Morphing Numerical Simulation of Incompressible Flows Using Seamless Immersed Boundary Method

Kyohei Tajiri(✉), Mitsuru Tanaka, Masashi Yamakawa, and Hidetoshi Nishida

Department of Mechanophysics, Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto, Japan
tajiri@kit.ac.jp

Abstract. In this paper, we proposed the morphing simulation method on the Cartesian grid in order to realize flow simulations for shape optimization with lower cost and versatility. In conventional morphing simulations, a simulation is performed while deforming a model shape and the computational grid using the boundary fitting grid. However, it is necessary to deform the computational grid each time, and it is difficult to apply to a model with complicated shape. The present method does not require grid regeneration or deformation. In order to apply the present method to models with various shapes on the Cartesian grid, the seamless immersed boundary method (SIBM) is used. Normally, when the SIBM is applied to a deformed object, the velocity condition on the boundary is imposed by the moving velocity of the boundary. In the present method, the velocity condition is imposed by zero velocity even if the object is deformed because the purpose of the present morphing simulation is to obtain simulation results for a stationary object. In order to verify the present method, two-dimensional simulations for the flow around an object were performed. In order to obtain drag coefficients of multiple models, the object was deformed in turn from the initial model to each model in the present morphing simulation. By using the present method, the drag coefficients for some models could be obtained by one simulation. It is concluded that the flow simulation for shape optimization can be performed very easily by using the present morphing simulation method.

Keywords: Computational fluid dynamics · Morphing simulation method · Immersed boundary method · Incompressible flow · Shape optimization

1 Introduction

There are many products around us that are closely related to the flow phenomenon. Improvements in the performance of these products are always

Supported by organization x.

V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12143, pp. 75–87, 2020.
https://doi.org/10.1007/978-3-030-50436-6_6

expected. On the other hand, reducing the time and cost required to develop these products is also an important issue. Shape optimization through flow simulations at the stage of design is one of these efforts. By determining the optimum shape from many candidate product shapes (candidate models) at the early stage of product development, the effort of the redesign is reduced. As a result, development costs are reduced. Conventionally, flow simulations have been performed for each of these many candidate models. However, in recent years, the cost required for flow simulations has increased because the number of candidate models has increased in order to develop higher performance products. In order to reduce the number of these simulations, simulations are performed while deforming the model shape and the computational grid in shape optimization using flow simulations [1]. In this method, the number of flow simulations for shape optimization can be reduced, and the optimum shape can be determined in the flow simulation because results for many models can be obtained in one simulation. However, it is necessary to deform the computational grid each time, and it is difficult to apply to a model with complicated shape. In addition, the simulation on the boundary fitted grid can be expected to have high computational accuracy, however, the computational efficiency is inferior to the simulation on the Cartesian grid. In this paper, in order to realize flow simulations for shape optimization with lower cost and versatility, a method is proposed to perform simulation while deforming a model on the Cartesian grid that does not require grid regeneration or deformation. We call this method the morphing simulation method.

In order to apply the present method to models with various shapes on the Cartesian grid, the seamless immersed boundary method (SIBM) [2], which is an improved method of the immersed boundary method (IBM) [3] is used. In the IBM, additional force terms are added to the momentum equations to satisfy the velocity conditions on the virtual boundary points where the computational grid and the boundary of the object intersect. In order to apply the IBM to an object with arbitrary shape, it is only necessary to know the position of the virtual boundary on the grid. Therefore, the IBM can be easily applied to an object with a complicated shape. As for the estimation of the additional forcing term, there are mainly two methods, that is, the feedback [4,5] and direct [6] forcing term estimations. Generally, the direct forcing term estimation is adopted because of the simplicity of the algorithm. However, the conventional IBM with the direct forcing term estimation generates the unphysical pressure oscillations near the virtual boundary because of the pressure jump between inside and outside of the virtual boundary. The SIBM was proposed in order to remove these unphysical pressure oscillations. In the past study, the SIBM was applied not only to stationary objects but also to moving or scaling objects [7,8]. Therefore, it is possible to use the SIBM in the morphing simulation method proposed in this paper. Normally, when the SIBM is applied to a moving or scaling object, the velocity condition in the estimation of the additional forcing term is determined by the moving velocity of the object. In the present method, the additional forcing term is estimated under the condition that the velocity is

zero even if the object is deformed because the purpose of the present morphing simulation is to obtain simulation results for a stationary object.

