A unified interpolation stencil for ghost-cell immersed boundary method for flow around complex geometries

Hosnieh KOR*, Mehdi BADRI GHOMIZAD* and Koji FUKAGATA*

*Department of Mechanical Engineering, Keio University
Hiyoshi 3-14-1, Kohoku-ku, Yokohama 223-8522, Japan
E-mail: fukagata@mech.keio.ac.jp

Received: 10 February 2017; Revised: 2 May 2017; Accepted: 17 May 2017

Abstract
We propose a unified interpolation stencil that is used for a ghost-cell immersed boundary method to satisfy wall boundary conditions in Cartesian-based numerical simulation of fluid flow with complex boundaries. As other ghost-cell methods do, the numerical boundary point is considered in the solid region and the required velocity is interpolated directly from the proximate points in the fluid region. In this paper, we propose a unified interpolation scheme based on a sequence of one-dimensional interpolations. Different interpolation stencils are examined and their convergence rates are compared by solving a benchmark problem on the flow between the concentric cylinders. In contrast to typical standard stencils, the proposed ones are versatile and do not require to be altered according to the irregularities in boundary shape. Namely, the boundary condition can be accurately imposed with a unique stencil for all numerical boundary points while preserving the convergence rate of the flow solver. Performance of the proposed method is studied by solving two-dimensional incompressible flows around a circular cylinder, a square cylinder, and a square cylinder inclined with respect to the main flow. Comparison with the existing numerical and experimental data shows good agreement, which confirms the capability of the proposed method.

Key words: Immersed boundary method, Ghost cell method, Interpolation stencil, Complex geometries, Incompressible viscous flow

1. Introduction

Systems involving complex stationary or moving bodies surrounded by a fluid flow are of great interest to many engineering and biological applications, such as airplane wings, wind turbines and swimming of fish, to name a few. These problems are conventionally solved by body-fitted grid-based methods, in which the governing equations are discretized on unstructured or structured grids that are aligned with the boundaries. Thus, simulating the flow around moving or deformable obstacles is cumbersome due to computationally expensive re-meshing procedures required at each time step.

In contrast, immersed boundary (IB) methods represent an attractive alternative approach to tackle flows with complex geometries. In recent years, these methods have become very appealing due to their simplicity, flexibility, and efficiency. The fundamental idea underlying these methods is to separately consider the immersed body and the fluid mechanics; namely, the fluid is solved on a regular Eulerian computational grid, while the immersed boundary is represented, e.g., by marker points. With this in mind, the immersed boundaries are mostly emulated either by adding fictitious forces to the momentum equations or by modifying the background grid structures as it is done in cut-cell approaches. This relaxation of boundary-conforming requirement, i.e., no need for aligning the background computational grid with the boundary, considerably simplifies the task of grid generation for complex geometries and also removes the need for re-meshing strategies for moving bodies. Besides the simplicity, it brings about higher efficiency as we can employ Cartesian flow solvers with energy conserving discretization properties. The advantages of Cartesian-based methods make them suitable for Direct Numerical Simulation (DNS) of complex turbulent flows where less numerical dissipation is of particular significance.
Although the effects of boundaries are accounted varyingly in different IB methods, they can be broadly classified into two main groups (Mittal and Iaccarino, 2005). The first category refers to the continuous forcing methods, in which a forcing term is added to the Navier-Stokes equations before they are discretized. Such methods are originally introduced by Peskin (1972) to simulate the cardiac mechanisms. In this approach, the forces at the immersed surfaces are calculated from proper constitutive laws (e.g., Hook’s law). Then, this method was extended by Goldstein et al. (1993) and Saiki and Biringen (1996) to be suitable for rigid bodies, in which they had to introduce some numerical parameters to tackle the complications of calculating the fictitious forces. However, these parameters affect the numerical stability and accuracy to some extent. For this reason, continuous forcing methods are basically suited for problems involving flexible elastic bodies (Mittal and Iaccarino, 2005). That the boundary conditions on the body surface are not accurately satisfied and the boundary is smeared over some points is another shortcoming of these methods. Therefore, use of this method is not recommended especially for flows at high Reynolds numbers (Iaccarino and Verzicco, 2003).

The second category refers to the discrete forcing methods, in which discrete forcing terms are directly applied to the discretized Navier-Stokes equations just in the vicinity of the solid body. This technique considerably improves the local accuracy because the effect of the boundary is not spread like the continuous forcing methods. Approaches such as those of Mohd-Yusof (1997), Udaykumar et al. (1999), Ye et al. (1999), Fadlun et al. (2000), Kim et al. (2001), Gibou et al. (2002), You et al. (2003), Balaras (2004), Marella et al. (2005), Ghias et al. (2007), and Ikeno and Kajishima (2007) fall in this category. In these methods, the velocity boundary condition can be precisely satisfied in the momentum solution. The capability of representation of sharp interface attracts attention especially for simulating flows at high Reynolds numbers (Mittal et al., 2008). Another merit in these methods is that a user-defined parameter is not required for calculating the forcing term. However, the expression for the forcing term is dependent on the discretization scheme, and these methods often involve additional procedures to avoid numerical instabilities, e.g., when the interpolation point is too close to the boundary.

