

Research Article

A Comparative Study on Evaluation Methods of Fluid Forces on Cartesian Grids

Taku Nonomura^{1,2} and Junya Onishi³

¹*Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, Yoshinodai 3-1-1 1816, Sagamihara, Kanagawa, Japan*

²*Department of Aerospace Engineering, Tohoku University, Sendai, Miyagi, Japan*

³*Institute of Industrial Science, The University of Tokyo, 7-1-26, Minatojima-minimi-machi, Chuo-ku, Kobe, Hyogo, Japan*

Correspondence should be addressed to Taku Nonomura; nonomura@aero.mech.tohoku.ac.jp

Received 7 April 2017; Revised 31 May 2017; Accepted 14 June 2017; Published 19 July 2017

Academic Editor: Rahmat Ellahi

Copyright © 2017 Taku Nonomura and Junya Onishi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate the accuracy and the computational efficiency of the numerical schemes for evaluating fluid forces in Cartesian grid systems. A comparison is made between two different types of schemes, namely, polygon-based methods and mesh-based methods, which differ in the discretization of the surface of the object. The present assessment is intended to investigate the effects of the Reynolds number, the object motion, and the complexity of the object surface. The results show that the mesh-based methods work as well as the polygon-based methods, even if the object surface is discretized in a staircase manner. In addition, the results also show that the accuracy of the mesh-based methods is strongly dependent on the evaluation of shear stresses, and thus they must be evaluated by using a reliable method, such as the ghost-cell or ghost-fluid method.

1. Introduction

In recent years, Cartesian grids have been widely used in computational fluid dynamics applications. This is not only because of the simplicity of data structure and algorithms, but also because of the fast, automatic, and robust grid generation for complex geometries. The latter reason is particularly important for industrial applications, since grid generation is one of the most time-consuming tasks. Cartesian grids have potential to overcome this difficulty.

However, flow simulation using Cartesian grids still has some issues to be addressed before being put into practice. A key issue is the development of accurate methods for evaluating physical quantities and fluxes at the surface of solid objects. This issue comes from the fact that Cartesian grids do not necessarily align with the surface of the objects. As a result, there arises some arbitrariness in the location and the orientation of the object surfaces, and thus the evaluation of physical quantities and fluxes at the object surface is highly dependent on the numerical schemes used. To improve the

numerical evaluation described above, many efforts have been already devoted since the pioneering work by Peskin [1]. An extensive review of the methods developed in the literatures is given by Mittal and Iaccarino [2].

Despite these efforts, however, there has been no established method that can be applied to any problems. This is partly due to the lack of the systematic investigation on the accuracy of the numerical schemes used in these methods. In this study, we assess and compare the accuracy and the computational efficiency of the numerical schemes for evaluating physical quantities and fluxes in Cartesian grid systems. In particular, we focus on the evaluation of the local fluid stresses at the surface of an object and on the evaluation of the total fluid forces acting on the object body, as the integration of the local stresses.

The fluid force acting on a solid body is evaluated by integrating the fluid stresses over the body surface:

$$F_j = \int_{\Gamma_s} \sigma_{ij} n_i dS, \quad (1)$$

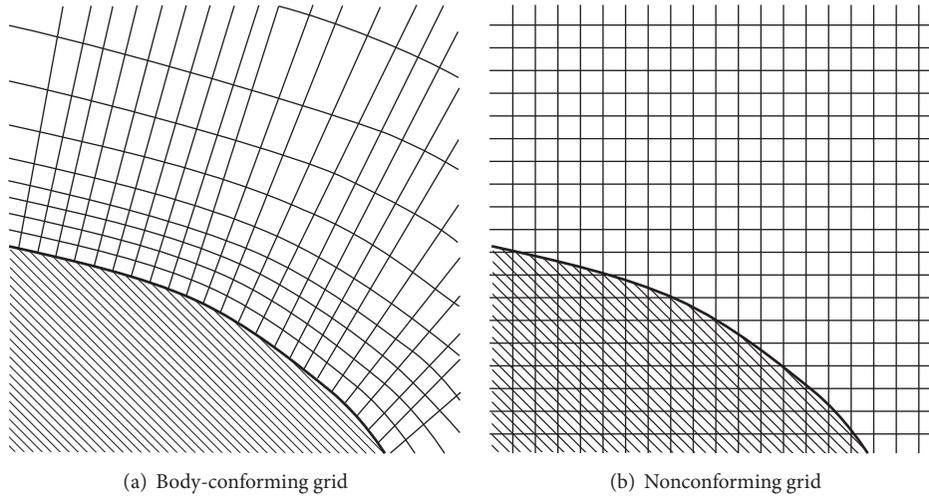


FIGURE 1: Examples of body-conforming and nonconforming grids.

where σ_{ij} is the stress tensor and n_j is the unit outward normal of the surface element. The stress tensor σ_{ij} of Newtonian fluids is given by

$$\sigma_{ij} = -p\delta_{ij} + \mu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right), \quad (2)$$

where p is the pressure, u_i is the velocity, and μ is the viscosity.

In numerical computations, the surface integral defined by (1) is evaluated in two steps:

- (i) discretization of the surface,
- (ii) estimation of the physical quantities at the surface.

The numerical procedures to conduct these steps depend on the grid system used in the simulation. In body-conforming curvilinear grid systems, shown in Figure 1(a), both steps can be implemented on the specified grids in a straightforward manner. However, in nonconforming grid systems, shown in Figure 1(b), some special treatment is required since the grids do not exactly match the body surface. To the best of our knowledge, there are two different ways which have been adopted in nonconforming grid systems, namely, the control volume method and the direct calculation.

The control volume method introduces a virtual volume which encompasses the body and estimates the fluid forces from the net momentum flux through the surface of this virtual volume. This method was firstly applied to steady forces by Lai and Peskin [3], and later, it was extended to unsteady forces by Balaras [4] and to moving body problems by Shen et al. [5]. The control volume method has an advantage that it is unnecessary to directly handle the complex geometries expressed by the immersed boundaries. On the other hand, it has a severe disadvantage that local stresses cannot be computed; that is, only the total force acting on the body can be evaluated. Moreover, the contributions from pressure and viscous forces cannot be evaluated separately.

In the direct calculation, the body surface is usually discretized by a set of polygons, and the local stresses on these

surface elements are estimated. Then, the local stresses are integrated over the body surfaces to obtain the total force, as in body-conforming grid systems. However, unlike the cases of body-conforming grid systems, the estimation of the flow variables on the surface elements is not straightforward and requires elaborate interpolation and/or extrapolation methods in order to accurately take into account the location (and orientation if necessary) of the surface. Moreover, the computational cost of the stress integration may become prohibitive when the body shape is complex, since it depends on the number of the polygons used to express the body surface. Such a dependency is undesirable, in particular, in engineering applications. For example, the number of the polygons reaches up to the order of 10^7 for an aerodynamic simulation using a detailed car model with both exterior and interior parts [6].

In this study, we discuss numerical schemes that are suitable for evaluating fluid forces in nonconforming grid systems. In particular, we propose a simple and computationally efficient method for calculating the surface integral in (1). The proposed method is a kind of direct calculation described above. Therefore, it has inherited an advantage over the control volume method in that the contributions from pressure and viscous forces can be evaluated locally and separately. Moreover, in the present method, the fluid stresses are integrated over the mesh faces (staircase surfaces) in the grid system used, not over the discretized surface elements of the original body. Therefore, the undesirable dependency on quality or numbers of surface polygons can be removed. Hereafter, the present method is referred to as a “mesh-based scheme (MS),” whereas the method based on the integration over surface polygons is called a “polygon-based scheme (PS).” We present a comparative study of the accuracy of these methods.

It is worth noting here that, in both mesh-based and polygon-based schemes, the physical quantities at the discretized surfaces must be estimated by interpolation or extrapolation, and the accuracy of this estimation would

influence the final results of the fluid forces. For the PS, we have employed the ghost-cell approach described by Mittal et al. [7]. For the MSs, we have tested three approaches that are often used in the studies of immersed boundary methods: a staircase approach, a ghost-cell approach, and a ghost-fluid approach, the details of which are described in Section 2.

The rest of the paper is organized as follows. Section 2 describes the details of the computational procedures of a polygon-based scheme and three mesh-based schemes. The computational grid, the discretization of the body surface, and the extrapolation method used in these schemes are fully discussed. In Section 3, both polygon-based and mesh-based schemes are applied to evaluate the drag force acting on a sphere in Stokes flow. The obtained results are compared with the analytical solution. The grid convergence test is also performed to investigate the numerical accuracy of the schemes. Stokes flow is used here because analytical solutions are available not only for the fluid forces but also for the flow fields, that is, pressure and velocity fields. By using the analytical solutions of the flow fields, the error associated with the force evaluation schemes can be assessed separately from the error in the computation of the flow fields themselves. In contrast, in higher Reynolds number flows, the flow fields must be numerically computed. Therefore, these errors are hardly evaluated separately. In Section 4, the extension to the moving body problem and the computational efficiency of the force evaluation schemes are discussed. In addition, to show the applicability of the present mesh-based scheme in higher Reynolds number flows, the results for the case of the Reynolds number of 100 are presented in Section 5. In Section 6, the discussion on the applicability to the complex body problem is conducted. Finally, the last section summarizes this study.