In this paper, the morphing simulation by the present method is performed for some models and compared with the conventional static SIBM where simulation is performed for each model and the effectiveness of the present method is discussed.

2 Morphing Numerical Simulation Using Seamless Immersed Boundary Method

2.1 Governing Equations

The governing equations are the continuity equation and the incompressible Navier-Stokes equations. Moreover, the forcing term is added to the Navier-Stokes equation for the SIBM. The non-dimensional continuity equation and incompressible Navier-Stokes equations are written as,

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial u_i}{\partial t} = F_i - \frac{\partial p}{\partial x_i} + G_i, \tag{2}$$

$$F_i = -u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{Re} \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{3}$$

where, Re denotes the Reynolds number defined by $Re = L_0 U_0 / \nu_0$. U_0, L_0 and ν_0 are the reference velocity, the reference length and the kinematic viscosity, respectively. $u_i = (u, v)$ and p are the velocity components and the pressure. G_i in Eq. 2 denotes the additional forcing term for the SIBM. F_i denotes the convective and diffusion terms.

2.2 Numerical Method

The incompressible Navier-Stokes equations (Eq. 2) are solved by the second order finite difference method on the collocated grid arrangement. The convective terms are discretized by the fully conservative finite difference method [9] and is written, for example, as,

$$\begin{aligned} v\frac{\partial u}{\partial y}\bigg|_{I,J} &= \frac{1}{2}\left(v\frac{\partial u}{\partial y}\bigg|_{I,J+\frac{1}{2}} + v\frac{\partial u}{\partial y}\bigg|_{I,J-\frac{1}{2}} \right) \\ &= \frac{1}{2}\left(v_{I,J+\frac{1}{2}} \frac{u_{I,J+1} - u_{I,J}}{\Delta} + v_{I,J-\frac{1}{2}} \frac{u_{I,J} - u_{I,J-1}}{\Delta} \right), \end{aligned} \tag{4}$$

where, v is the y component of velocity I, J are the grid index and Δ is grid spacing. The velocity at the midpoint (for example, $J + \frac{1}{2}$) of the grid is calculated by linear interpolation. The diffusive and pressure terms are discretized

by the conventional second order centered finite difference method. For the time integration, the fractional step approach [12] based on the forward Euler method is applied. For the incompressible Navier-Stokes equations in the SIBM, the fractional step approach can be written by

$$u_i^* = u_i^n + \Delta t F_i^n, \tag{5}$$

$$u_i^{n+1} = u_i^* + \Delta t \left(-\frac{\partial p^n}{\partial x_i} + G_i^n \right), \tag{6}$$

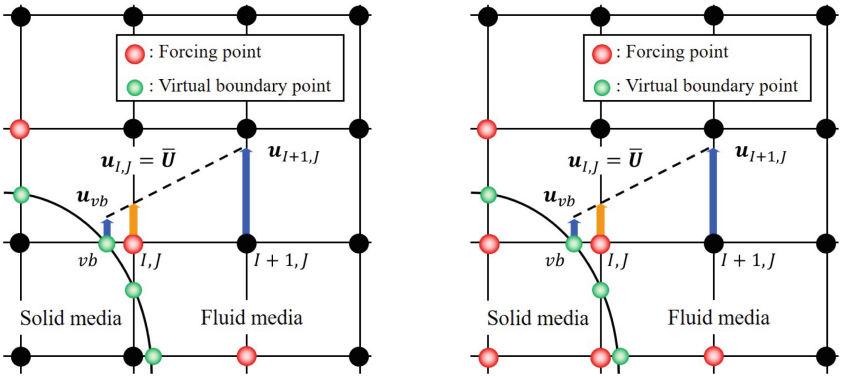
where u_i^* denotes the fractional step velocity and Δt is the time increment. The resulting pressure equation is solved by the successive over-relaxation (SOR) method.