Another IB approach that has much in common with direct forcing methods and sometimes is considered as an extension of the direct forcing method is the ghost-cell method (Fedkiw, 1999; Fedkiw, 2002; Tseng and Ferziger, 2003; Mittal et al., 2008). This method is based on the calculation and reconstruction of the flow properties on some grid points in the solid phase which are adjacent to the physical boundary. The value of these points is directly interpolated from the grid points in the fluid region. In this method, we do not need neither smearing of the boundary nor any momentum explicit forcing terms on fluid points to precisely impose the boundary conditions on the surface. These methods bring about a sharp representation of the immersed bodies and let the Navier-Stokes equation govern all the points in the fluid domain of interest.

In all of the IB methods mentioned above, interpolations are needed in order to consider the effect of the bodies. However, it is hardly possible to use just one interpolation stencil for all points since there are always special cases. For instance, one of the points on the interpolation stencil is inside the boundary and the value there is unknown (probably it is another ghost point): in such a case, a different treatment is used (Kim et al., 2001; Yang and Balaras, 2006). To avoid such a complexity, which makes the coding especially cumbersome, we propose a new method based on a one-dimensional interpolation in the direction of normal to the surface and just using values in the fluid region, by which a single stencil can be employed throughout the computational domain.

In the present paper, we consider the ghost-cell method that allows us to simulate incompressible flows involving complex two-dimensional stationary immersed boundaries that do not coincide with the underlying Cartesian grids. As for the interpolation, which is very influential on the accuracy, a new scheme based on the simple linear interpolation is proposed. Then, the proposed scheme is validated in some conventional test cases. For brevity, only a limited number of cases are presented in this paper.

This paper is organized as follows. Section 2 describes the main features of the flow solver; different interpolation stencils are introduced and compared in Section 3; results of the numerical simulations are presented and validated in Section 4; and finally, conclusions are drawn in Section 5.

2. Numerical method

In this section, we will first describe the numerical procedure for flow simulation and then we introduce the common procedure used in the present ghost-cell method.
2.1. Governing equations

The problem of interest here is a fixed obstacle with an arbitrary shape in a two-dimensional unsteady, viscous, incompressible flow. The governing equations are the continuity equation and the Navier-Stokes equation in terms of the primitive variables, i.e.,

\[ \nabla \cdot \mathbf{u} = 0, \quad (1) \]

\[ \frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u}\nabla) - \nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}, \quad (2) \]

where \( \mathbf{u} \) is a velocity vector normalized by the reference velocity \( U \); \( p \) denotes the pressure normalized by \( \rho U^2 \), where \( \rho \) is the density of the fluid. The Reynolds number is defined as \( Re = UL/v \), where \( L \) is the reference length and \( v \) is the kinematic viscosity of the fluid.

In the present paper, we discretize this set of equations on a Cartesian mesh using a staggered grid system, although in general a different discretization scheme may be chosen separately. The spatial discretization is done by using the energy-conservative second-order finite difference method (Kajishima, 1999). The time integration is based on a fractional-step method, where a pseudo-pressure is used to correct the velocity field so that the continuity equation is satisfied at each computational time step. We use a third-order Runge-Kutta/Crank-Nicolson (RK3/CN) scheme for the time integration of convection and diffusion terms. The present simulation code is developed based on the DNS code of Fukagata et al. (2006), extended to deal with an inflow and outflow boundary condition (Kametani and Fukagata, 2011); the base code without immersed boundary has been validated for a turbulent channel flow and a spatially developing turbulent boundary layer.

2.2. Ghost-cell method

The present study employs a ghost-cell method. Namely, the values of the flow properties inside the obstacle are directly imposed at every time step to exactly satisfy the boundary condition on the physical boundary.

The first step in the ghost-cell method is to identify the ghost points inside the body. The ghost points are the nodes nearest to the physical boundary that have at least one neighbor on the fluid side. To do so, a proper description of the physical boundary would be invaluable. Here we opt for the methodology introduced by Udaykumar et al. (1996, 1997): the interface between the solid and fluid regions is represented as a series of interfacial markers connected by piecewise quadratic curves. The distance between the markers is set equal to the local grid size and the quadratic polynomial is fitted to every marker (\( i \)) with its two neighbors (\( i + 1 \) and \( i - 1 \)). Implementations of this methodology to a wide range of problems have shown its efficiency (Balaras, 2004; Udaykumar et al., 1996, 1997).

Having had a well-represented distribution of the physical boundary, we can start identifying the ghost points, i.e., the seeking and tagging process around the physical boundary. The details of this process can be found in Balaras (2004). The important aspects of this procedure are as follow. Given the location of every physical boundary point, we are able to determine a box of \( 3 \times 3 \) closest grid nodes around each of them (we consider a two-dimensional case for simplicity in explanation). To recognize which of these points are inside the solid region, a normal vector \( \mathbf{n} \) from all of them to the physical boundary is constructed. Then, the inner product of \( \mathbf{n} \) and the normal vector of the surface \( \mathbf{n} \) determines whether the point is inside the body or not. Since these two vectors are parallel to each other, the sign of the inner product indicates whether the two vectors are in the same direction or not. According to the definition, the ghost point is any point in the package which is inside the body and has at least one fluid-side neighbor belonging to the very \( 3 \times 3 \) box. The next step is computing the value of the velocity components on the ghost points. The details of this calculation are explained in the next section.

