last adaptation. A simple indicator of mesh adaptation that accounts for the solution progression is defined. The unsteady flow and the fluid-solid interface are recomputed on the adapted mesh. There is no phase shift in time between the computed solution and the adapted mesh, and the frequency of mesh adaptation can also be controlled to reduce the interpolation errors due to the solution transferring. To validate the present method, several unsteady flows with fixed or moving boundaries have been simulated.

1 Introduction

Techniques of mesh adaptation have been extensively explored as an indispensable approach not only to pursue an accurate numerical solution but also to reduce the computational cost. For time-dependent problems, mesh adaptation is more crucial as the physical phenomena may progress arbitrarily in the computational domain.

In most of the existing methods of mesh adaptation for unsteady flows, the approach taken is to simply adapt the mesh per n time-steps using the initial solution of this period to construct the adaptation indicators. Therefore, the adapted mesh always lags behind the unsteady solution. The features of interest may

move outside the refined region. In order to reduce the lag and contain the feature evolution within the resolved region, the mesh was adapted frequently [1-3]. In this situation, an important source of errors due to solution transferring from the old mesh to the current adapted mesh is introduced. In the approaches of [1] and [2], two layers of cells adjacent to the marked critical region are refined to ensure that the marked region contains the traveling feature. In the mentioned approaches, the adaptation frequency cannot be controlled and the errors due to the interpolation of solution from the previous mesh to the adapted one cannot be prevented effectively. In [4], Cavallo et al. proposed a mesh adaptation technique for transient flows, which is based on a new projection and error-wake concept. Mesh refinement is performed by projecting the error ahead of its current position. Mesh coarsening is performed in the “wake” region where the errors have propagated. Alauzet et al. [5-6] proposed an approach, in which the flow-field and mesh are computed iteratively over each period till their coupling is converged. The mesh adaptation is based on a metric intersection in time procedure and reflects the flow-field evolving during that period.

In this paper, we present an approach for mesh adaptation in the simulation of an unsteady flow over moving immersed boundaries. For the remainder, the content is arranged as follows. In Section 2, the basic numerical method is described in brief. In Section 3, the mesh adaptation approach is presented. In Section 4, numerical results are presented and compared with experimental data or the published results obtained on

conventional meshes. Finally, in Section 5, we summarize this work and present the conclusions.

2 Basic Numerical Methods

The Navier-Stokes/Euler equations are solved by using the so-called domain-free discretization (DFD) method. This method has been described in detail in [7-8], so only a brief description is given herein. In the DFD method, a partial differential equation is discretized at all mesh points inside the solution domain, but the discrete form at an interior point may involve some points outside the domain, which serve as the role to implement the boundary condition. The critical issue for successful implementation of the DFD method is how to calculate the functional values at the exterior dependent points, i.e. to construct some approximate form of solution in the vicinity of the wall boundary. The functional values are updated at each time step by proper extrapolation along the direction normal to the wall boundary in conjunction with the no-slip (for viscous flows) or no penetration (for inviscid flows) conditions and the simplified momentum equation in the vicinity of the wall. The Galerkin finite-element approximation [9] is employed for spatial discretization, and the discrete equations are integrated in time via a dual-time-stepping scheme [10].

The DFD method belongs to the non-boundary-conforming methods. Therefore, the moving-boundary flows can be simulated on a fixed mesh and there is no need to update the mesh at each time-step in order to follow the motion or deformation of the solid object.

3 Mesh Adaptation for an Unsteady Flow with Immersed Moving-Boundaries

The conventional strategy of mesh adaptation for unsteady flows is to adjust the mesh per a specified number n of time-steps only according to the solution at the first time-step of this period, so the mesh always lags behind the unsteady solution. If the mesh is adapted frequently to diminish the lag, an important

source of errors due to solution transferring (by interpolation) from the old mesh to the newly-adapted one is introduced. For a long-time scale simulation, the interpolation errors may accumulate and the solution accuracy will decrease quickly.