2.3 Seamless Immersed Boundary Method

In order to adopt the SIBM, the additional forcing term in the momentum equations, G_i, should be estimated. In the SIBM, the additional forcing term is estimated by the direct forcing term estimation [2]. The direct forcing term estimation is shown in Fig. 1. We explain in two-dimensions but the extension to three-dimensions is straightforward. For the forward Euler time integration, the forcing term can be determined by

$$G_i^n = -F_i^n + \frac{\partial p^n}{\partial x_i} + \frac{\bar{U}_i^{n+1} - u_i^n}{\Delta t}, \tag{7}$$

where $\bar{U}_i^{n+1}$ denotes the velocity linearly interpolated from the velocity on the near grid point and the velocity (u_{vb}) determined by the velocity condition on the virtual boundary. Namely, the forcing term is specified as the velocity components at next time step satisfy the relation, $u_i^{n+1} = \bar{U}_i^{n+1}$. In the IBM, the grid points added forcing term are restricted near the virtual boundary only (show Fig. 1(a)). In this approach, the non-negligible velocity appears inside the virtual boundary. Also, the pressure distributions near the virtual boundary show the unphysical oscillations because of the pressure jump. In the SIBM, the forcing term is added not only on the grid points near the virtual boundary but also in the region inside the virtual boundary shown in Fig. 1(b) in order to remove the unphysical oscillations near the virtual boundary. In the region inside the virtual boundary, the forcing term is determined by satisfying the relation, $\bar{U}_i^{n+1} = \bar{U}_b$, where $\bar{U}_b$ is the velocity which satisfies the velocity condition at the grid point. When applying the SIBM to a stationary object, the velocity condition on and inside the virtual boundary is zero velocity. As mentioned above, an algorithm of the SIBM is very simple and can easily be extended to three dimensions. Therefore, it is applied to flow around moving or scaling objects [7,8]. When applying the SIBM to a moving or scaling object, the velocity condition on and inside the virtual boundary are obtained by the moving velocity of the object at that point. Moreover, there are also examples of application to turbulence flow [10,11].



(a) Conventional IBM. (b) Seamless IBM.

Fig. 1. Grid points added forcing terms.

2.4 Morphing Numerical Simulation

The morphing of a model on the Cartesian grid is shown in Fig. 2. In the present method, the object is deformed in turn from the first model to the model that requires simulation results. In the SIBM, only the position of the virtual boundary of the object on the fixed grid is updated even if the object deforms. In the SIBM, the virtual boundary of the object with arbitrary shape is represented by boundary nodes in the two-dimensional simulation as shown in Fig. 2. The boundary between these boundary nodes is approximated by straight lines. By determining the intersection between the boundary and the grid that is the virtual boundary point, SIBM can be applied to an object having an arbitrary shape. In the three-dimensional simulation, the virtual boundary of the object with arbitrary shape is represented by triangular polygons and boundary nodes [7]. In the present morphing simulation, the object is deformed from one model to another model by moving these boundary nodes every time step. Once the position of the boundary nodes at each time step is determined, it is easy to apply the SIBM to the model. In the present method, the boundary nodes for the model before deformation is linearly moved to the position of the boundary nodes for the next model. Therefore, the algorithm in the present method is extremely easy. Normally, in the SIBM for the moving or deforming object, the additional forcing term is determined by the moving velocity of the object or boundary. In the present method, the additional forcing term is estimated under the condition that the velocity is zero even if the object is deformed. It is because the purpose of the present morphing simulation is to obtain simulation results for a stationary each model.