3. Interpolation scheme

3.1. Image point

In order to find any flow properties that should be imposed on the ghost point, one has to extrapolate those properties from the computational points in the fluid region. However, such an extrapolation often causes numerical instabilities when the fluid points used for extrapolation is too close to the boundary (Majumdar et al., 2001; Kim et al., 2001). Although this extrapolation is algebraically correct, the large negative weighting factor in the extrapolation equation (even for a simple linear extrapolation) results in severe instabilities especially at the very first time steps.

The technique we use here to avoid such numerical instabilities is to mirror-reflect the location of the ghost point into the fluid domain along the normal vector \( \mathbf{n} \), as shown in Figure 1, similar to Mittal et al. (2008). This mirror-reflecte
2.3. Interpolation stencils

For the computation of quantities on the image point, \( \varphi_{ip} \), we propose two different interpolation stencils as described in the following.

**Stencil 1** The first stencil we consider is as shown in Fig. 1 (referred to as Stencil 1). A normal line from a ghost point to the immersed boundary is extended to the fluid domain to find its intersections with two adjacent grid lines either horizontal or vertical, as indicated by filled triangles in Fig. 1. As illustrated in Fig. 1(b), this one-dimensional coordinate point to the immersed boundary is extended to the fluid domain to find its intersections with two adjacent grid lines either horizontal or vertical, as indicated by filled triangles in Fig. 1. As illustrated in Fig. 1(b), this one-dimensional coordinate along the normal line is referred to as \( \xi \), whose origin is defined at the ghost point.

As shown in Fig. 1(a), the location of the first two intersection points can be before or after the image point. We use the values of the flow properties on these intersections, \( \varphi_1 = \varphi(\xi_1) \) and \( \varphi_2 = \varphi(\xi_2) \) for calculating the value on the image point, \( \varphi_{ip} = \varphi(\xi_{ip}) \), using the third-order Lagrange interpolation (Burden and Faires, 2005), i.e.,

\[
\varphi_{ip} = \frac{(\xi_{ip} - \xi_1)(\xi_{ip} - \xi_2)}{(\xi_{ph} - \xi_1)(\xi_{ph} - \xi_2)} \varphi_{ph} + \frac{(\xi_{ip} - \xi_{ph})(\xi_{ip} - \xi_2)}{(\xi_1 - \xi_{ph})(\xi_1 - \xi_2)} \varphi_1 + \frac{(\xi_{ip} - \xi_{ph})(\xi_{ip} - \xi_1)}{(\xi_2 - \xi_{ph})(\xi_2 - \xi_1)} \varphi_2, \tag{5}
\]

where \( \xi_{ip} \) denotes the absolute value, and \( \varphi_1 \) and \( \varphi_2 \) are equal (see Fig. 1(a)), the quantity at the ghost point \( \varphi_g = \varphi(x_g, y_g) \) is simply calculated by

\[
\varphi_g = 2\varphi_{ph} - \varphi_{ip}, \tag{4}
\]

where \( \varphi_{ph} = \varphi(x_{ph}, y_{ph}) \) denotes the value that we want to impose at the physical boundary point.

Note that different image points should be defined for different quantities when they are defined at different ghost points, such as in the case of staggered grid system. Also note that, for the fixed bodies, we need to determine the location of the ghost points and other points in the interpolation stencil just once; thus a small portion of the simulation time is devoted to this determination. For moving bodies, in contrast, the procedure of finding the ghost points should be repeated every time step.

3.2. Interpolation stencils

For the computation of quantities on the image point, \( \varphi_{ip} \), we propose two different interpolation stencils as described in the following.

**Stencil 1** The first stencil we consider is as shown in Fig. 1 (referred to as Stencil 1). A normal line from a ghost point to the immersed boundary is extended to the fluid domain to find its intersections with two adjacent grid lines either horizontal or vertical, as indicated by filled triangles in Fig. 1. As illustrated in Fig. 1(b), this one-dimensional coordinate along the normal line is referred to as \( \xi \), whose origin is defined at the ghost point.

As shown in Fig. 1(a), the location of the first two intersection points can be before or after the image point. We use the values of the flow properties on these intersections, \( \varphi_1 = \varphi(\xi_1) \) and \( \varphi_2 = \varphi(\xi_2) \) for calculating the value on the image point, \( \varphi_{ip} = \varphi(\xi_{ip}) \), using the third-order Lagrange interpolation (Burden and Faires, 2005), i.e.,

\[
\varphi_{ip} = \frac{(\xi_{ip} - \xi_1)(\xi_{ip} - \xi_2)}{(\xi_{ph} - \xi_1)(\xi_{ph} - \xi_2)} \varphi_{ph} + \frac{(\xi_{ip} - \xi_{ph})(\xi_{ip} - \xi_2)}{(\xi_1 - \xi_{ph})(\xi_1 - \xi_2)} \varphi_1 + \frac{(\xi_{ip} - \xi_{ph})(\xi_{ip} - \xi_1)}{(\xi_2 - \xi_{ph})(\xi_2 - \xi_1)} \varphi_2, \tag{5}
\]
which has the third-order accuracy with respect to the spacing between stencil points.

