Journal of Computational Physics 228 (2009) 6753–6772

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method

Seongwon Kang *, Gianluca Iaccarino, Frank Ham, Parviz Moin

Center for Turbulence Research, Stanford University, CA 94305, USA

ARTICLE INFO

Article history:
Received 13 August 2008
Received in revised form 7 May 2009
Accepted 20 May 2009
Available online 3 June 2009

Keywords:
Immersed boundary
Turbulent flow
Wall-pressure fluctuation
Pressure decoupling constraint
LES
DNS
Space–time correlation

ABSTRACT

The objective of this paper is to assess the accuracy and efficiency of the immersed boundary (IB) method to predict the wall pressure fluctuations in turbulent flows, where the flow dynamics in the near-wall region is fundamental to correctly predict the overall flow. The present approach achieves sufficient accuracy at the immersed boundary and overcomes deficiencies in previous IB methods by introducing additional constraints – a compatibility for the interpolated velocity boundary condition related to mass conservation and the formal decoupling of the pressure on this surfaces. The immersed boundary-approximated domain method (IB-ADM) developed in the present study satisfies these conditions with an inexpensive computational overhead. The IB-ADM correctly predicts the near-wall velocity, pressure and scalar fields in several example problems, including flows around a very thin solid object for which incorrect results were obtained with previous IB methods. In order to have sufficient near-wall mesh resolution for LES and DNS computations, the present approach uses local mesh refinement. The present method has been also successfully applied to computation of the wall-pressure space–time correlation in DNS of turbulent channel flow on grids not aligned with the boundaries. When applied to a turbulent flow around an airfoil, the computed flow statistics – the mean/RMS flow field and power spectra of the wall pressure – are in good agreement with experiment.

1. Introduction

In industrial applications of computational fluid dynamics (CFD), it is not uncommon to devote more time to grid generation than to the flow solution. The immersed boundary (IB) method has emerged as an alternative, since it can reduce the difficulty and time requirement of mesh generation. The IB method does not require the computational mesh to conform to the physical boundaries. Instead, the solution algorithm is locally modified to enforce the desired boundary conditions. This feature is attractive for very complex geometries, because a very simple mesh structure, such as a Cartesian mesh, can be used. The IB method is also advantageous for problems associated with moving geometries. Another advantage presents itself in multi-phase or multi-material problems, where the interface between different materials can be regarded as an immersed boundary.

The IB method was first introduced by Peskin [41] for computing blood flow in the cardiovascular system; subsequently, there have been numerous efforts to enhance its accuracy and stability. Readers can refer to articles by Mittal and Iaccarino [37] and Iaccarino and Verzicco [23] for further information on the previous studies. So far, IB methods have been applied to a wide range of applications: compressible flows [14,36], particulate flows [51,58], micro-scale flows [2], interaction with solid bodies ([15,61], among others), multi-phase flows [12], conjugate heat transfer [22,59], environmental flows [45],

* Corresponding author.
E-mail address: seongwon@stanford.edu (S. Kang).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.05.036

bio-fluids [10], etc. However, a relatively small number of studies have been published for turbulent flows with high Reynolds numbers.

Although many previous studies have reported theoretical improvements and encouraging results, a few issues remain to be addressed. In the direct forcing techniques ([38,9,47], among others), for example, issues related to enforcing mass conservation have been raised. In some approaches, mass conservation at the IB is satisfied by the velocity fields both in the fluid and solid regions. In this case, the (unphysical) velocity field in the solid becomes important because it affects the pressure and velocity distribution through the velocity divergence across the IB. This issue can become more serious in the reconstruction methods (e.g., [9,16,15]), since treatment for the velocities at the first grid points into the solid region is notionally undefined. Iaccarino and Verzicco [23] observed that different treatments of the solid velocity field do not affect the flow field in the fluid region in their numerical experiments; this is tested in the present study. Further issues on the general IB method and the reconstruction method will also be discussed later.