3 Application to Two-Dimensional Model

In this paper, in order to verify the present method, two-dimensional simulations for the flow around an object are performed. In order to obtain drag coefficients

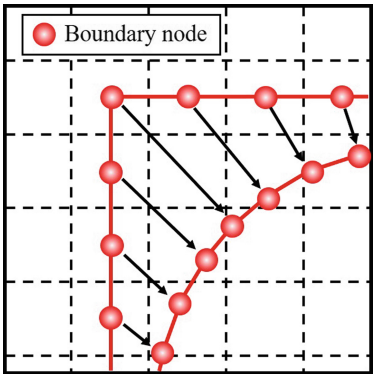


Fig. 2. Morphing of model on the Cartesian grid.

of multiple models, the object is deformed in turn from the initial model to each model in the present morphing simulation. In another case, the morphing has downtime in each model. In this paper, two-dimensional flows around square, circular and elliptic cylinder whose drag coefficients can be compared with the reference results are considered. The computational domain is shown in Fig. 3. In this simulation, a model is set as shown in Fig. 4 and the model is deformed in order from 1 to 5. Each process indicates morphing processes. For example, the model is deformed from 1 to 2 in the process 1, and the model is deformed from 3 to 2 in the process 3. In the present morphing simulation, firstly, the static SIBM simulation is performed for the model 1 and then the processes 1 to 4 are performed in the morphing simulation. This deformation may be larger than the deformation in general shape optimization. In each model, the length of the side of the square cylinder, the diameter of the circular cylinder, and the length of the major axis of the elliptic cylinder are the reference length $L = 1$. The length of the minor axis of the elliptic cylinder is 0.5. As a result, the processes 1 and 2 are scaling down the model and the processes 3 and 4 are scaling up the model. In addition, the processes 1 and 4 are two-dimensional deformations and the processes 2 and 3 are one-dimensional deformations. As for the computational conditions, the impulsive start determined by the uniform flow ($u = 1$, $v = 0$) is adopted. On the inflow boundary, the velocity is fixed by the uniform flow and the pressure is imposed by the Neumann condition obtained by the normal momentum equation. On the outflow and side boundaries (right, top and bottom boundaries), the velocity is extrapolated from the inner points and the pressure is obtained by the Sommerfeld radiation condition [13]. On the virtual boundary and inside the boundary, the velocity condition is the velocity is zero. The Reynolds number is set as $Re = 40$. The flow around each model is steady flow under this Reynolds number. In order to reduce the number of grid points, the hierarchical Cartesian grid with level 4 is introduced. The grid resolution near the model is $\Delta = 1/80$. The number of boundary nodes in each model is 400 and the distance between the nodes in the case of the square cylinder

(models 1, 5) which is the largest model is smaller than the grid spacing. In addition, the boundary node also exists at each vertex of the square cylinder as shown in Fig. 2.

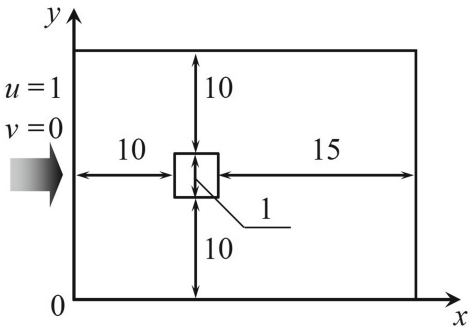


Fig. 3. Computational domain.

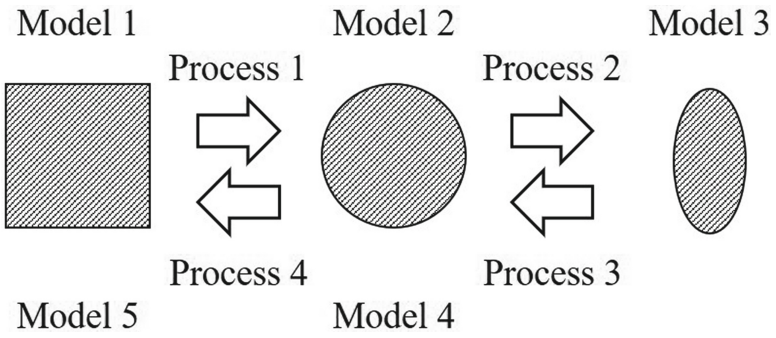


Fig. 4. Configuration of model morphing.