In this work, mesh adaptation is also performed per n time-steps (one adaptation period), but the mesh is adapted to the phenomenon evolving in each period. This is the principle of the present approach.

3.1 Adaptation Strategy

In the present strategy, mesh adaptation is performed per n time-steps. Starting from the initial solution for each adaptation period, the instantaneous solution and fluid-solid interface at each time-step is predicted firstly. Then, an adaptation indicator that takes into account the solution progression is calculated, and a new adaptive mesh is generated for this period. After that, the initial solution on the previous mesh is transferred onto the newly-adapted mesh. Finally, the computation is restarted on the new mesh to obtain the time-dependent solution and the position of the evolving fluid-solid interface for this time period. For each adaptation period, the mesh is refined in the regions where the solution evolves and is coarsened in the regions where the phenomena or the solid objects deviate since the last adaptation. Using the adapted mesh as the initial mesh and the solution at the last time-step as the initial solution, the next adaptation period can be started.

The solution obtained on the old mesh must be recovered on the current adapted mesh to restart the computation from the previous state. This stage is critical in the mesh-adaptive simulation of unsteady flows as the errors due to solution transferring may accumulate throughout the long-time computations. The preservation of conservation property is also mandatory. Therefore, a P1-conservative interpolation [11] is used in the present method to transfer solution from the previous mesh to the current mesh.

In the present adaptation procedure, the mesh is adaptive to solution progression, so

there is no lag between the adapted mesh and the computed solution. Also, there is no need to adapt the mesh frequently and the errors due to solution-transferring can be controlled. In our numerical experiments, n is determined empirically to be 50 ~ 100, which is much larger than the values ($n = 5 \sim 10$) taken in those methods [1-3] which are based on the frequent mesh adaptation.

For more details of description and discussion of this mesh adaptation strategy, the readers are referred to our previous paper [12].

3.2 Criteria for Mesh Adaptation

It is necessary to define an indicator of mesh adaptation which can determine automatically the zones of mesh where some refinement or coarsening are required. In this work, the gradient of density is employed to capture shock waves, and the vorticity is used to capture boundary layers and vortices.

For unsteady flows, the adaptation indicator must take into account the solution progression since the physical phenomena develop with time. We solve this problem by defining the indicator of mesh adaptation for each triangle K during the i th adaptation period as

$$E_i(K) = \max_{j_0+1 \leq j \leq j_0+n} \left\{ |\nabla \times \mathbf{u}|_j^K + |\nabla \rho|_j^K \right\} \quad (1)$$

where j is the index for time-step. For the piecewise linear solution, the magnitude of gradient of density and the magnitude of vorticity on each triangle can be calculated as below

$$\begin{aligned} |\nabla \times \mathbf{u}|_j^K &= A_K \left| \frac{\partial u_1}{\partial x_2} - \frac{\partial u_2}{\partial x_1} \right|, \\ |\nabla \rho|_j^K &= A_K \sqrt{\left(\frac{\partial \rho}{\partial x_1} \right)^2 + \left(\frac{\partial \rho}{\partial x_2} \right)^2} \end{aligned} \quad (2)$$

where A_K is the area of element K . For each vertex P , the adaptation indicator is defined as

$$E_i(P) = \max_{K \in \Omega_p} E_i(K) \quad (3)$$

where Ω_p is the sub-region formed by all triangles that share the vertex P .

Two values of E_{\max} and E_{\min} are specified for mesh adaptation. In the i th mesh adaptation

period, when $E_i(K) > E_{\max}$ the element K is refined and when $E_i(K)$ or $E_i(P) < E_{\min}$ the element K or the vertex P is deleted. In the non-boundary-conforming methods, such as the DFD method, the mesh in the vicinity of wall should be fine enough to reflect the effect of the solid boundary. So, the elements near the wall are specified large values of indicator to ensure that the mesh in this region will be the finest. To prevent extremely large gradient of mesh density and keep the geometric integrity of the domain as a whole, the original mesh will not be coarsened further in the mesh adaptation.