The values at the interpolation stencils, i.e., \( \varphi_1 \) and \( \varphi_2 \), can be computed simply using a one-dimensional linear interpolation from the two grid points around them, since the points of interpolation stencil are defined on the grid-line. This is another advantage of the proposed method in contrast to the other ghost-cell IB methods that requires bilinear interpolation in general. Using the values at two neighboring grid points, \( \varphi_{1a} \) and \( \varphi_{1b} \) as exemplified in Fig. 1(b), the interpolation formula for \( \varphi_1 \) simply reads

\[
\varphi_1 = \alpha \varphi_{1a} + (1 - \alpha) \varphi_{1b}, \tag{6}
\]

where \( \alpha = \delta / \Delta x \) (or \( \delta / \Delta y \)) with \( \delta \) being the distance between the intersection point and the grid point \( 1a \). The interpolation for \( \varphi_2 \) is done similarly using \( \varphi_{2a} \) and \( \varphi_{2b} \).

From above, the overall convergence rate of the proposed method is expected to be of the second order.

**Stencil 2** Another interpolation stencil we consider is illustrated in Fig. 2 (referred to as Stencil 2). This stencil is a three-point interpolation stencil. The main difference from Stencil 1 is that we use only two points closest to the image point (enclosed by lines in Fig. 2), which are either two intersection points or the physical boundary point and one intersection point. In this case, the linear interpolation can be applied instead of Eq. (5). If the two neighboring points are the two intersection points, it reads

\[
\varphi_{ip} = \frac{(\xi_{ip} - \xi_2) \varphi_1 + (\xi_1 - \xi_{ip}) \varphi_2}{\xi_1 - \xi_2}. \tag{7}
\]

Likewise, if the two neighboring points are the physical boundary point and the intersection point \( \xi_1 \), it reads

\[
\varphi_{ip} = \frac{(\xi_{ip} - \xi_{ph}) \varphi_1 + (\xi_1 - \xi_{ip}) \varphi_{ph}}{\xi_1 - \xi_{ph}}. \tag{8}
\]

In this case, too, the expected overall convergence rate is the second order.

### 3.3. Convergence rate for different stencils

To confirm the rate of convergence, we solve the flow between two concentric cylinders using the proposed method. As shown in Fig. 3(a), the inner cylinder is rotating at the angular velocity \( \omega \) (i.e., \( \omega_0(R_2) = \omega R_2 \)) and the outer cylinder is fixed (i.e., \( \omega_0(R_2) = 0 \)), where \( R_1 = 0.5 \) and \( R_2 = 1.5 \) denote the radii of the inner and outer cylinders, respectively.
Fig. 3: The concentric cylinder problem: (a) schematics of the problem; (b) velocity profiles in different resolutions, where $N$ denotes the number of grid points in one diameter of outer cylinder (in one dimension; the total number of grid points is $N^2$); (c) zoom-up view in the region enclosed by the circle inside Fig. 3(b).

Figures 3(b) and 3(c) shows the velocity profiles computed with different resolutions together with the analytical solution for this problem, i.e.,

$$u_0(r) = \frac{1}{R_2^2 - R_1^2} \left[ - (\omega R_1^2) r + \frac{R_1^2 R_2^2}{r} \omega \right],$$

and the error is calculated as the $L_2$ norm of the difference between Eq. (9) and the computed velocity. As can be observed in the figure, the velocity profile is well reproduced by the present method, and it converges to the analytical solution as the resolution is increased. The kinks observed in the region of $r < 0.5$ is simply due to that this region is inside the inner cylinder and values are not explicitly imposed except for the ghost point closest to the boundary.

Figure 4 shows the convergence rate for the different stencils considered. Although a second-order convergence rate was expected for Stencil 1, the expected order was not obtained. The convergence rate for the three-point stencil (Stencil 2) is found to be nearly second order. For Stencil 1, use of the higher-order, but less compact interpolation may have a negative influence on the accuracy. The compactness of Stencil 2 itself is another merit over Stencil 1 because the point of interpolation stencil is located within one diagonal grid spacing. This is especially so when the boundary shape is highly concave and local grid refinement is required so that all the points of interpolation stencil are located in the fluid region.