Firstly, in order to obtain the reference results for each model, the conventional static simulations by SIBM without morphing are performed under the above conditions. In Table 1, the drag coefficient is shown with the reference results [14–16]. In this paper, the drag coefficient is estimated by

$$C_D = \frac{-2\int_O (G_x - u_j \frac{\partial u_i}{\partial x_j} - \frac{\partial u_i}{\partial t})ds}{\rho_0 U_0^2 L}, \tag{8}$$

where O denotes the region to which the forcing term is added in the SIBM. ρ_0 and U_0 denote the reference density and velocity of the flow. The drag coefficient by the conventional static SIBM is in good agreement with the reference result in each model. Therefore, these drag coefficients are used as reference results for verifying the present morphing simulation method.

Table 1. Drag coefficient of each model in static SIBM.

	Square cylinder	Circular cylinder	Elliptic cylinder
Static SIBM (Presents)	1.728	1.568	1.631
Sen et al. [14]	1.787	–	–
Dennis et al. [15]	–	1.522	–
Sen et al. [16]	–	–	1.567

In this paper, the morphing simulation is performed under some deformation speeds. The deformation speed is set by the non-dimensional time for each process in Fig. 4. In the present simulation, there is no difference in deformation time between processes. Then, simulations are performed in the case of non-dimensional time is 1, 2, 4, 8 and 16 for the processes (Case 1 to 5). In other words, the deformation speed is slower in Case 5 than in Case 1. In addition, in order to investigate the possibility that the deformation time can be set shorter, a simulation is performed in which downtime of deformation is set after the deformation to each model. In this simulation, the deformation time is the same 1 as Case 1, and the downtime of deformation is 1 (Case 6). That is, the total time for each process in Case 6 is shorter than Case 3 to 5. The above conditions are summarized in Table 2.

Table 2. Non-dimensional time for a process in each condition.

	Total time for a process	
	Deformation time	Downtime
Case 1	1	–
Case 2	2	–
Case 3	4	–
Case 4	8	–
Case 5	16	–
Case 6	1	1

Figures 5, 6, 7 and 8 show the pressure contours of each model. Note that the pressure contours of the models 2 and 4 by the static SIBM is same. In all cases, the pressure contours obtained by the present method are similar to those obtained by the static SIBM. In particular, those in Cases 5 and 6 are in good agreement with those in the static SIBM.

Figure 9 shows time histories of the comparison of the drag coefficients at each deformation speed. The horizontal axis shows the non-dimensional time converted into the model number. For example, the model number is 2 when the horizontal axis is 2 and the model is being deformed from 2 to 3 (process 2)

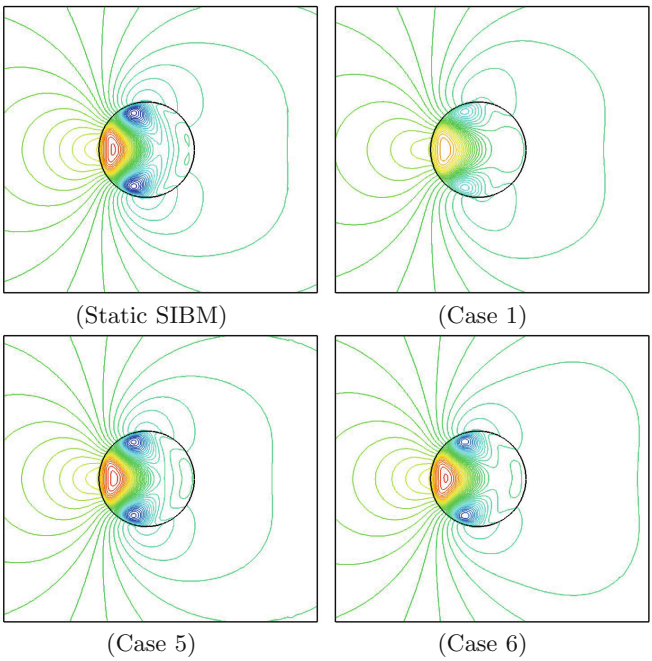


Fig. 5. Pressure contours of model 2.