3.3 Refinement and Coarsening

In this work, the original triangular mesh is generated by dividing the square cells of a uniform Cartesian mesh. Therefore, the original mesh consists of equilateral right-angled triangles. Two techniques are developed to generate adaptive meshes dynamically.

In the first method, a multi-level refinement/coarsening strategy is employed. The mesh is regularly and successively refined by dividing a triangle into four similar triangles. This creates a tree structure of triangles. Only a refined triangle will possibly be unrefined by deleting its four sons. In order to eliminating the ‘‘hanging point’’ generated in the regular refinement and coarsening, the triangle with one hanging point is irregularly refined into two triangles by connecting the hanging point to the opposite vortex of the triangle. The generation of a newly-adapted mesh begins by removing all triangles generated by irregular refinement in the last mesh adaptation. Then, the regular refinement starts from the top level. In the refinement process, any triangle not marked for refinement will be refined if it has more than one hanging points or it has more than one neighbors that have been regularly refined [13]. The coarsening starts also from the top level. In the coarsening process, any triangle marked for coarsening will not be coarsened if the number of its neighbors that will not be coarsened is greater than one. If a triangle is not marked for coarsening, the neighbors of its farther triangle will not be coarsened.

In the second method, the refinement procedure is based on the longest-edge-bisection algorithm proposed by Rivara [14]. The selected element is bisected along its longest edge by the midline. The neighbor element sharing the longest edge is also bisected in the same way. The process is recursively applied to the longest edge neighbors till it stops. The coarsening step consists of deleting vertices from the mesh, rather than directly deleting elements. Each vertex P is associated with a sub-region Ω_p . In the coarsening process, the selected vertex and all its incident edges are eliminated from the mesh. The sub-region associated with this vertex is then re-triangulated.

A value of minimal mesh size is specified to control the mesh density and prevent numerical instability due to the elements with very small volume.

Employing the first method, all the triangles on the adapted mesh except those obtained from irregular refinement are similar to the triangles on the original mesh, and any triangle is irregularly refined at most one time. So, after mesh adjusting, the detriment of the geometrical quality of triangles can be controlled. Its disadvantage is the record of some history information, while there is no extra requirement of memory in the second method.

4 Numerical Experiments

In this section, numerical experiments for unsteady flows with stationary or moving boundaries are performed to validate the present mesh adaptation method. E_{\max} is related to the mesh size. The value of E_{\max} are determined empirically. In the following computations, we set $E_{\max} = 2.5 \times 10^{-3}$ for the circular cylinder and the NACA0012 airfoil and $E_{\max} = 5 \times 10^{-4}$ for the swimming fish, corresponding to different expected minimal mesh sizes. In all computations, we always set $E_{\min} = E_{\max} / 10$.

The longest-edge-bisection/vortex-deleting strategy is employed to generate dynamic adaptive meshes in the first test case while the

multi-level refinement/coarsening strategy is employed for all the other cases.

4.1 Unsteady Flow over a Stationary Circular Cylinder for $Re=300$

A large domain of $50D \times 30D$ has been used to minimize the outer boundary effects. The initial mesh has 126×76 nodes. The non-dimensional time-step size is taken to be 10^{-2} . The mesh is adapted per $n = 50$ time-steps.

The adapted mesh in the wake region near the cylinder for the period $[75, 75.5]$ and the vorticity field at the end of this time period are shown in Figure 1. The node number of the current mesh is 47995. In Figure 2, the initial position (blue) and the end position (red) of vorticity progressing in this period are plotted. It can be seen from these figures that the mesh in the region of vorticity convection is refined while mesh in the region where the phenomena deviate is coarsened, and the mesh density matches well the intensity of the vorticity.