In the literature, several IB methods have been applied to LES/DNS of various turbulent flows: flow over a wavy wall [47,33,57], flow over a sphere [60], flow inside a piston engine [53], flow in an impeller-stirred tank [52], flow over a mannequin [6], inclined channel [24], flow in a nuclear rod-bundle [24], flow over a building [45], stator–rotor interaction [49], etc. Most of these studies have focused on the flow field away from the wall. In some cases, the distribution of the time-averaged pressure is shown, but there is no previous study which presents high-order statistics of the wall pressure at the IB. Thus, it is very important to assess the ability of the IB method to predict wall variables correctly in turbulent simulations.

Another important issue arising in LES/DNS of turbulent flows is generation of a sufficiently resolved mesh. While the combination of a Cartesian mesh and the IB method has attractive features, it has a limitation in practical cases, especially in complex turbulent flows. The Kolmogorov scale is proportional to $Re^{-3/4}$; thus, the smallest mesh size decreases as the Reynolds number increases. In order to reduce the total number of mesh points, the mesh size in the far-field needs to remain large. Thus, the ratio of the largest and smallest mesh sizes increases with the Reynolds number in turbulent simulations; as a result, a flow with a high Reynolds number can not be handled efficiently with a purely Cartesian mesh. In the present study, the local mesh refinement technique is employed. Fig. 1 shows examples of a locally refined mesh and body-fitted mesh. Two geometric features of a locally refined mesh are (*i*) hanging nodes, which enable intensive local mesh refinement, and (*ii*) Cartesian hexahedral mesh, which makes the development of an automatic mesh-generation algorithm easy. The present local mesh refinement is described in greater details in Section 2.1.

The objective of the present study is to assess the effectiveness of the IB method to correctly predict the wall-pressure fluctuations in turbulent flows. In order to accomplish this, the pressure fluctuations are computed for a channel flow and a flow around an airfoil. Then, wall pressure RMS fluctuations as well as the spectra are computed and compared with previous studies and experimental data. In the next section, the Navier–Stokes solver and the local mesh refinement technique are described. In Section 3, various aspects of the IB method are discussed. In order to resolve identified issues, revisions to existing IB methods are devised. In Sections 4 and 5, results of the verification and validation study are presented, followed by the conclusion in Section 6.

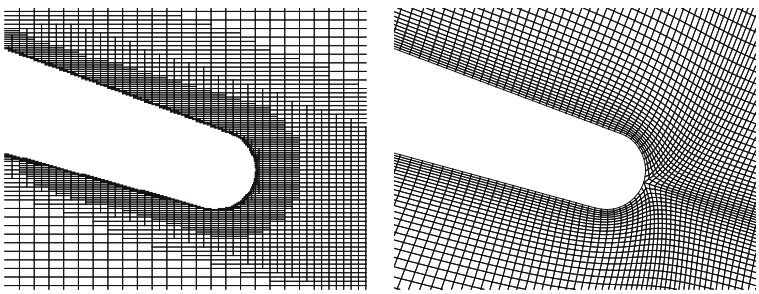
2. Numerical methods

2.1. Description of the Navier–Stokes solvers

The Navier–Stokes equation and the continuity equation for unsteady incompressible viscous flow in Cartesian coordinates are (in non-dimensional form):

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{1}$$

$$\frac{\partial u_i}{\partial x_i}=0, \tag{2}$$



(a) A locally refined mesh (b) A body-fitted mesh

Fig. 1. Examples of a locally refined mesh and body-fitted mesh.

where t is the time, $Re = U_0 L/\nu$ is the Reynolds number, U_0 is a reference velocity, L is a reference length, ν is the kinematic viscosity, u_i is the velocity component in the i direction, p is the pressure non-dimensionalized by ρU_0^2, and ρ is the density.