2. Numerical Methods

Herein, Section 2.1 describes the computational grid and defines the cell types. The polygon-based force evaluation scheme, which seems to be commonly used in the immersed boundary method, and the mesh-based force evaluation scheme proposed in this paper are described in Sections 2.2 and 2.3, respectively.

2.1. Computational Grids and Definition of Cell Type. The grid system used in this work is illustrated in Figure 2. In this figure, the entire domain is divided into equally sized cubic cells. These cells are classified into two groups according to their locations:

- (i) fluid cells (cells whose center is in the fluid region),
- (ii) solid cells (cells whose center is in the solid region).

Within these categories, special cells close to the interface are defined as follows:

- (i) boundary cells (fluid cells adjacent to at least one solid cell).

The flow properties used to evaluate (1), namely, the pressure and the velocity, are assumed to be available only at the

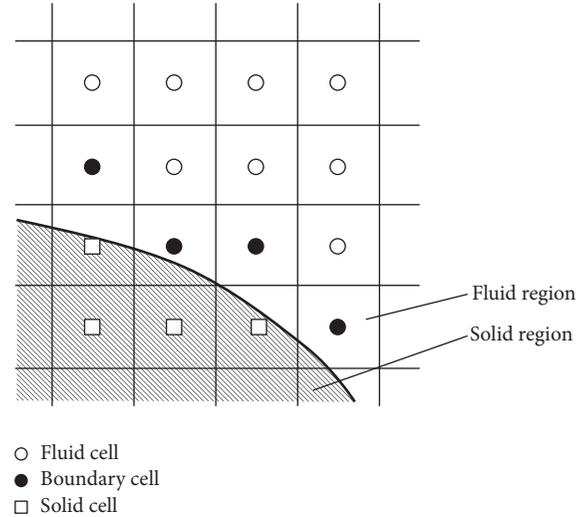


FIGURE 2: Classification of the computational cells.

center of the fluid cells, including the boundary cells. When the values at the solid cells are needed, they must be interpolated or extrapolated by some methods considering a body shape. In the following, we call such solid cells ghost cells.

Note that, although we use collocated layout of variables throughout this study, the extension to staggered layout of variables is trivial. The accuracy of mesh-face integration, which is key idea of this paper, does not seem to be affected by the layout of variables. Note also that the above cell classification is not trivial, especially when the solid region has a complicated shape. This issue, however, is beyond the scope of the present study and is not further discussed here.

2.2. Polygon-Based Scheme. The polygon-based scheme (PS) appears to be most commonly implemented in the immersed boundary method [8]. This scheme divides three-dimensional body surfaces into sets of polygons or two-dimensional surfaces into series of lines. Hereafter, to avoid confusion, these surface elements will be referred to as “facets.” A schematic of this approach is shown in Figure 3(a).

In PS, the surface integral defined by (1) is conveniently evaluated by summing over the facets:

$$F_j = \sum_{\text{facet}} (\sigma_{ij} n_i \Delta S)_{\text{facet}}, \quad (3)$$

where ΔS is the surface area of the facet, and the normal stresses are summed over all facets. The remaining task is to estimate the bracketed quantities at the center of each facet, which requires the geometric properties (the surface normal and the area of the facet) and the physical properties (the pressure and the velocity). The former quantities can be computed from the coordinates of the polygon vertexes. In this paper, the latter are estimated by an approach using the normal, which is described in the following.

As shown in Figure 3(a), a normal probe is extended from the center of the facet (indicated by I), and two markers

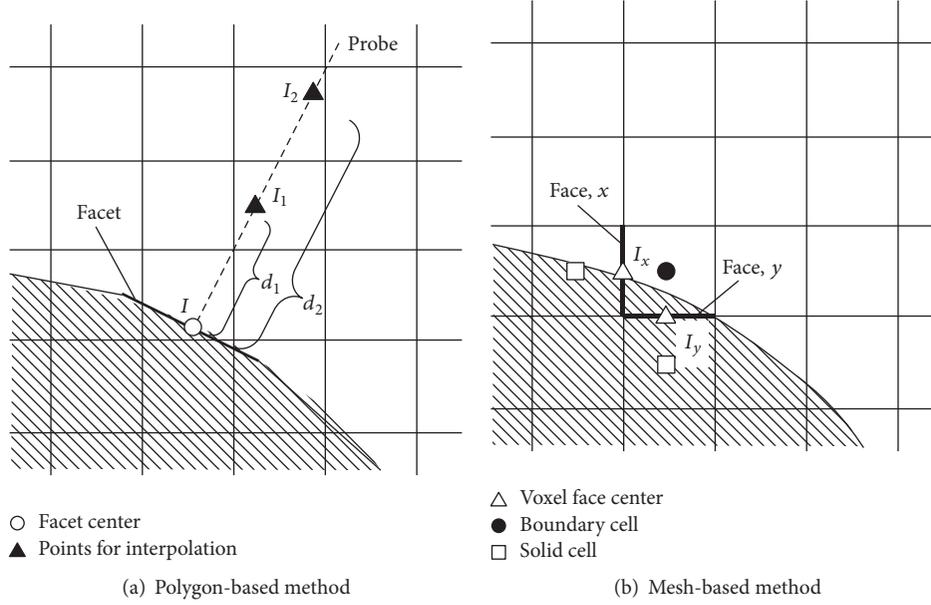


FIGURE 3: Polygon and mesh-based methods.

(indicated by I_1 and I_2) are placed at distances d_1 and d_2 , respectively, along the normal probe. These markers are used as reference points for interpolating the value at the center of the facet. The pressure field assumes a linear distribution along the probe:

$$p(d) = C_1 d + C_0. \quad (4)$$

The coefficients C_1 and C_0 are determined such that

$$p(d_1) = p_1, \quad (5)$$

$$p(d_2) = p_2,$$

where p_1 and p_2 are the values at the reference points I_1 and I_2 , respectively. From this linear extrapolation, the value at the point I is obtained as

$$p_I = p(0) = p_1 - \frac{d_1}{d_2 - d_1} (p_2 - p_1). \quad (6)$$

The velocity components are computed from a quadratic function. For example, the x -component of the velocity is fitted to

$$u(d) = C_2 d^2 + C_1 d + C_0, \quad (7)$$

where the coefficients are determined by the values at both reference points and the center of the facet:

$$u(d_1) = u_1, \quad (8)$$

$$u(d_2) = u_2,$$

$$u(0) = u_B,$$

where u_1 and u_2 are the velocities at the reference points I_1 and I_2 , respectively, and u_B is the velocity at the center of the

facet. Note that u_B is the velocity of the object because no-slip and no-penetration conditions are imposed. Differentiating this obtained function yields the components of the velocity gradient tensor at the point I

$$\begin{aligned} \frac{\partial u}{\partial n} \Big|_I &= \frac{u_1 d_2^2 - u_2 d_1^2}{d_1 d_2 (d_2 - d_1)}, \\ \frac{\partial u}{\partial s} \Big|_I &= 0, \\ \frac{\partial u}{\partial t} \Big|_I &= 0, \end{aligned} \quad (9)$$

where the coordinates n , s , and t are defined along the normal and the tangential directions at the facet, respectively. The other components are computed similarly.

In the above extrapolation, the values at the reference points should be estimated in advance. In this paper, the reference values are computed by trilinear interpolation from the quantities in the surrounding fluid cells. If the reference points are not carefully chosen, the interpolation involves a quantity in a solid cell that must also be extrapolated. To avoid such recursive interpolations and extrapolations, the distances to the reference points are simply selected to be long enough. In this paper, d_1 and d_2 are set to $1.75\Delta x$ and $3.5\Delta x$, respectively. Note that the value $1.75\Delta x$ comes from the possible largest distance between neighboring cell centers, namely, $\sqrt{3}\Delta x$ for a three-dimensional case, and the value $3.5\Delta x$ is twice as long as it. This method is called a simplified ghost-cell method in this paper.

2.3. Mesh-Based Scheme. The PS described above is simple in algorithm, but it seems to be prohibitive for practical applications where complex geometries must be handled. This is

because the PS requires operations of $O(N_S)$, where N_S is the number of the polygons used to represent the geometries. For example, N_S is of the order of 10^7 for a detailed car model with both exterior and interior parts [6]. This problem becomes significant, in particular, on distributed memory platform, since the operations are intensive only in the small area of the computational domain and thus the work load is not balanced well.

As an alternative to the PS, we propose mesh-based schemes (MSs), in which the body is simplified by mesh faces between boundary and solid cells, and the surface integral defined by (1) is discretized over the mesh faces. A schematic illustration of this approach is shown in Figure 3(b).

In MSs, the surface integral defined by (1) is decomposed into its directional components, each approximated from the mesh face in its corresponding direction:

$$\begin{aligned} F_x &= \int \sigma_{xx} n_x dS + \int \sigma_{yx} n_y dS + \int \sigma_{zx} n_z dS \\ &\approx \sum_{\text{face},x} (\sigma_{xx} n_x \Delta S)_{\text{face},x} + \sum_{\text{face},y} (\sigma_{yx} n_y \Delta S)_{\text{face},y} \\ &\quad + \sum_{\text{face},z} (\sigma_{zx} n_z \Delta S)_{\text{face},z}. \end{aligned} \quad (10)$$

Here, since the directional components can be independently summed, the surface normal and the area are easily evaluated from the geometrical characteristics of the mesh face. For example, $n_x = \pm 1$ and $\Delta S = \Delta y \Delta z$ are held for the mesh faces in the x -direction. The surface normal and the areas of mesh faces in other orientations can be similarly evaluated. Note also that the summations involve only the faces of the boundary cells, which simplifies the computational implementation.