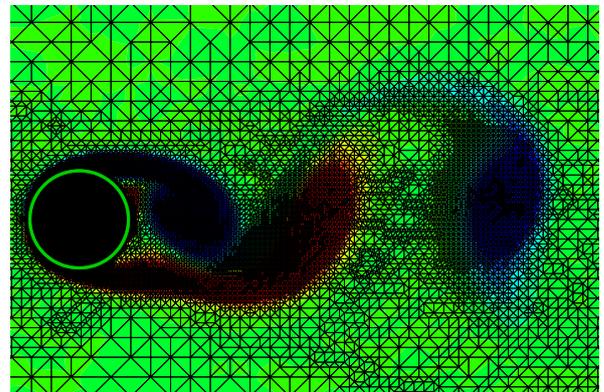


Fig. 1. Adapted Mesh and Vorticity Distribution in the Wake Region for $Re=300$

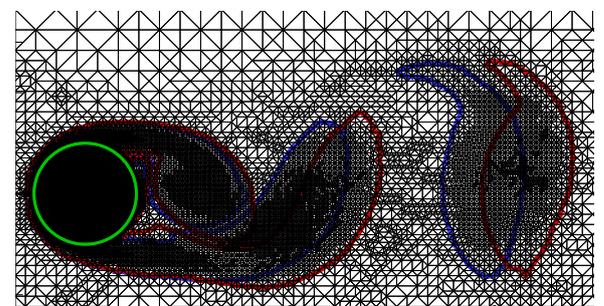


Fig. 2. Evolving of Vorticity in One Adaptation Period for $Re=300$

4.2 Transonic Inviscid Flow around an Oscillating NACA0012 Airfoil

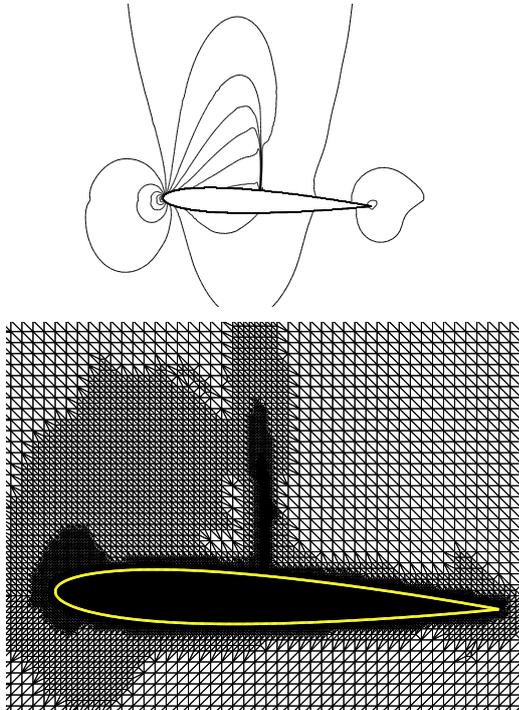
In this subsection, the performance of the present method is investigated with the simulation of the inviscid transonic flow at $M_\infty = 0.755$ past the NACA0012 airfoil undergoing forced oscillation in pitch

$$\alpha(t) = 0.016^\circ + 2.51^\circ \sin(0.1618t) \quad (4)$$

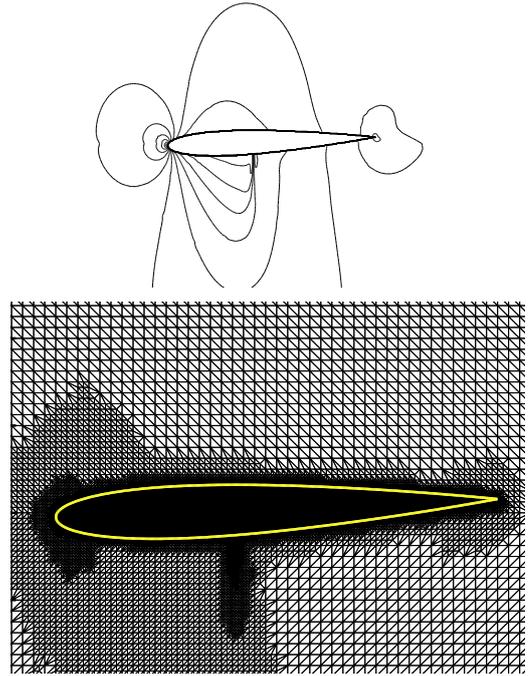
about the quarter-chord point. It is the CT5 case of the experiment of AGARD [15]. The flow is characterized by the presence of a strong shock wave, which develops alternatively on the upper and lower surface of the airfoil. The non-dimensional time-step size is taken to be 10^{-2} , and the mesh adaptation is performed per every $n = 50$ time-steps.