One of the attractive features of the proposed method is its unified interpolation procedure for any ghost points. In this benchmark problem, too, we exactly used the interpolation stencils as described in Sec. 3.2 without any exceptions or any additional treatments. In other schemes in literature, in contrast, there are usually many situations where an exceptional stencil has to be used. For instance, the trilinear interpolation scheme that Mittal et al. (2008) employed to calculate the value of a generic variable in three-dimensional simulation encounters special cases where one of the eight nodes in the stencil is a ghost point. In this case, they replaced it with the corresponding physical boundary point. As another example, Kim et al. (2001) applied the bilinear interpolation for the two-dimensional case, while they needed to substitute it by the linear interpolation for some cases. In the present method, in contrast, the interpolation is carried out simply in the direction normal to the surface. Therefore, as long as the surface curvature is locally smooth such as surface in cylinder and airfoil, all the nodes of the stencil are automatically defined outside the body.

4. Some numerical examples

In this section, results of the numerical simulations of flows around a circular cylinder and a square cylinder are presented and validated. In all cases, Stencil 2 is used to implement the boundary condition. Following Kim et al. (2001), the size of computational domain is set to be $70d$ in the streamwise ($x$) direction and $100d$ in the transverse ($y$) direction, where $d$ is the cylinder diameter. The center of the cylinder is located at $30d$ downstream the inlet. The computational grid is uniform in $x$ direction and nonuniform in $y$ direction. The number of the grid points are $2048$ in $x$ direction and $6$.
500 in $y$ direction, whereof about 30–40 grid points (depending on the problems) are uniformly distributed within the cylinder diameter in every direction. A uniform velocity at the inlet and a convective boundary condition at the outlet of the computational domain are imposed.

4.1. Flow over a circular cylinder

The main reason for simulating the flow around a circular cylinder at $Re = 40$ and 100 is the variety of the numerical and experimental results that have been obtained so far, which makes it possible to verify the proposed interpolation scheme.

Table 1 presents the wake length $L_w$, the separation angle $\theta_s$, the drag coefficient $C_D$, and two parameters on the wake shape $a$ and $b$ defined in Fig. 5 computed for the flow at $Re = 40$. The present results agree well with those of the other simulations (Linnick and Fasel, 2005; Taira and Colonius, 2007) and the experiment (Coutanceau and Bouard, 1977). Note that ”streamlines” presented in Fig. 5 are the trajectories of fluid particles but not the streamlines with their strict definition; however, since the results of present study are compared with the existing data of what they call ”streamlines,” we follow the same manner.

With regard to the unsteady features of flow at higher Reynolds number, we performed the simulation at $Re = 100$. This Reynolds number brings about vortex shedding as shown in Fig. 6. The drag coefficient $C_D$, the root-mean-square fluctuations of lift coefficient $C_{L\text{rms}}$, and the Strouhal number $St$ obtained with the present simulation are compared in Table 2 with those reported in literature obtained using different techniques. The table also shows the values of $C_{L\text{rms}}$ and $St$ calculated using the empirical correlation derived from the compilation of a number of data (Norberg, 2003). Although some scatters are found even among previous studies, the present results are found to be in reasonable agreement with the literature.

4.2. Flow over a square cylinder

As for the flow around a square cylinder, which has many different applications in engineering such as skyscrapers, towering structures and long-spanned bridges (Yoon et al., 2010), we carried out some simulations to examine the capability of the proposed method to capture the flow around sharp edges. In addition, we compared some characteristics of the flow past a square cylinder with and without inclination with other numerical results. Recent studies showed that there are three different flow patterns around a square cylinder in the flow concerning the positioning of the four sharp edges (Yoon et al., 2010). Due to change of the location of the separation points, three distinguishable flow patterns are steady (S), vortex merging (VM), and main separation (MS).

As examples of the steady flow pattern, the flows around a square cylinder at two different angles of inclination with respect to the main flow, $\theta = 0^{\circ}$ and $29.7^{\circ}$, at Re = 20 are considered. Streamlines of the flow around square cylinder at $\theta = 0^{\circ}$ and $29.7^{\circ}$ are presented in Fig. 7. It should be noted that, following Yoon et al. (2010), the representative length

![Fig. 4: Convergence rate for the various interpolation stencil: (a) four-point stencil (Stencil 1); (b) three-point stencil (Stencil 2).](image-url)
is defined as the projection length but not the edge length of the cylinder, which is identical to the edge length in \( \theta = 0^\circ \) case but not in \( \theta \neq 0^\circ \) as can be noticed from Fig. 7. The flow is symmetric when \( \theta = 0^\circ \) but asymmetric for the inclined cylinder. Table 3 quantitatively compares the re-circulation length in both cases, showing good agreement between the present results and literature.

Subsequently, as an example of the MS pattern, a flow at \( \text{Re} = 100 \) without inclination is considered. In this case, too, the grid is intentionally generated not to match the boundary. The main objective here is to investigate whether the present method can accurately reproduce the flow pattern and the period of vortex shedding. In Fig. 8, two instantaneous streamlines are shown, which agree well with the result of Yoon et al. (2010).

The drag coefficients \( C_D \) at different Reynolds numbers (\( \text{Re} = 20, 60, 100 \)) are presented in Table 4. It shows good agreement with literature. However, \( C_D \) computed using IB methods (i.e., both present and Yoon et al. (2010)) is slightly lower than those obtained using the body-fitted Cartesian simulation. Among the references presented in Table 4, Anzai et al. (2017) used the simulation code identical to the present one except for the treatment of boundary, which suggests that the slight difference observed in \( C_D \) is purely due to the different boundary treatments.