The stress at the center point of the mesh face (I_x and I_y in Figure 3(b)) can be computed in several ways. In the following, three methods, which seem to be used widely in the studies of immersed boundary methods, are discussed.

2.3.1. MS1: A Staircase Approach. In MS1, the pressure is simply extrapolated using the cell-centered value:

$$\begin{aligned} p_{I,x} &= p_{b.c.}, \\ p_{I,y} &= p_{b.c.}, \end{aligned} \quad (11)$$

where $p_{I,x}$ and $p_{I,y}$ are, respectively, the pressures at I_x and I_y and $p_{b.c.}$ is the pressure at the related boundary cell. To approximate the shear stress, the velocities at I_x and I_y are assumed to correspond to that of the object $u_I = 0$:

$$\begin{aligned} \left. \frac{\partial u}{\partial x} \right|_{I,x} &= \frac{u_{i,j} - u_I}{\Delta x/2}, \\ \left. \frac{\partial u}{\partial y} \right|_{I,x} &= \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta x}, \end{aligned} \quad (12)$$

where the computational stencil used for these derivatives is depicted in Figure 4.

Although the above staircase approach may be used less often nowadays, it is introduced in order to show its inaccuracy and to clearly demonstrate the important aspects of the other schemes introduced later.

2.3.2. MS2: A Ghost-Cell Approach. In MS2, the assumed pressure at I_x is the average of the pressures at the centers of the adjacent boundary cell and the solid cell.

$$p_{I,x} = \frac{p_{i,j} + p_{i-1,j}}{2}. \quad (13)$$

The shear stress is approximated by a second-order central difference scheme. The x - and y -derivatives of the x component of the velocity at the point I_x are estimated as follows:

$$\begin{aligned} \left. \frac{\partial u}{\partial x} \right|_{I,x} &= \frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \\ \left. \frac{\partial u}{\partial y} \right|_{I,x} &= \frac{1}{2} \left[\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta x} + \frac{u_{i-1,j+1} - u_{i-1,j-1}}{2\Delta x} \right]. \end{aligned} \quad (14)$$

The computational stencil used to compute these derivatives is also depicted in Figure 4(a).

Note that the computation of both x - and y -derivatives requires quantities inside the solid region, in other words, in the ghost cells. For MS2, these quantities are estimated by the simplified version (avoiding recursive interpolation and extrapolation) of the ghost-cell method [7]. Concretely, the quantities in solid cells are computed by the fitting functions (see (6) and (7)) constructed in the normal probe approach illustrated in Figure 4(b):

$$\begin{aligned} p_s &= p(-d_0), \\ u_s &= u(-d_0), \end{aligned} \quad (15)$$

where d_0 is the distance between the solid cell and the closest point on the body surface.

2.3.3. MS3: A Ghost-Fluid Approach. In MS3, the ghost-fluid method described by Gibou et al. [9] is applied to estimate the quantities in the ghost cells.

For example, in order to estimate a quantity at the solid cell $(i-1, j)$ in Figure 4(a), a quadratic function along the x -axis is introduced and is fitted to the two points at the fluid cells (i, j) and $(i+1, j)$ and the interface point:

$$f(x) = C_2 x^2 + C_1 x + C_0, \quad (16)$$

where f denotes the pressure or a component of the flow velocity, x is the coordinate of the x -axis, and C_i ($i = 0, 1, 2$) is the fitting coefficients to be determined. Note that the interface point is defined as the intersection between the solid surface and the line segment connecting (i, j) and $(i-1, j)$. Here, the origin is set to (i, j) , and the distance from the origin to the interface point is assumed to be $\theta \Delta x$, without loss of generality. For the x -component of the flow velocity, three coefficients in (16) are uniquely determined by solving the following system of equations:

$$\begin{aligned} u(0) &= u_{i,j}, \\ u(\Delta x) &= u_{i+1,j}, \\ u(-\theta \Delta x) &= u_B, \end{aligned} \quad (17)$$

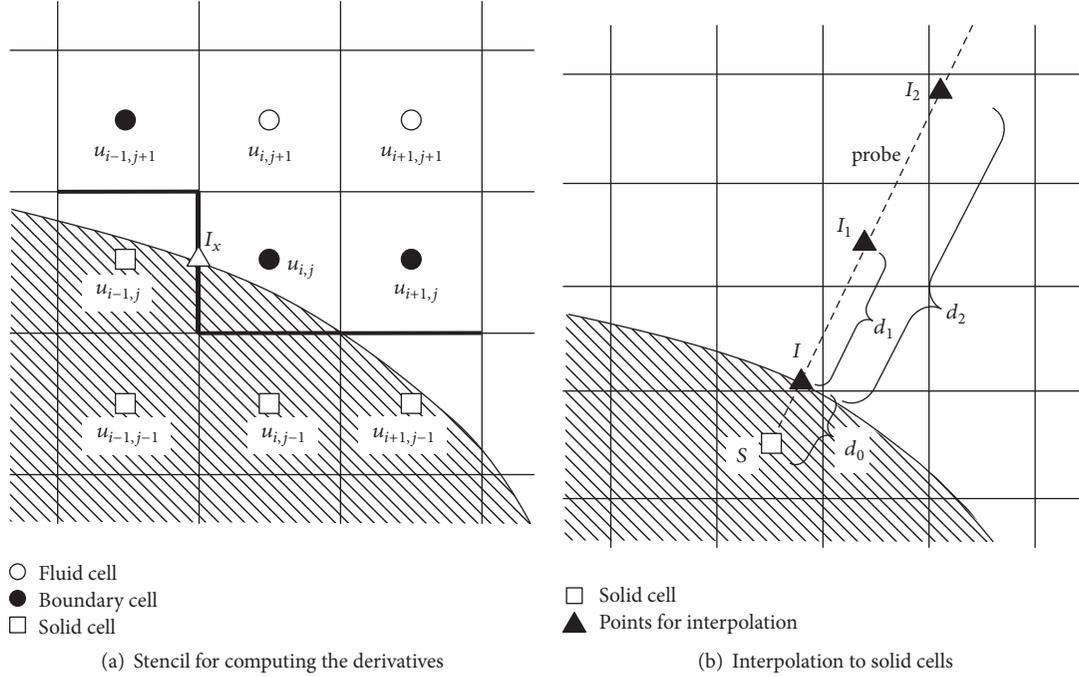


FIGURE 4: Stencils for stress evaluation and extrapolation to the solid cell.

where u_B is the prescribed boundary value for u . As a result, the target value at the solid cell is estimated as follows:

$$u_{i-1,j} = u(-\Delta x). \quad (18)$$

The values for the other components can be calculated in a similar manner. The pressure can be also estimated in a similar manner, but it is necessary to replace one of the conditions by the Neumann boundary condition:

$$\begin{aligned} p(0) &= p_{i,j}, \\ p(\Delta x) &= p_{i+1,j}, \\ \frac{\partial p}{\partial x}(-\theta \Delta x) &= \left. \frac{\partial p}{\partial x} \right|_B, \end{aligned} \quad (19)$$

where $(\partial p / \partial x)_B$ is the x -component of the pressure gradient at the boundary, which is assumed to be 0 in this study.

3. Force Estimation for Stokes Flows

In this section, the accuracies of PS, MS1, MS2, and MS3 are assessed by comparison with analytical solutions of Stokes flow past a sphere.

3.1. Stokes Flow Past a Sphere. Analytical solutions to the pressure and velocity of Stokes flow around a sphere are given by [10]:

$$\begin{aligned} p &= p_\infty - \frac{3}{2} \frac{\mu}{\rho} u_\infty \frac{ax}{r^3}, \\ u &= u_\infty - \frac{1}{4} u_\infty \frac{a}{r} \left(3 + \frac{a^2}{r^2} \right) - \frac{3}{4} u_\infty \frac{ax^2}{r^3} \left(1 - \frac{a^2}{r^2} \right), \end{aligned}$$

$$\begin{aligned} v &= -\frac{3}{4} u_\infty \frac{axy}{r^3} \left(1 - \frac{a^2}{r^2} \right), \\ w &= -\frac{3}{4} u_\infty \frac{axz}{r^3} \left(1 - \frac{a^2}{r^2} \right), \end{aligned} \quad (20)$$

where a is the radius of the sphere, ρ is the mass density, μ is the viscosity, u_∞ is the freestream velocity, p_∞ is the reference pressure, x , y , and z are the Cartesian coordinates, and r is the distance from the origin (where the origin is the center of the sphere).