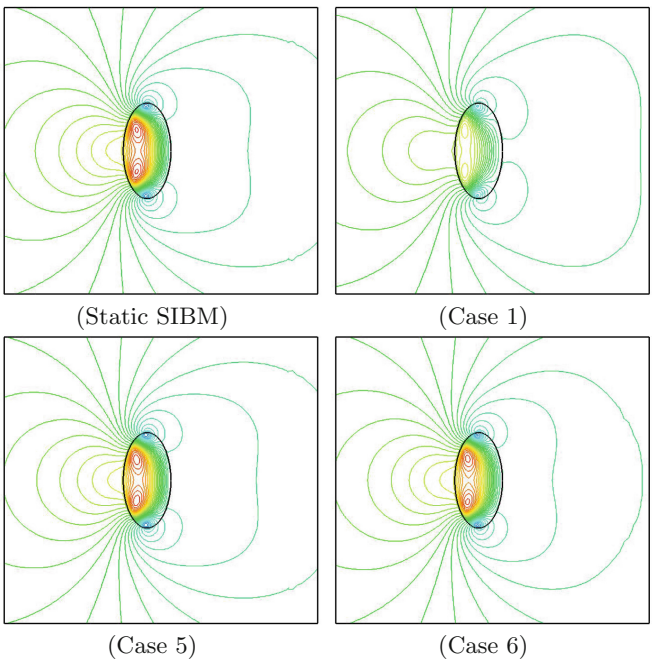


Fig. 6. Pressure contours of model 3.

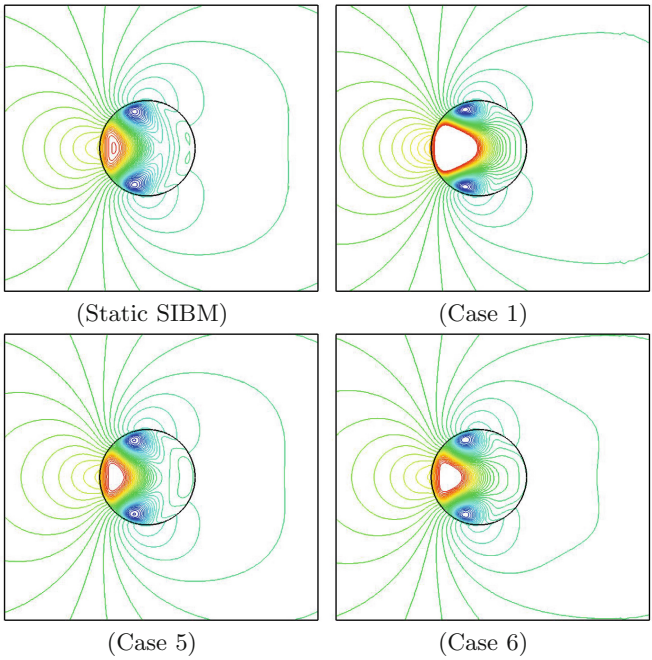


Fig. 7. Pressure contours of model 4.

when the horizontal axis is between 2 and 3. Table 3 shows the comparison of the drag coefficients of each model at each deformation speed. In Case 1–5, the drag coefficients of each model are closer to the reference values as the deformation speed is slower. When the model is deformed, oscillations of the drag coefficient due to the virtual boundary moving across the grid are observed. These oscillations are remarkable in the case of two-dimensional deformation. In particular, the results for each model in Case 4 and 5 are close to the reference results. Therefore, it was shown that the results equivalent to the results by the conventional static SIBM can be obtained by the present morphing method. In Case 6, the drag coefficients are shown different from the reference value just like Case 1 immediately after deformation, however, the values become to the same level as in Cases 4 and 5 in the deformation downtime. Therefore, it was shown that the present morphing simulation with downtime can set the deformation speed faster than the present morphing simulation without downtime.