Finally, we examine the VM pattern, which appears when the Reynolds number and the angle of incidence are high. Figure 9 shows instantaneous streamlines in a period of vertex shedding of the flow around a cylinder at \( \text{Re} = 130 \) and \( \theta = 34.8^\circ \). The main feature of this pattern is that two small vortices, which are formed after separation, merge and reattach at two neighboring square edges. In this regard, Fig. 9 indicates that our scheme properly captures this flow pattern. To have more quantitative comparison, the profiles of mean pressure coefficient on the cylinder wall \( C_p = (p_w - p_\infty)/(1/2)pU^2 \) (where \( p_w \) and \( p_\infty \) denote the wall pressure and the free-stream pressure, respectively) at \( \text{Re} = 100 \) and \( \theta = 0^\circ, 29.7^\circ \) are compared in Fig. 10. Table 5 compares the mean pressure drag coefficient \( C_{Dp} \) and the friction drag coefficient \( C_{Df} \) with the reference. Although slight difference can be noticed in \( C_p \) profile, the present results are found to agree quantitatively well with the reference (Yoon et al., 2010), which confirms the capability of the present method. The slight difference may be due to a difference in the treatment of the corner singularity, which is not always explained in literature. In the present method, more than one normal lines can be defined for one ghost point near the corners; in the present simulation, however, we simply adopted one of them. The treatment of the corner singularity itself should be an important subject for
Table 1: Comparison of the results on the flow around a circular cylinder at Re = 40 (parameters $a$, $b$, $L_w$ and $\theta_s$ are defined in Fig. 5).

| Method                        | $a$  | $b$  | $L_w$ | $\theta_s$ | $C_D$ |
|-------------------------------|------|------|-------|------------|-------|
| Present study                 | 0.73 | 0.59 | 2.28  | 53.1       | 1.55  |
| Linnick and Fasel (2005)      | 0.72 | 0.60 | 2.28  | 53.6       | 1.54  |
| Taira and Colonius (2007)     | 0.73 | 0.60 | 2.30  | 53.7       | 1.54  |
| Coutanceau and Bouard (1977)  | 0.76 | 0.59 | 2.13  |            | 53.8  |

Table 2: Comparison of the results on the flow around a circular cylinder at Re = 100.

| Method                        | $C_D$ | $C_{L_{rms}}$ | St  |
|-------------------------------|-------|---------------|-----|
| Present study                 | 1.43  | 0.29          | 0.160 |
| Tseng and Ferziger (2003)     | 1.42  | 0.29          | 0.164 |
| Lai and Peskin (2000)         | 1.44  | 0.32          | 0.165 |
| Kim et al. (2001)             | 1.33  | 0.32          | 0.165 |
| Park et al. (1998)            | Boundary-fitted (C-grid) | 1.33 | 0.33 | 0.165 |
| Naito and Fukagata (2012)     | Boundary-fitted (O-grid) | 1.33 | 0.23 | 0.165 |
| Norberg (2003)                | Empirical correlation | –   | 0.23 | 0.164 |

Fig. 7: "Streamlines" for a square cylinder at Re = 20 (steady flow regime): (a) $\theta = 0^\circ$; (b) $\theta = 29.7^\circ$.

improving the immersed boundary methods, and it should be left as a future work.

5. Conclusions

We have proposed a unified interpolation stencil that is used for the ghost-cell immersed boundary (IB) method to satisfy wall boundary conditions in Cartesian-based numerical simulation of fluid flow with complex boundaries. The key technique in the present method is to consider a one-dimensional interpolation along the line normal to the surface. Two different interpolation stencils are proposed and examined by a benchmark problem on the flow between two concentric cylinders. It has been demonstrated that the compact three-point stencil (Stencil 2) has the second order convergence rate, while the four-point stencils did not attain the expected convergence rate.

The numerical procedure of the proposed scheme with Stencil 2 can be summarized as follows:

1. find the image point by mirror-reflecting the ghost point (Fig. 1(a));
2. find the intersection points along the normal line (Fig. 1(b));
3. compute the quantity on the intersection points using the linear interpolation from the two neighboring grid points (Eq. (6));
4. compute the quantity on the image point using the linear interpolation (Eq. (7) or (8));
5. impose the value on the ghost point (Eq. (4)).

This unified procedure can be used everywhere as far as the boundary is smooth. Numerical instability due to the interpolation, which is often problematic in the conventional discrete-forcing IB methods, is avoided by the present definition of
Table 3: Recirculation length $L_w$ for a square cylinder with different inclination angles $\theta$ at Re = 20.

| Method                        | $\theta = 0^\circ$ | $\theta = 29.7^\circ$ |
|-------------------------------|---------------------|------------------------|
| Present study                 | 1.79                | 1.49                   |
| Yoon et al. (2010)            | 1.8                 | 1.44                   |
| Sharma and Eswaran (2004)     | 1.83                | -                      |

Fig. 8: Instantaneous "streamlines" around a square cylinder at Re = 100.