The fluid is assumed to flow parallel to the x -axis; therefore, the drag force acting on the sphere is the x -component of the force defined in (1). Substituting the analytical solutions (20) into this definition, the pressure drag and the frictional drag are, respectively, obtained as

$$\begin{aligned} C_{Dp} &= \frac{F_x^p}{(1/2) \rho u_\infty^2 A} = \frac{2\pi a \mu u_\infty}{(\rho u_\infty^2 / 2) (\pi a^2)} = \frac{8\mu}{\rho u_\infty D} \\ &= \frac{8}{\text{Re}}, \\ C_{Df} &= \frac{F_x^v}{(1/2) \rho u_\infty^2 A} = \frac{4\pi a \mu u_\infty}{(\rho u_\infty^2 / 2) (\pi a^2)} = \frac{16\mu}{\rho u_\infty D} \\ &= \frac{16}{\text{Re}}, \end{aligned} \quad (21)$$

where C_{Dp} and C_{Df} are the coefficients of the pressure drag and frictional drag, respectively, and $\text{Re} = \rho u_\infty D / \mu$ is the Reynolds number. $D = 2a$ and $A = \pi a^2$ define the diameter and the projected area of the sphere, respectively.

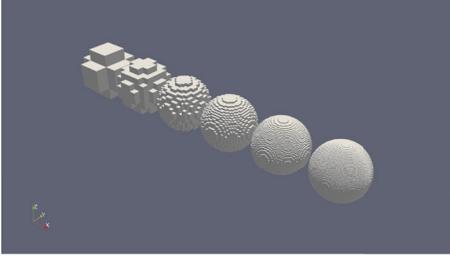


FIGURE 5: Spheres constructed on meshes with different resolutions. From left to right, a sphere of diameter D is constructed on mesh resolved to $\Delta x = a/4, a/8, a/16, a/32, a/64,$ and $a/128$.

3.2. Computational Settings and Data. The drag force is now computed by the numerical methods described in Section 2. To compute the drag forces, a sphere of the diameter D is centralized within a cubic domain of the side length L . The value of L is arbitrary but must be sufficiently large to encompass the sphere. The present study selects $L = 2D$. The computational cells are generated by dividing the cubic domain into $2N$ equally sized cells in each direction. The fineness of the sphere is altered by setting the number of grid points N occupied by the sphere in each direction to $N = 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048$. The solid cells constituting the sphere of grid points less than 128 are illustrated in Figure 5. The PS requires an additional polygon mesh. The force computed on each Cartesian mesh properly converges with that on a spherical equally distributed polygon mesh with $3,200 \times 3,200$ points. Therefore, this polygon mesh is adopted in the remainder of the study. The analytical solutions (20) are assumed to be available only at the centers of the fluid cells, including the boundary cells.

3.3. Results

3.3.1. Total Drag Forces. The forces estimated by PS, MS1, MS2, and MS3 at different mesh resolutions are plotted in Figure 6. Here, the Reynolds number is $Re = 0.01$ for all the cases. Correspondingly, the pressure and friction drags are, respectively, solved as $C_{Dp,exact} = 800$ and $C_{Df,exact} = 1600$, using the analytical solutions. The result shows that the convergence rate for the pressure drag is first order in MS1 and second order in PS and MS2 and that the convergence rate for the friction drag is zeroth order in MS1 and first to second order in PS, MS2, and MS3. These results indicate that the proposed MS2 and MS3 can evaluate the fluid force as accurately as PS, but at much smaller computational cost and using a much simpler computational code. Thus, fluid forces can be simply and efficiently evaluated by MS2 and MS3 in the immersed boundary method. Comparing MS2 and MS3, it interestingly shows that MS2 has fewer errors than MS3. This clarifies that the MS2 based on the ghost-cell method is better for the force estimation than MS1 based on the ghost-fluid method. Apart from the success of MS2 and MS3, MS1 is shown to be unacceptable for practical use since the friction drag is not converged even when a very fine mesh is used. At $\Delta x = 0.06$, sudden decrease in friction drag of PS is observed and this might be caused by canceling out two different error sources which is not clarified well in this study.

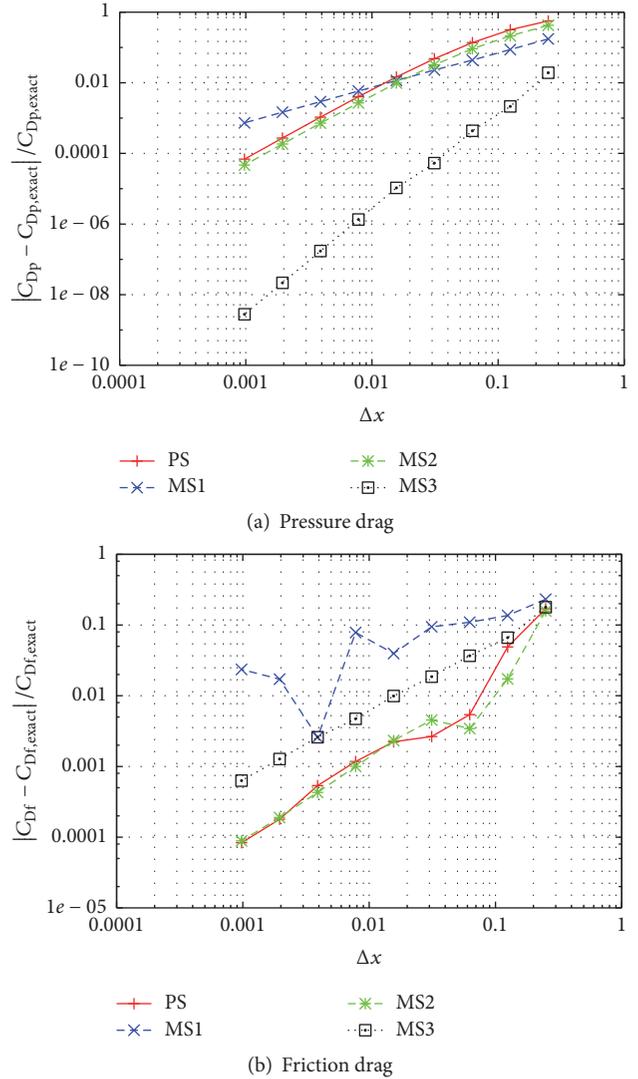


FIGURE 6: Estimated drag in Stokes flow around a sphere as functions of mesh spacing. Analytical pressure and friction drag are $C_{Dp,exact} = 800$ and $C_{Df,exact} = 1600$, respectively.

It is worth noting that the PS results are sensitive to the density of the polygon mesh at coarser polygon mesh resolutions, whereas we use a very fine polygon mesh in this study to eliminate the error introduced by coarse polygon mesh. This implies that the polygon density requires special attention in the PS method.

3.3.2. Force Distributions. Then, we visualized the force distributions on the each cell projected on x , y , and z plane. The partial drag coefficients for each projected cell are defined as follows:

$$\begin{aligned}
 C_{d_{p,x}} &= \frac{\int_{\text{cell}} p \, dy \, dz}{(1/2) \rho u_{\infty}^2 \Delta y \Delta z} \\
 C_{d_{f,y}} &= \frac{\int_{\text{cell}} \tau_{xy} + \tau_{yz} \, dx \, dz}{(1/2) \rho u_{\infty}^2 \Delta x \Delta z} \\
 &\vdots
 \end{aligned} \tag{22}$$