The contours of Mach number and the adapted meshes are displayed in Figure 3. It can be seen clearly that the mesh in the regions near body surface and the shock wave has been refined dynamically.

The hysteresis curve of the lift coefficient is depicted in Figure 4. DFD Euler results on uniform mesh [16], the experimental data [15] and Euler results obtained on adaptive body-fitted grid [17] are also plotted in this figure for comparison. We can see that the numerical results agree well with each other.



(a) $\alpha = 2.34^\circ$ (downward)



(b) $\alpha = -2.41^\circ$ (downward)

Fig. 3. Contours of Mach Number and Adapted Meshes for Oscillating NACA0012 Airfoil

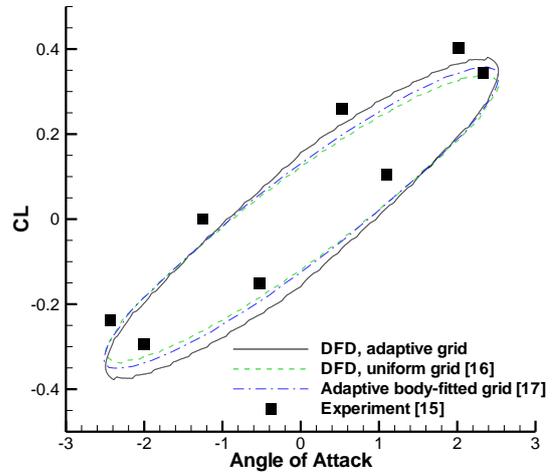


Fig. 4. Hysteresis Curve of Lift Coefficient for Oscillating NACA0012 Airfoil

4.3 Simulation of Self-propelled Anguilliform Swimming

Here, we apply the proposed mesh adaptation method to the numerical simulation of self-propelled anguilliform swimming. The deformation of the fish body is prescribed in the local body system. The position and orientation of the fish body are defined in the global system, which are determined by the fluid force. The algorithm for loose coupling of fluid-structure

interaction presented in [18] is employed to simulate the self-propelled swimming. The fish geometry and the prescribed deforming motion of the body are taken from the work of Kern and Koumoutsakos [19].

All simulations are conducted with a constant viscosity of $\mu = 1.4 \times 10^{-4}$, body length $L = 1$, density $\rho_{\text{fluid}} = \rho_{\text{body}} = 1$, and undulation frequency $f = 1$. The size of computational domain is 40×12 and the initial mesh has 101×31 nodes. The time-step size is taken to be 5×10^{-4} . The mesh is adapted per every $n = 100$ time-steps.

In Table 1, the asymptotic forward velocity, the amplitude of lateral velocity, and the amplitudes of the coefficients of longitudinal force, lateral force and yaw moment are compared with the results of Zhou and Shu [18], Kern and Koumoutsakos [19]. All numerical results agree well with each other. Due to the reduction of the numerical dissipation, the asymptotic mean forward velocity of the present computation is slightly larger than those obtained on conventional meshes.