Fig. 9: Instantaneous "streamlines" around an inclined square cylinder (Re = 130, $\theta = 34.8^\circ$).

The proposed scheme with Stencil 2 was used to simulate the flow past different obstacles, which are typical of engineering-relevant applications. First, simulation of the flow around the circular cylinder at two different Reynolds number were carried out. The computed values of the wake parameter and the lift and drag coefficients were found to be in good agreement with the literature. To verify the ability of the proposed scheme in simulating flows near sharp edges, the flow over an inclined square cylinder was studied as well. The results of our simulation agree well with literature.

Through these numerical studies, we have demonstrated that the proposed scheme with Stencil 2, despite its simplicity, can be used for simulating the flow past arbitrary geometries with an accuracy comparable to the existing schemes that require more complicated interpolation procedures.

Although we only studied fixed boundaries in the present study, the proposed method should be applicable to moving boundary problems with a minor modification such that the procedure of finding the ghost points is repeated every time step. However, it is also presumed that the time step should be carefully chosen to avoid numerical instabilities due to the movement of boundary points. Investigation on the applicability of the present method for moving boundary problems is therefore left as a future work.

[DOI: 10.1299/jfst.2017jfst0011] © 2017 The Japan Society of Mechanical Engineers
Table 4: Drag coefficient for a square cylinder at different Reynolds numbers, Re (θ = 0°).

| Method                           | Re = 20 | Re = 60 | Re = 100 |
|----------------------------------|---------|---------|----------|
| Present study Ghost-cell IBM     | 2.16    | 1.56    | 1.45     |
| Yoon et al. (2010) Discrete-forcing IBM of Kim et al. (2001) | 2.3     | 1.54    | 1.46     |
| Sharma and Eswaran (2004) Body-fitted (Cartesian) | 2.39    | 1.64    | 1.51     |
| Okajima et al. (1997) Body-fitted (Cartesian) | 2.1     | –       | –        |
| Sohankar et al. (1998) Body-fitted (Cartesian) | –       | 1.62    | –        |
| Anzai et al. (2017) Body-fitted (Cartesian) | –       | –       | 1.51     |

Table 5: The mean pressure drag coefficient $C_{DP}$ and the mean friction drag coefficient $C_{DF}$ for an inclined square cylinder (θ = 29.7° at Re = 100).

| Method                           | $C_{DP}$ | $C_{DF}$ |
|----------------------------------|---------|---------|
| Present study Ghost-cell IBM     | 1.50    | 0.21    |
| Yoon et al. (2010) Discrete-forcing IBM of Kim et al. (2001) | 1.42    | 0.20    |

Acknowledgments

The authors are grateful to Drs. Shinnosuke Obi and Keita Ando (Keio University) for fruitful discussion. H. Kor and M. Badri Ghomizad acknowledge the financial supports from Top Global University MEXT Scholarship and Otsuka Toshimi Foundation.

References

Anzai, Y., Fukagata, K., Meliga, P., Boujo, E., and Gallaire, F., Numerical simulation and sensitivity analysis of a low-Reynolds number flow around a square cylinder controlled using plasma actuators. Physical Review Fluids, Vol. 2, Paper No. 043901 (2017).

Balaras, E., Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. Computers and Fluids, Vol. 33 (2004), pp. 375–404.

Burden, R. and Faires, J., Numerical Analysis, 8th edition. Thomson Brookes/Cole (2005).

Coutanceau, M. and Bouard, R., Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. Journal of Fluid Mechanics, Vol. 79 (1977), pp. 231–256.

Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., Combined immersed boundary finite-difference methods for three-dimensional complex flow simulations. Journal of Computational Physics, Vol. 161 (2000), pp. 35–60.

Fedkiw, R., Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with plasma actuators. Journal of Computational Physics, Vol. 175 (2002), pp. 200–224.

Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). Journal of Computational Physics, Vol. 152 (1999), pp. 457–492.
Fukagata, K., Kasagi, N., and Koumoutsakos, P., A theoretical prediction of friction drag reduction in turbulent flow by superhydrophobic surfaces. Physics of Fluids, Vol. 18 (2006), Paper No. 051703.

Ghias, R., Mittal, R., and Dong, H., A sharp interface immersed boundary method for viscous compressible flows. Journal of Computational Physics, Vol. 225 (2007), pp. 528–553.

Gibou, F., Fedkiw, R., Cheng, L., and Kang, M., A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. Journal of Computational Physics, Vol. 176 (2002), pp. 205–227.

Goldstein, D., Handler, R., and Sirovich, L., Modeling a no-slip flow boundary with an external force field. Journal of Computational Physics, Vol. 105 (1993), pp. 354–366.

Iaccarino, G. and Verzicco, R., Immersed boundary technique for turbulent flow simulations. Applied Mechanics Reviews, Vol. 56 (2003), pp. 331–347.