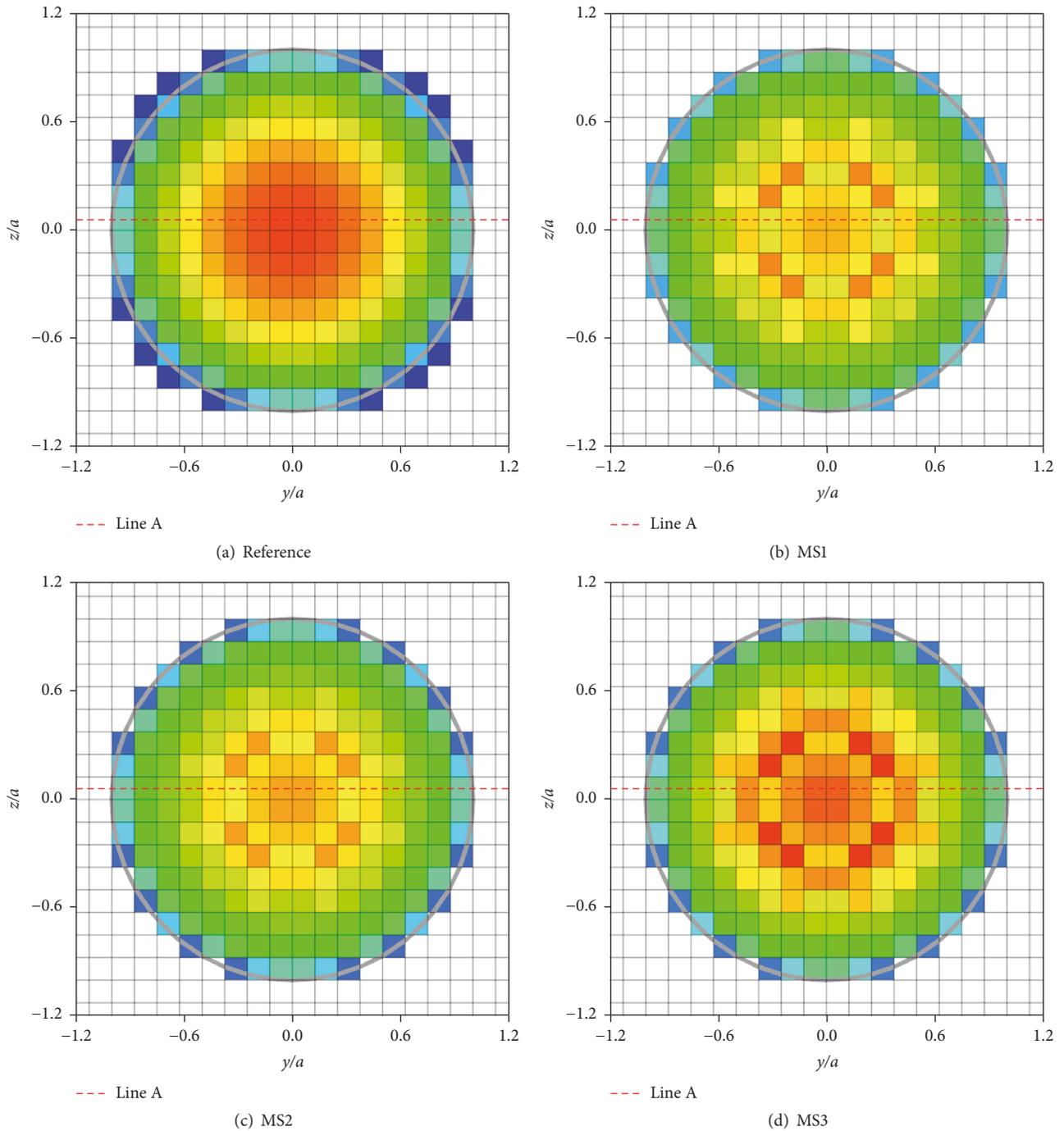


FIGURE 7: Distribution of partial pressure drag for each cell front-projected on x plane. Contour range is set to be $-0.1 < C_{dp,x}/C_{Dp,exact} < 0.8$. Analytical pressure drag is $C_{Dp,exact} = 800$.

Here, integration is conducted for the surface in each projected cell. In the case of a sphere, there are two projections: front and rear projections. In this definition, the reference solution can be obtained with very fine discretization. On the other hand, MSs evaluate them in simplified ways as in the inside summation of (10). The distributions of reference solutions and MSs are shown in Figures 7 and 8. Here, the reference solution is obtained with the ten times finer

submesh. The results of MS1 have strong oscillation in friction drag because this method does not compute shear stress correctly. On the other hand, MS2 and MS3 can predict them quantitatively, though they have weak oscillation. This behavior is illustrated in Figures 9 and 10, which shows the partial drag on the line indicated in Figures 7 and 8. These figures show the same trend more quantitatively. Figure 8 shows that the MS2 predict partial friction drag more

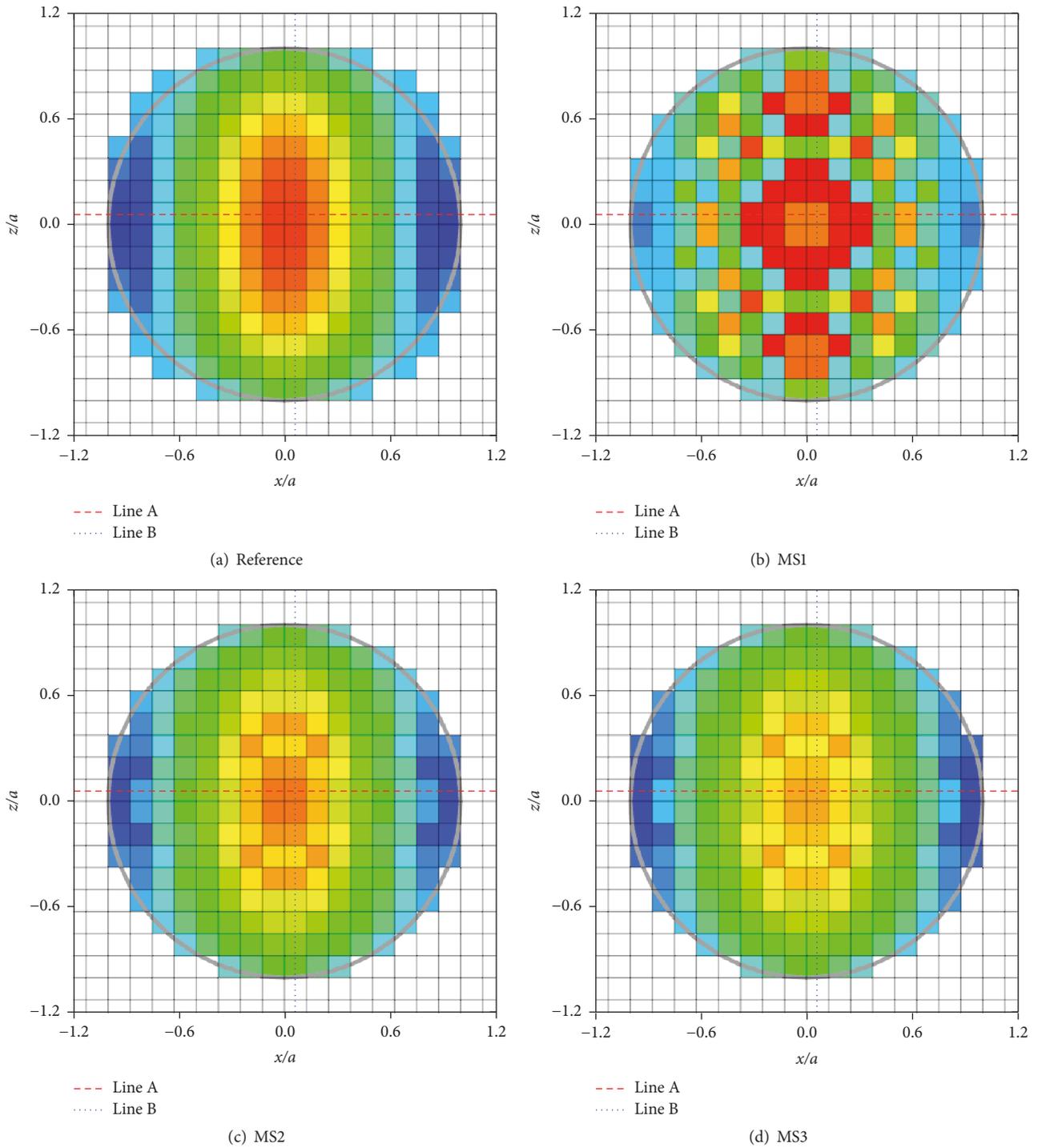


FIGURE 8: Distribution of partial friction drag for each cell front-projected on y plane. Contour range is set to be $-0.1 < C_{df,y}/C_{Df,exact} < 0.4$. Analytical pressure drag is $C_{Df,exact} = 1600$.

accurately than MS3, as discussed above for total friction drag.

3.3.3. Effects of Estimation Methods for Ghost Cells. In MS, the error in the friction drag appears to arise from two sources: (1) the estimated shear stress and (2) the estimated surface area vector. These error sources are distinguished for clarity.

The error in the estimated shear stress at the interface of the staircase body can be completely removed by analytically computing the shear stress in the Stokes flow, rather than approximating it by a finite difference scheme. The estimated drag with analytical shear stress at the boundary is plotted in Figure 11. This solution is labeled as “MS with the analytical shear stress” in the figure. Clearly, the analytical shear stress

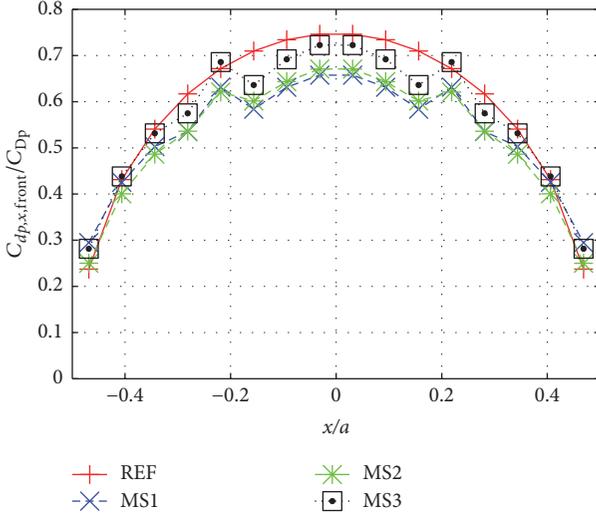


FIGURE 9: Distribution of partial pressure drag for each cell front-projected on x plane on line A in Figure 7. Analytical pressure drag is $C_{Dp,exact} = 800$.

yields much more accurate results than MS1, MS2, and MS3, in which the shear stress is numerically computed. This result illustrates that (1) the error in the estimated shear stress is significant and prevents the numerically computed force from converging, and (2) the error in the surface area vector is probably insignificant.

Also note that, according to Figure 11, the error in shear stress for MS2 depends on the accuracy of the ghost-cell method. Therefore, the order of accuracy in MS2 (shown in Figure 6) seems to be determined by the formal order of accuracy in the ghost-cell method.

3.3.4. Surface Area Vector Estimation. We now discuss the error in the estimated surface area vector. The surface area vectors ($x > 0$, $y > 0$, and $z > 0$ of the staircase body represented by the mesh faces) are summed in Figure 12. The sum converges to the exact integral over the surface area as the mesh density increases. It should be noted that the surfaces of the staircase body never exactly sum to the area of the body surface, even on very fine meshes. The estimated force acting on the body depends on the surface area vector, rather than on the surface area itself. This may explain why the staircase body approximation for integration does not matter.