Table 1. Results for a Swimming Fish

References	\bar{V}_{\parallel}	\tilde{V}_{\perp}	\tilde{C}_{\parallel}	\tilde{C}_{\perp}	\tilde{C}_M
Present work	0.557	0.038	0.031	0.038	0.030
Zhou et al. [18]	0.550	0.039	0.030	0.040	0.031
Kern et al. [19]	0.540	0.04	0.03	0.04	0.03

\bar{V}_{\parallel} : mean value of V_{\parallel} ;
 $\tilde{V}_{\perp}, \tilde{C}_{\parallel}, \tilde{C}_{\perp}, \tilde{C}_M$: amplitudes of $V_{\perp}, C_{\parallel}, C_{\perp}, C_M$

The global view of vorticity field at $t = 15$ is shown in Figure 5. The partial view of vorticity distribution in the region near the fish tail at $t = 15$ and the adapted mesh in the adaptation period $[14.95, 15]$ are shown in Figure 6. The solid object is moving relative to the mesh. The flow is only induced by the motion of the fish body, so the convection velocity of vortex is small. In Figure 6, the triangles concentrated around the vortices have been refined obviously.

Figure 7 illustrates the motion and deformation of the fish within this adaptation

period. The red line represents the position and shape of the body at the start instant, and the green represents those at the end instant. During each adaptation period, the mesh in the region through which the fish is passing has been refined dynamically. So the effect of geometry can be reflected accurately in the non-boundary-conforming method, and the boundary layer adjacent to the solid object can also be resolved.

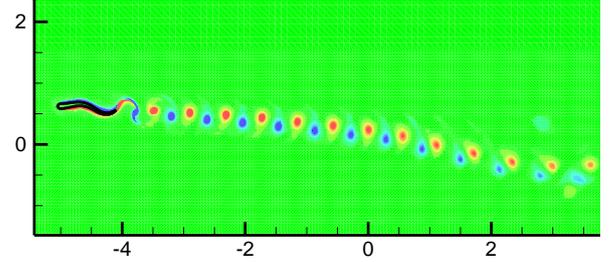


Fig. 5. Global View of Vorticity Field around a Swimming Fish at $t = 15$

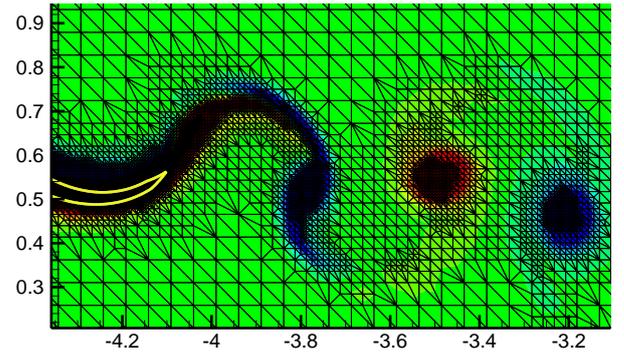


Fig. 6. Close up View of Adapted Mesh and Vorticity Distribution near the tail at $t = 15$

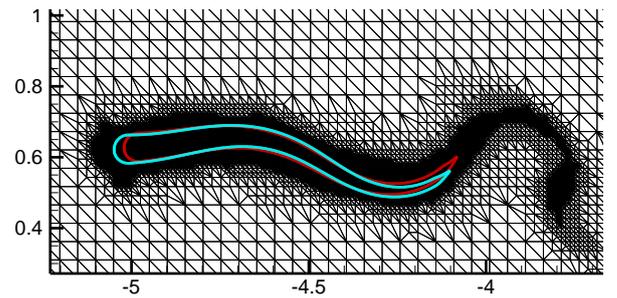


Fig. 7. Motion and Deformation of the Fish Body during $[14.95, 15]$

5 Summary and Conclusions

In this work, we have developed a mesh adaptation method for unsteady flows with moving immersed boundaries. Via a predictor-corrector step, the mesh in each adaptation

period is adapted to the phenomena evolving and the motion or deformation of solid objects. There is no phase shift in time between the solution and the adapted mesh. The frequency of mesh adaptation can be controlled to reduce the errors due to solution transferring. The number of mesh nodes can be reduced greatly when using a non-boundary-conforming method to simulate complex moving-boundary flows.