Table 4 shows the comparison of the rate of computational time of each morphing process at each deformation speed. Each computational time is based on the computational time of the process 1 in Case 1. It can be observed that the longer the non-dimensional time for deformation is, the longer the computational time is. The computational time of Case 6 is almost the same as Case 2 where the non-dimensional time of each process is the same. According to the above results, it was shown that the present method can be accelerated by adding in the morphing process the downtime.

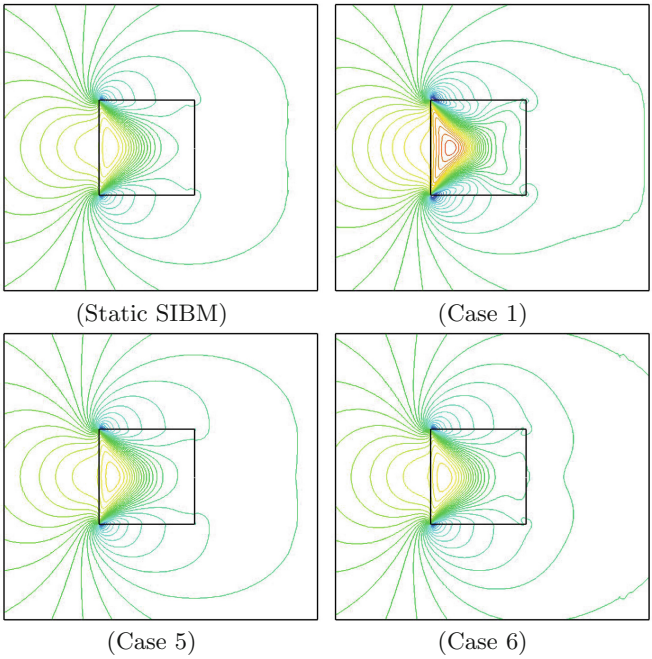
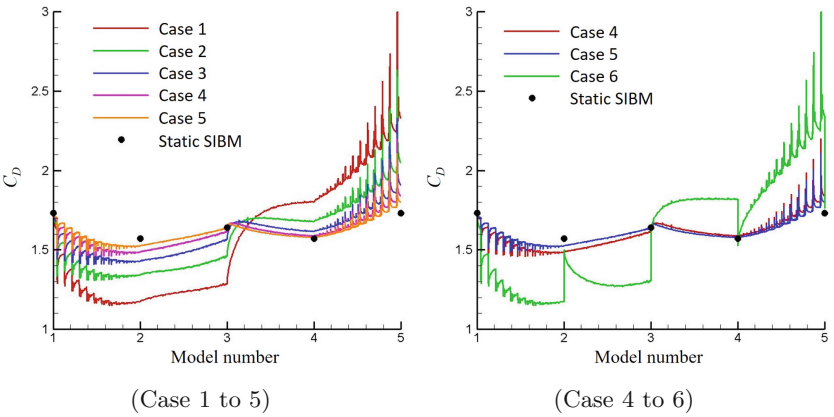


Fig. 8. Pressure contours of model 5.



(Case 1 to 5) (Case 4 to 6)

Fig. 9. Comparison of drag coefficients at each deformation speed.

Table 3. Drag coefficient of each model.

	Model 2	Model 3	Model 4	Model 5
Case 1	1.175	1.288	1.802	2.327
Case 2	1.342	1.461	1.677	2.048
Case 3	1.428	1.567	1.615	1.908
Case 4	1.487	1.617	1.587	1.840
Case 5	1.525	1.638	1.577	1.801
Case 6	1.500	1.607	1.527	1.760
Static SIBM	1.568	1.631	1.568	1.728

Table 4. Rate of computational time of each morphing process.