3.3.5. Computational Costs. The computational costs for PS and MS2 for $N = 4$ to 128 are discussed in this subsection. Here, costs for MS1 and MS3 are almost the same as MS2. The computation is measured by a core of the computer process unit of Xeon X5450 3.0 GHz. For the PS computation, computational costs strongly depend on the number of polygon meshes. In this study, numbers of polygon mesh for each grid resolutions are determined with increasing it twice until the drag coefficient is firstly converged with 10% error. The results are shown in Table 1. The number of

TABLE 1: Computational costs for body force estimation of Stokes flow around a sphere.

N	PS		MS2
	Computational costs [s]	Number of polygons	Computational costs [s]
4	0.0006043	64×64	0.0001197
8	0.002353	128×128	0.0003260
16	0.04947	512×512	0.001319
32	0.1955	1024×1024	0.00677
64	0.6897	2048×2048	0.05297
128	2.939	4096×4096	0.3567

polygon meshes for convergence is larger than we expected, and resulting computational costs of PS are much higher than that of MS2. The computational cost of MS2 is 5–30 times lower than that of PS, and table shows that the MS2 is much more efficient than PS. This is because MS2 only requires the mesh-face loop to compute the body force while PS requires the polygon mesh loop, the number of which is much larger than that of the mesh face.

4. Evaluation of Force on a Moving Sphere: Various Alignment of Body and Grid

Then, the Stokes flow around a moving sphere is considered because the immersed boundary method is often used for the moving body problems. In the moving body problems, we should consider following additional two points:

- (i) change in the evaluation of the shear stress,
- (ii) the various locations of the body compared with the grid.

If we consider the exact solution of the Stokes flow around a moving sphere, it is exactly corresponding to the superimposition of the exact solution of the Stokes flow and the uniform velocity of moving body. If we use the MS1, MS2, MS3, and PS for the evaluation of the shear stress, it exactly corresponds to that in the static case because the uniform velocity added in the flow fields is perfectly canceled out by the boundary velocity imposed on the moving body. Therefore, the first point, that is, change in evaluation of the shear stress, does not appear for MS1, MS2, MS3, and PS, in the case we considered. On the other hand, the second point seems to be more important for moving body problems. The smaller force variation due to the alignment of the body and the grid is preferred. In this section, force variation due to the alignment of the body and the grid for each method is investigated.

In this test case, the center of the sphere is shifted by $(c_x \Delta x, c_y \Delta y, c_z \Delta z)$ from the original location used in the previous section, where $-0.5 < c_x \leq 0.5$, $-0.5 < c_y \leq 0.5$, and $-0.5 < c_z \leq 0.5$. We set $c_y = c_z = 0$ for simplicity and force is evaluated with changing c_x , assuming flow around the moving sphere through the quiescent air.

The force is evaluated with the grid resolutions of $N = 4, 8, 16, 32$, and 64 in this section. The results of $N = 16$ are

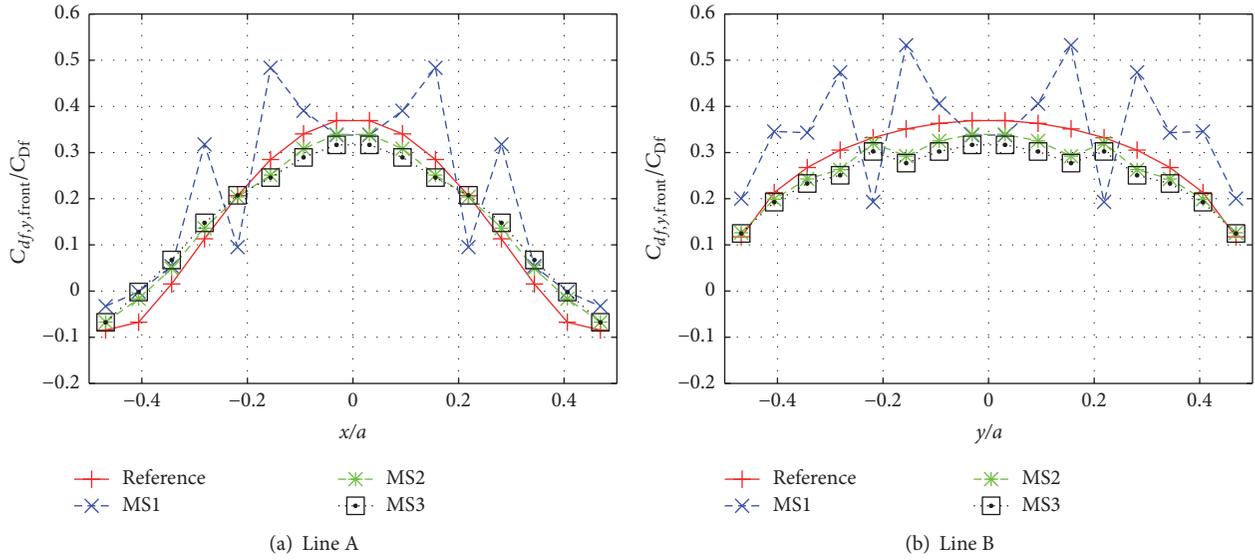


FIGURE 10: Distribution of partial pressure drag for each cell front-projected on x plane on lines A and B in Figure 8. Analytical friction drag is $C_{Df,exact} = 1600$.

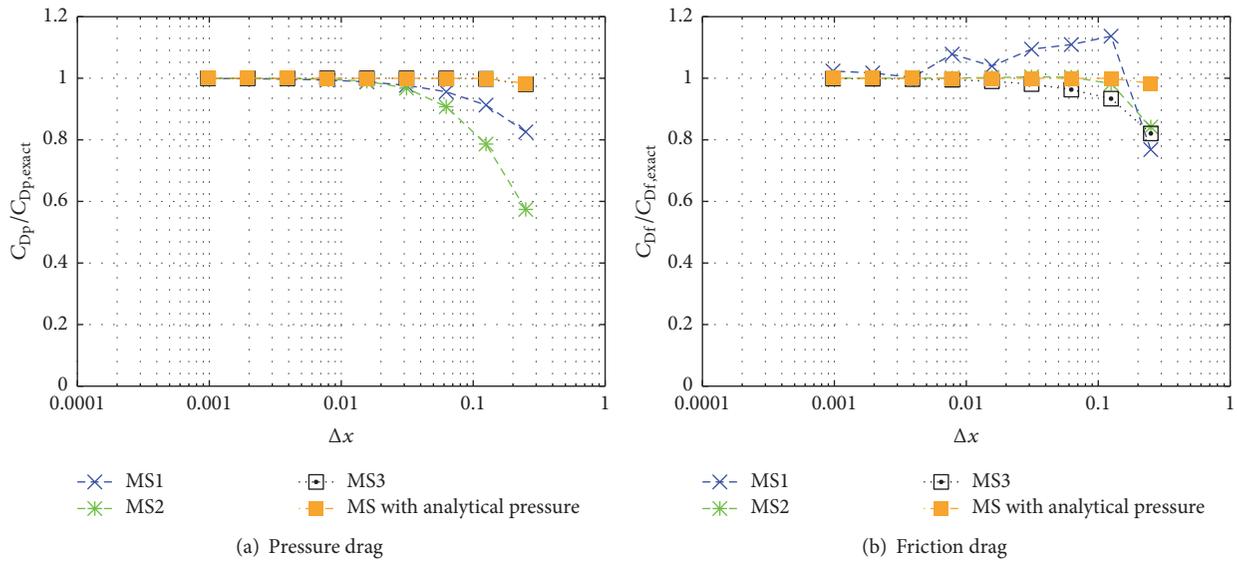


FIGURE 11: Estimated drag in Stokes flow around a sphere as functions of mesh spacing using a mesh-base scheme with analytical shear stress. Analytical estimates of pressure and friction drag are $C_{Dp,exact} = 800$ and $C_{Df,exact} = 1600$, respectively.

shown in Figure 13. The results show that the friction drag of MS1 has large oscillations depending on the alignment of the body and the grid, while the other results have much smaller oscillations. Interestingly, the oscillation is significantly suppressed by PS, though the accuracy of its averaged value is worse than MS2 in this case.

In order to quantitatively evaluate the force oscillation by the alignment of the body and the grid, the root mean square value is introduced in Figure 14, where the bracket represents the averaging operator for $-0.5 < c_x \leq 0.5$. The root mean square of the force oscillation normalized by the analytical force for each case is shown by the thin lines in Figure 14, together with the error of averaged force compared with the

analytical force by the thick lines. Here, MS1 has very large oscillations due to the alignment as well as very large errors in averaged value as discussed in the previous section. MS2 and MS3 have similar or smaller oscillations than the errors in averaged value and their oscillations are considered to be acceptable. Besides, PS has very small oscillation and shows the best characteristics for the grid alignment.

The test case shown in this section illustrates that the proposed MS2 and MS3 methods have similar or smaller oscillation than error and can be used for the moving body problems, though PS shows the best characteristics for the grid alignment of the body and the grid.