Ikeno, T. and Kajishima, T., Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations. Journal of Computational Physics, Vol. 226 (2007), pp. 1485–1508.

Kajishima, T., Finite-difference method for convective terms using non-uniform grid. Transactions of the JSME Series B, Vol. 65, No. 633 (1999), pp. 103–108. (in Japanese)

Kametani, Y. and Fukagata, K., Direct numerical simulation of spatially developing turbulent boundary layer with uniform blowing or suction. Journal of Fluid Mechanics, Vol. 681 (2011), pp. 154–172.

Kim, J., Kim, D., and Choi, H., An immersed boundary finite-volume method for simulations of flows in complex geometries. Journal of Computational Physics, Vol. 171 (2001), pp. 132–150.

Lai, M. and Peskin, C., An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. Journal of Computational Physics, Vol. 160 (2000), pp. 705–719.

Linnick, M. and Fasel, H., A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. Journal of Computational Physics, Vol. 204 (2005), pp. 157–192.

Majumdar, S., Iaccarino, G., and Durbin, R., RANS solvers with adaptive structured boundary non-conforming grids. Center for Turbulence Research Annual Research briefs (2001), pp. 353–366.

Marella, S., Krishnan, S., Liu, H., and Udaykumar, H., Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. Journal of Computational Physics, Vol. 210 (2005), pp. 1–31.

Mittal, R. and Iaccarino, G., Immersed boundary methods. Annual Review of Fluid Mechanics, Vol. 37 (2005), pp. 239–261.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A., and Von Leerbeek, A., A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. Journal of Computational Physics, Vol. 227 (2008), pp. 4825–4852.

Mohd-Yusof, J., Combined immersed boundary/B-spline methods for simulations of flows in complex geometries. NASA Ames Research Center/Stanford University Center for Turbulence Research, Stanford, CA: (1997), pp. 317–327.

Naito, H. and Fukagata, K., Numerical simulation of flow around a circular cylinder having porous surface. Physical of Fluids, Vol. 24 (2012), Paper No. 117102.

Nosberg, C., Fluctuating lift on a circular cylinder: review and new measurements Journal of Fluids and Structures, Vol. 17 (2003), pp. 57–96.

Okajima, A., Yi, D., Sakuda, A., and Nakano, T., Numerical study of blockage effects on aerodynamic characteristics of an oscillating rectangular cylinder. Journal of Wind Engineering and Industrial Aerodynamics, Vol. 67–68 (1997), pp. 91–102.

Park, J., Kwon, K., and Choi, H., Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. KSME International Journal, Vol. 12 (1998), pp. 1200–1205.

Peskin, C., Flow patterns around the heart valves. Journal of Computational Physics, Vol. 10 (1972), pp. 252–271.

Rai, M. and Moin, P., Direct numerical simulations of turbulent flow using finite difference scheme. Journal of Computational Physics, Vol. 96 (1991), pp. 15–53.

Saiki, E. and Biringen, S., Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. Journal of Computational Physics, Vol. 123 (1996), pp. 450–465.

Sharma, A. and Eswaran, V., Heat and fluid across a square cylinder in the two-dimensional laminar flow regime. Numerical Heat Transfer A, Vol. 45 (2004), pp. 247–269.

Sohankar, A., Norberg, C., and Davidson, L., Low-Reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition. International Journal for Numerical Methods in Fluids, Vol. 26 (1998), pp. 39–56.
Taira, K. and Colonius, T., The immersed boundary method: a projection approach. Journal of Computational Physics, Vol. 225 (2007), pp. 2118–2137.

Tseng, Y. and Ferziger, J., A ghost-cell immersed boundary method for flow in complex geometry. Journal of Computational Physics, Vol. 192 (2003), pp. 593–623.

Udaykumar, H., Kan, H.-C., Shyy, W., and Tran-Son-Tay, R., Multiphase dynamics in arbitrary geometries on fixed Cartesian grids. Journal of Computational Physics, Vol. 137 (1997), pp. 366–405.

Udaykumar, H., Mittal, R., and Shyy, W., Computational of solid-liquid phase fronts in the sharp interface limit on fixed grids. Journal of Computational Physics, Vol. 153 (1999), pp. 535–574.

Udaykumar, H., Shyy, W., and Rao, M., A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries. International Journal for Numerical Methods in Fluids, Vol. 22 (1996), pp. 691–712.

Yang, J. and Balaras, E., An embedded-boundary formulation for large-eddy simulation of turbulent flows with moving boundaries. Journal of Computational Physics, Vol. 215 (2006), pp. 12–40.

Ye, T., Mittal, R., Udaykumar, H., and Shyy, W., An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. Journal of Computational Physics, Vol. 156 (1999), pp. 209–240.

Yoon, D.-H., Yang, K.-S., and Choi, C.-B., Flow past a square cylinder with angle of incidence. Physics of Fluids, Vol. 22 (2010), Paper No. 043603.

You, D., Wang, M., Mittal, R., and Moin, P., Study of rotor tip-clearance flow using large eddy simulation. AIAA Paper (2003), Papar No. AIAA 2003-0838.