	Process 1	Process 2	Process 3	Process 4
Case 1	1.00	0.67	1.04	1.88
Case 2	1.46	1.19	1.54	2.19
Case 3	2.02	2.03	2.22	2.55
Case 4	2.77	3.05	3.13	3.20
Case 5	3.68	4.50	4.49	4.03
Case 6	1.25	1.01	1.46	2.10

4 Conclusions

In this paper, we proposed the morphing simulation method on the Cartesian grid in order to realize flow simulations for shape optimization with lower cost and versatility. By using SIBM that is the Cartesian grid approach, the present method could be applied very easily to an object with arbitrary shape. In order to verify the present method, the two-dimensional simulations for the flow around an object were performed. In order to obtain drag coefficients of multiple models, the object was deformed in turn from the initial model to each model in the present morphing simulation. By using the present method, the drag coefficients for some models could be obtained by one simulation. These drag coefficients became closer to the reference values by decreasing the deformation speed of the model. Furthermore, by setting the downtime after the deformation, drag coefficients close to the reference values were obtained even when the deformation speed was high. Therefore, it can be concluded that the flow simulation for shape optimization can be performed very easily and the number of times of flow simulation for many models can be significantly reduced by using the present morphing simulation method.

References

1. Hino, T.: Shape optimization of practical ship hull forms using Navier-Stokes analysis. In: Proceedings of the 7th International Conference on Numerical Ship Hydro (1999)
2. Nishida, H., Sasao, K.: Incompressible flow simulations using virtual boundary method with new direct forcing terms estimation. In: Deconinck, H., Dick, E. (eds.) Computational Fluid Dynamics 2006, pp. 371–376. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92779-2_57
3. Peskin, C.S., McQueen, D.M.: A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid. J. Comput. Phys. **81**(2), 372–405 (1989)
4. Goldstein, D., Handler, R., Sirovich, L.: Modeling a no-slip flow boundary with an external force field. J. Comput. Phys. **105**(2), 354–366 (1993)
5. Saiki, E.M., Biringen, S.: Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. J. Comput. Phys. **123**(2), 450–465 (1996)
6. Fadlun, E.A., Verzicco, R., Orlandi, P., Mohd-Yosof, J.: Combined immersed-boundary finite-difference methods for three-dimensional complex simulations. J. Comput. Phys. **161**(1), 35–60 (2000)
7. Nishida, H., Tajiri, K.: Numerical simulation of incompressible flows around a fish model at low reynolds number using seamless virtual boundary method. J. Fluid Sci. Technol. **4**(3), 500–511 (2009)
8. Nishida, H., Tajiri, K., Tanaka, M.: Seamless immersed boundary method for flow around a scaled object. In: Proceedings 12th Asian Computational Fluid Dynamics Conference, pp. 1–9 (2018)
9. Morinishi, Y., Lund, T.S., Vasilyev, O.V., Moin, P.: Fully conservative higher order finite difference schemes for incompressible flow. J. Comput. Phys. **143**(1), 90–124 (1998)
10. Tajiri, K., Nishida, H., Tanaka, M.: Large eddy simulation of turbulent flow using seamless immersed boundary method. In: Proceedings of the 8th International Conference on Computational Fluid Dynamics, ICCFD8-197, pp. 1–13 (2014)
11. Tajiri, K., Nishida, H., Tanaka, M.: Property of seamless immersed boundary method for large eddy simulation of incompressible turbulent flows. J. Fluid Sci. Technol. **9**(2), 1–8 (2014)
12. Rhie, C.M., Chow, W.L.: Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J. **21**(11), 1525–1532 (1983)
13. Kawakami, K., Nishida, H., Satofuka, N.: An open boundary condition for the numerical analysis of unsteady incompressible flow using the vorticity-streamfunction formulation. Trans. Jpn. Soc. Mech. Eng. Ser. B **60**(574), 1891–1896 (1994). (Japanese)
14. Sen, S., Mittal, S., Biswas, G.: Flow past a square cylinder at low Reynolds numbers. Int. J. Numer. Methods Fluids **67**(9), 1160–1174 (2011)
15. Dennis, S.C.R., Chang, G.Z.: Numerical solutions for steady flow past a circular cylinder at Reynolds num-bers up to 100. J. Fluid Mech. **42**(3), 471–489 (1970)
16. Sen, S., Mittal, S., Biswas, G.: Steady separated flow past elliptic cylinders using a stabilized finite-element method. Comput. Model. Eng. Sci. **86**(1), 1–26 (2012)