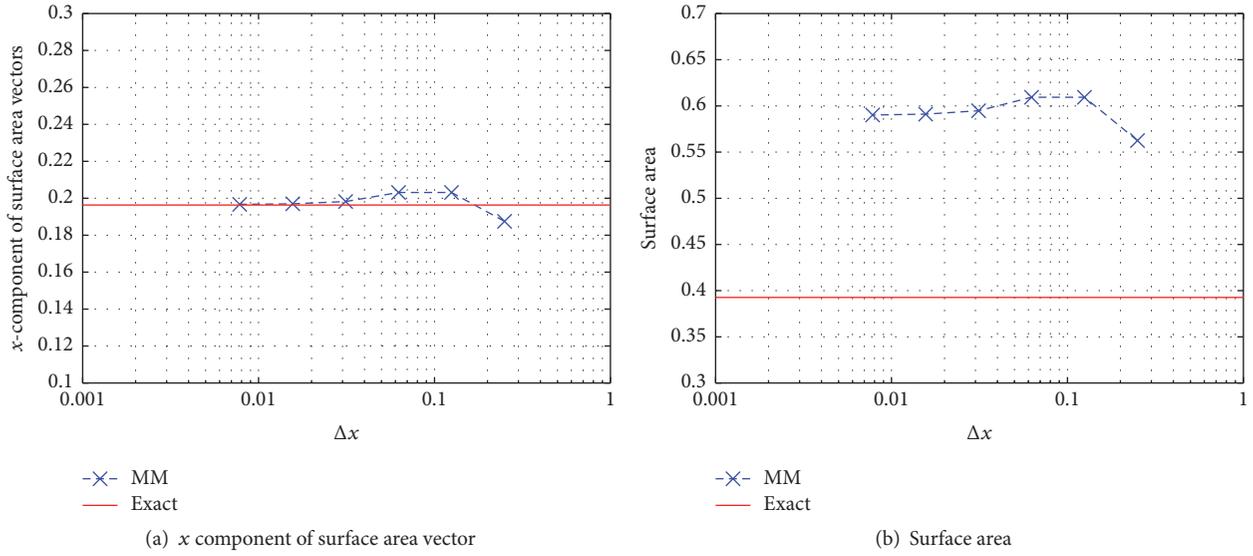


FIGURE 12: Evaluation of summation of surface area vectors and surface areas in MS.

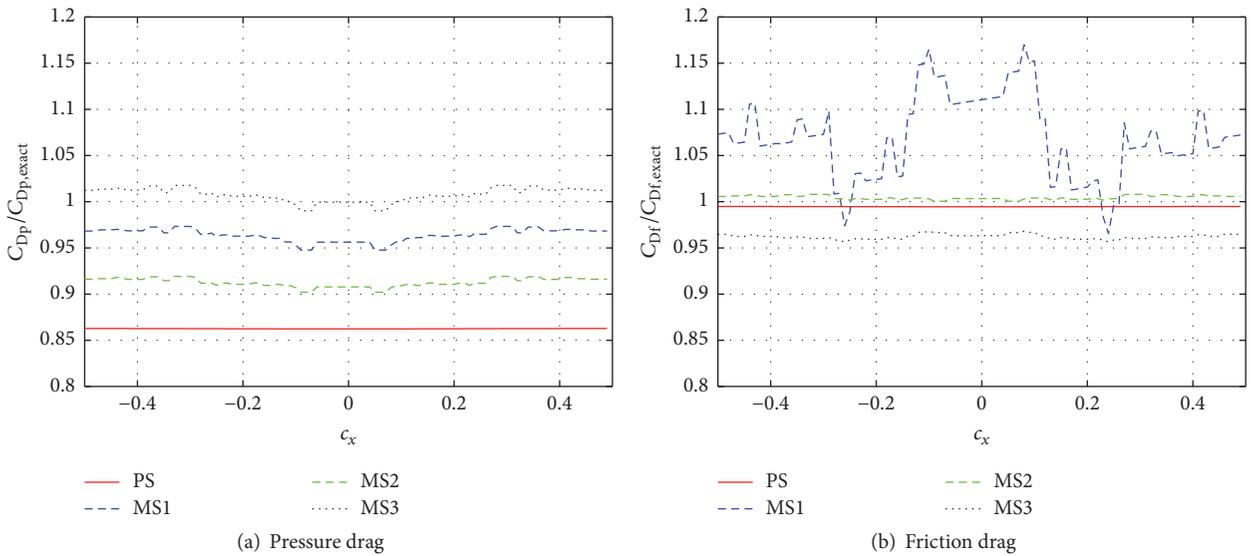


FIGURE 13: Effects of alignment of body and grid on estimated drag for $N = 16$. Analytical estimates of pressure and friction drag are $C_{Dp,exact} = 800$ and $C_{Df,exact} = 1600$, respectively.

5. Estimating the Force in the Flow around a Sphere at the Reynolds Number 100

To evaluate MS at the higher Reynolds number, the force was estimated in fluid flowing around a sphere at the Reynolds number 100. Since no analytical solution was available for such higher Reynolds number flows, numerical solution was obtained by using an incompressible flow solver developed by Onishi et al. [11]. This solver adopts a simple marker and cell algorithm with collocated variable layout. The grid system is based on the building cube method proposed by Nakahashi [12]. Space is discretized by a second-order finite difference scheme [13] and the simple immersed boundary method based on the ghost-fluid based method proposed

by Sato et al. [14]. This test case includes the errors in the computed flow fields, and therefore the result does not directly indicate the performance of the force estimation method.

The forces evaluated by PS, MS1, MS2, and MS3 are shown in Figure 15. The result shows that the forces estimated by PS, MS2, and MS3 seem to converge to 1.09, which is consistent with the existing data obtained by the empirical relation derived from a number of experiments [15] and the numerical computations [7, 16].

Interestingly, MS3 has less error in a coarse grid. This might be because fluid dynamic computation and force evaluation methods are consistent. On the other hand, the magnitudes of the errors in PS and MS2 are almost identical

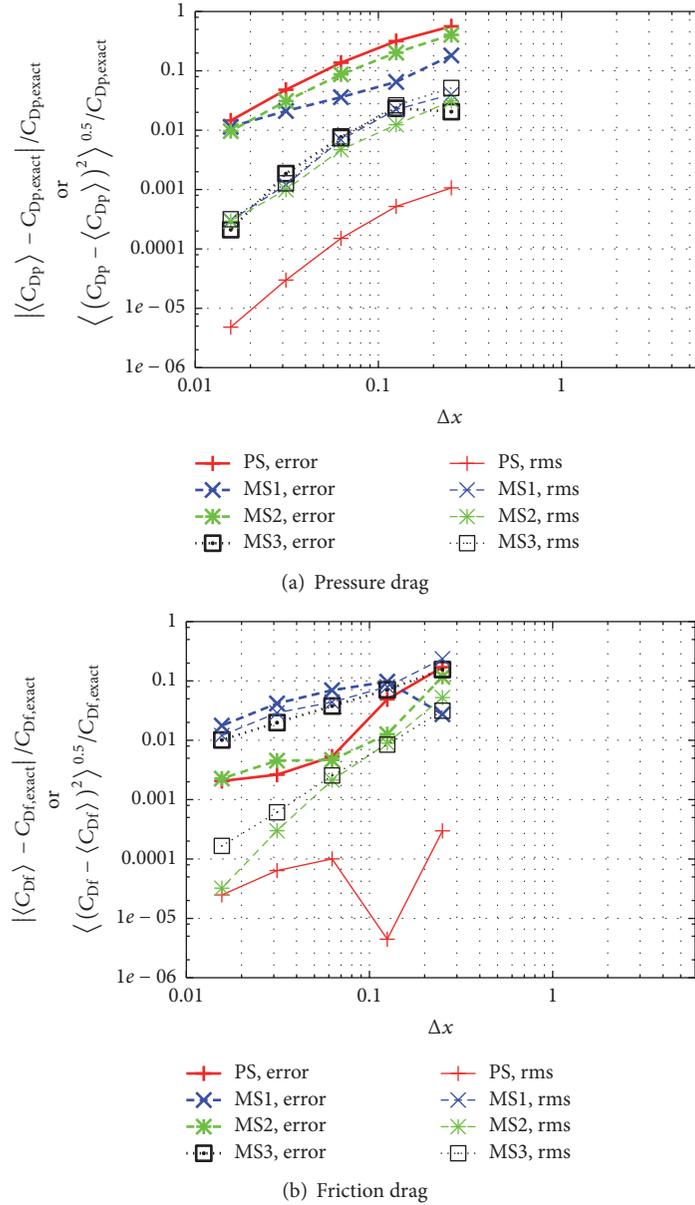


FIGURE 14: Estimated averaged drag and oscillation due to alignment of body and grid in Stokes flow around a moving sphere as functions of mesh spacing. Analytical estimates of pressure and friction drag are $C_{Dp,exact} = 800$ and $C_{Df,exact} = 1600$, respectively.

on the coarser and finer meshes. This result illustrates that MS2 and MS3 are adequate methods for evaluating forces acting on flows around objects in the immersed boundary method, although only the information of mesh and flow variables is used in the flow and boundary cell computations.