Acknowledgement

This work has been supported by National Science Foundation of China under Grant 11072113.

References

- [1] Lohner R and Baum J D. Adaptive h-refinement on 3D unstructured grids for transient problems, *Int. J. Num. Meth. Fluids*, Vol. 14, pp 1407-1419, 1992.
- [2] Speares W, Berzins M. A 3D unstructured mesh adaptation algorithm for time-dependent shock-dominated problems, *Int. J. Num. Meth. Fluids*, Vol. 25, pp 81-104, 1997.
- [3] Baker T J. Adaptive modification of time evolving meshes, *Comput. Methods Appl. Mech. Engrg.*, Vol. 194, pp 4977-5001, 2005.
- [4] Cavallo P A, Arunajatesan S, Sinha N and Baker T J. Transient mesh adaptation using an error wake and projection method, *AIAA-2006-1149*, 2006.
- [5] Alauzet F, George P L, Mohammadi B. 3D transient mesh adaptation for time-dependent problems: Application to CFD simulations. *J. Comput. Phys.*, Vol. 222, pp 592-623, 2007.
- [6] Frey P J, Alauzet F. Anisotropic mesh adaptation for CFD computations, *Comput. Methods Appl. Mech. Engrg.*, Vol. 194, pp 5068-5082, 2005.
- [7] Zhou C H, Shu C and Wu Y Z. Extension of domain-free discretization method to simulate compressible flows over fixed and moving bodies, *Int. J. Numer. Meth. Fluids.*, Vol. 53 pp 175-199, 2007.
- [8] Zhou C H and Shu C. A local domain-free discretization method for simulation of incompressible flows over moving bodies. *Int. J. Numer. Meth. Fluids.*, Vol. 66, pp 162-182, 2011.
- [9] Mavriplis D J and Jameson A. Multigrid solution of Navier-Stokes equations on triangular meshes, *AIAA J.*, Vol. 28, No. 8, pp 1415-1425, 1990.
- [10] Belov A, Martnelli L and Jameson A. Three-dimensional unsteady incompressible flow calculations using multigrid, *AIAA-1997-0443*, 1997.
- [11] Alauzet F and Mehrenberger M. P¹-conservative solution interpolation on unstructured triangular meshes, *Int. J. Numer. Meth. Engng.*, Vol. 84, pp 1552-1588, 2010.
- [12] Zhou C H and Ai J Q. Mesh adaptation for simulation of unsteady flow with moving immersed boundaries, *Int. J. Num. Meth. Fluids*, Vol. 72, pp 453-477, 2013.
- [13] Welfert B D. A posteriori error estimates and adaptive solution of fluid flow problems. PhD. Thesis of University of California at San Diego, 1990.
- [14] Rivara M C. Mesh refinement processes based on the generalized bisection of simplicities, *SIAM Journal of Numerical Analysis*, Vol. 21, pp 604-613, 1984.
- [15] Landon R H. Compendium of Unsteady Aerodynamic Measurement, *AGARD Report*, No. 702, 1983.
- [16] Zhou C H. Improvement and application of the Local Domain-Free Discretization method for numerical simulation of compressible flows, *Chinese Journal of Computational Mechanics*, Vol. 27, No. 5, pp 874-880, 2010.
- [17] Bramkamp F, Lamby P H, Muller S. An adaptive multiscale finite volume solver for unsteady and steady state flow computations, *J. Comput. Phys.*, Vol. 197, pp 460-490, 2004.
- [18] Zhou C H and Shu C. Simulation of self-propelled anguilliform swimming by local domain-free discretization method. *Int. J. Numer. Meth. Fluids.*, Vol. 69, pp 1891-1906, 2012.
- [19] Kern C and Koumoutsakos P. Simulations of optimized anguilliform swimming, *The Journal of Experimental Biology*, Vol. 209, pp 4841-4857, 2006.

Contact Author Email Address

mailto:chzhou@nuaa.edu.cn

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS 2014 proceedings or as individual off-prints from the proceedings.