6. Applicability to More Complex Problems: Body with Sharp Edges, Multibody Treatment, and Compressibility

In this paper, we introduced the mesh-based force evaluation methods in the immersed boundary methods. When it is applied to the complex body problems, there are the sharp

edge problems or other multiple body problems. However, this method can be applied to such complex body problems, as far as the base immersed boundary method can treat such bodies. This is because the proposed method just uses the cell-face value which is given by the base immersed boundary method, regardless of whichever method, such as a ghost-cell method or a ghost-fluid method, is used as the base immersed boundary method. Therefore, the applicability of this method to the complex body problem is highly depending on the base immersed boundary methods and the further discussion is left for the other papers which discuss applicability of the base immersed boundary method to the complex body problems.

In addition, it is worth emphasizing that the present method can be applied to compressible flows as well as

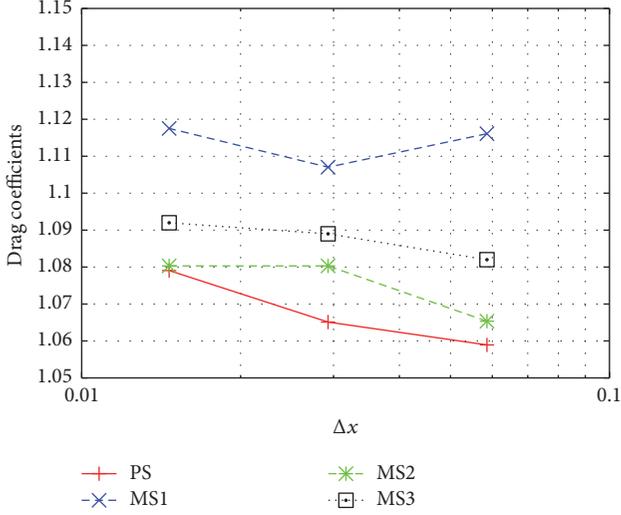


FIGURE 15: Estimated drags in flow around a sphere at the Reynolds number 100 as functions of mesh spacing.

incompressible flows. This is because the present method does not rely on the details of fluid flows (i.e., the divergence of velocity field). Although all the numerical results shown in this paper have been obtained by incompressible flows, several studies can be found in the literature on this subject [17, 18].

7. Conclusions

We have proposed a simple method for evaluating the forces acting on flows around bodies in the immersed boundary scenario. This method has been developed by employing a novel mesh-face integration method and an extrapolation method for evaluating pressure and shear stresses at the mesh faces, such as the first-order, ghost-cell, or ghost-fluid methods. The present method is, in principle, advantageous over the conventional methods based on control volumes in that pressure and shear stress can be evaluated separately.

Moreover, we have applied the present method to the computation of the drag force acting on a sphere in Stokes flow and have investigated the effects of grid spacing and extrapolation methods on the errors originating from the present force estimation method by using the existing analytical solutions. In addition, we have addressed the computational costs. As a result, the accuracy of the proposed mesh-based scheme has been proven to be comparable to that of the polygon-based scheme, which is commonly adopted in straightforward implementation. This indicates that the proposed scheme works better than the polygon-based one when complex geometries are involved, since its implementation is simple and its computational cost is low.

The error sources in the proposed implementation are sourced from (1) the surface area vector of the staircase body shape and (2) the approximated shear stress. Of these, error in the evaluated shear stress dominates and is significant. If the shear stress is appropriately evaluated, the fluid force

can be accurately obtained by summing over the mesh faces, because the surface area vector components converge with increasing grid density while the surface area does not. The shear stress is adequately evaluated by the second-order finite differencing scheme with the ghost-cell or ghost-fluid method. Sometimes, it is difficult to estimate the shear stress accurately with this method by its complex shape. It should be noted that this difficulty is caused by the immersed boundary methods themselves and the present idea using the staircase integration does not have difficulty.

Nomenclature

ΔS :	Surface area of a facet [m^2]
Δx :	Grid spacing [m]
δ_{ij} :	Kronecker's delta function
Γ_S :	Surface of a solid object
μ :	Dynamic viscosity [$\text{Pa}\cdot\text{s}$]
ρ :	Mass density [kg/m^3]
σ_{ij} :	Fluid stress tensor [Pa]
a :	Radius of a sphere [m]
C_{D_f} :	The coefficient of frictional drag
C_{D_p} :	The coefficient of pressure drag
d :	Distance from an interface [m]
d_1, d_2 :	Distance of a marker 1 (or 2) from an interface [m]
F_j :	Hydrodynamic force [N]
n_i :	Unit normal vector
p :	Pressure [Pa]
p_∞ :	Reference pressure [Pa]
r :	Distance from the center of a sphere [m]
Re:	Reynolds number
S :	Area of surface [m^2]
u_∞ :	Free stream velocity [m/s]
u_1, u_2 :	Velocity at a marker 1 (or 2) [m]
u_B :	Velocity at an interface [m/s]
u_i :	Velocity [m/s]
x_i :	Coordinate [m].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Taku Nonomura and Junya Onishi equally contributed to this paper.

Acknowledgments

A portion of this research was supported by the grant for "Strategic Programs for Innovative Research" Field no. 4, Industrial Innovations from the Ministry of Education, Culture, Sports, Science, and Technology's (MEXT's) "Development and Use of Advanced, High-Performance, General-Purpose Supercomputers Project," and JSPS KAKENHI Grant no. 17K06167. And a part of the results was obtained by using the K computer at the RIKEN Advanced Institute for Computational Science and by pursuing HPCI Systems

Research Projects (Proposals nos. hp130001 and hp130018). The authors express their thanks to all parties involved.

References

- [1] C. S. Peskin, "Fluid dynamics of heart valves: experimental, theoretical, and computational methods," *Annual Review of Fluid Mechanics*, vol. 14, pp. 235–259, 1982.
- [2] R. Mittal and G. Iaccarino, "Immersed boundary methods," in *Annual review of fluid mechanics*. Vol. 37, vol. 37 of *Annu. Rev. Fluid Mech.*, pp. 239–261, Annual Reviews, Palo Alto, CA, 2005.
- [3] M.-C. Lai and C. S. Peskin, "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity," *Journal of Computational Physics*, vol. 160, no. 2, pp. 705–719, 2000.
- [4] E. Balaras, "Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations," *Computers and Fluids*, vol. 33, no. 3, pp. 375–404, 2004.
- [5] L. Shen, E.-S. Chan, and P. Lin, "Calculation of hydrodynamic forces acting on a submerged moving object using immersed boundary method," *Computers and Fluids*, vol. 38, no. 3, pp. 691–702, 2009.
- [6] K. Ono and J. Onishi, "A challenge to large-scale simulation of fluid flows around a car using ten billions cells," *Supercomputing News*, vol. 15, pp. 59–69, 2013 (Japanese).
- [7] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, and A. von Loebbecke, "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," *Journal of Computational Physics*, vol. 227, no. 10, pp. 4825–4852, 2008.
- [8] Y. Takahashi and T. Imamura, "Force calculation and wall boundary treatment for viscous flow simulation using cartesian grid methods in," in *Proceedings of the 26th (Japanese) Computational Fluid Dynamics Symposium*, 2012.
- [9] F. Gibou, R. P. Fedkiw, L.-T. Cheng, and M. Kang, "A second-order-accurate symmetric discretization of the Poisson equation on irregular domains," *Journal of Computational Physics*, vol. 176, no. 1, pp. 205–227, 2002.
- [10] F. M. White, "Visous fluid flows," 1991.
- [11] J. Onishi, K. Ono, and S. Suzuki, "Development of a CFD software for large-scale computation: An approach to grid generation for arbitrary complex geometries using hierarchical blocks," in *Proceedings of the 12th International Symposium on Fluid Control, Measurement and Visualization (FLUCOME '13)*, 2013.
- [12] K. Nakahashi, "High-Density Mesh Flow Computations with Pre-/Post-Data Compressions," in *Proceedings of the 17th AIAA Computational Fluid Dynamics Conference*, Toronto, Ontario, Canada.
- [13] Y. Morinishi, T. S. Lund, O. V. Vasilyev, and P. Moin, "Fully conservative higher order finite difference schemes for incompressible flow," *Journal of Computational Physics*, vol. 143, no. 1, pp. 90–124, 1998.
- [14] N. Sato, S. Takeuchi, T. Kajishima, M. Inagaki, and N. Hori-nouchi, "A cartesian grid method using a direct discretization approach for simulations of heat transfer and fluid flow," *Joint EUROMECH/ERCOTAC Colloquium 549—Immersed Boundary Methods: Current Status and Future Research Directions*, 2013.
- [15] M. W. R. Clift and J. Grace, *Bubbles, Drops, and Particles*, Dover Publications, 2005.
- [16] M. Tabata and K. Itakura, "A precise computation of drag coefficients of a sphere," *International Journal of Computational Fluid Dynamics*, vol. 9, no. 3-4, pp. 303–311, 1998.
- [17] S. Takahashi, T. Nonomura, and K. Fukuda, "A numerical scheme based on an immersed boundary method for compressible turbulent flows with shocks: application to two-dimensional flows around cylinders," *Journal of Applied Mathematics*, Article ID 252478, Art. ID 252478, 21 pages, 2014.
- [18] Y. Mizuno, S. Takahashi, T. Nonomura, T. Nagata, and K. Fukuda, "A simple immersed boundary method for compressible flow simulation around a stationary and moving sphere," *Mathematical Problems in Engineering*, Article ID 438086, Art. ID 438086, 17 pages, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