In the present study, the solution of the Navier–Stokes equations (1) and (2) is obtained using a fully implicit LES solver based on an unstructured collocated mesh (CDP [1]). In this solver, the time-staggered scheme of Pierce [42] is employed. The discretized momentum equation at $(n+1/2)$-th time step is written as:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}+\frac{1}{2}\left[\frac{\partial u_j^{n+\frac{1}{2}}u_i^{n+1}}{\partial x_j}+\frac{\partial u_j^{n+\frac{1}{2}}u_i^{n}}{\partial x_j}\right]=-\frac{\partial p^{n+\frac{1}{2}}}{\partial x_i}+\frac{1}{2Re}\left[\frac{\partial^2 u_i^{n+1}}{\partial x_j\partial x_j}+\frac{\partial^2 u_i^{n}}{\partial x_j\partial x_j}\right]+\mathcal{O}(\Delta t^2). \tag{3}$$

In CDP, the spatial derivatives are computed using a finite volume method (FVM) for node-based collocated mesh. The flux at the face of a control volume is evaluated by applying a second-order interpolation and mid-point rule for numerical integration. Further details about CDP are available in [18,17]. It is obviously more time-consuming to use an unstructured mesh solver with the IB method than to develop an algorithm tailored for structured meshes. However, locally refined meshes (i.e., with hanging nodes) do not require special treatment, and the present approach can easily support load-balanced computations in a massively parallel computing environment. The present algorithm treats cells around hanging nodes as polyhedral elements. It achieves second-order accuracy for Cartesian meshes, but the accuracy is reduced locally to first order at hanging nodes. Even in this case, the L-2 norm of the error is observed to be the second order, since the fraction of hanging nodes over total nodes is never large. To solve the discretized momentum (Eq. (3)) and continuity equations (Eq. (2)) efficiently, a variant of the fractional-step method ([29,42]) is employed.

Locally refined mesh, sometimes referred to as adaptive mesh refinement (AMR) has been discussed in several previous studies ([4,1,3,19,21], among others) and typically constructed using a hierarchical tree structure (e.g., [4,8]) to record the connectivity of mesh elements at different refinement levels. On the other hand, Aftosmis et al. [1] and Ham et al. [19] used a fully unstructured approach by handling the elements with hanging nodes as polyhedra. Although the unstructured approach requires more memory than the tree structure, it has the advantage of easily allowing anisotropic (directional) refinement. In the present study, the unstructured approach is naturally employed by the flow solver. Based on the features noted in the previous studies, locally refined mesh structure provides (*i*) easy control of the local resolution; (*ii*) a fast turn-around time with respect to unstructured mesh generation for complex geometries; (*iii*) development of automatic mesh-generation strategies.

3. Treatment of immersed boundary

In this section, the mathematical formulation of the IB method is derived. As a starting point of the previous approaches, the standard reconstruction method is presented. In order to address the limitations of this approach, a new IB treatment is devised and compared with the original method.

3.1. Basic equations of the IB method

In order to handle solid objects immersed in the fluid, the governing equations, Eqs. (1) and (2) need to be modified to enforce boundary conditions at the fluid-solid interface in the physical domain. From the work of Peskin [41], enforcing the velocity boundary condition is expressed via an additional forcing term to the original momentum equation:

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j\partial x_j}+f_i, \tag{4}$$

$$f_i = 0 \quad \text{in } \Omega_{fluid}$$
$$= F_i \quad \text{in } \Omega_{solid} \text{ or } \Gamma_{IB},$$

$$u_i = u_{i,IB} \quad \text{at } \Gamma_{IB}, \tag{5}$$

$$\frac{\partial u_i}{\partial x_i}=0 \quad \text{in } \Omega_{fluid}+\Omega_{solid}, \tag{6}$$

where Ω denotes a domain of fluid or solid. Γ_{IB} denotes the interface between Ω_{solid} and Ω_{fluid}. Enforcing the continuity equation for Ω_{solid} is discussed later in this study. The forcing term F_i is determined such that the velocity boundary condition is satisfied at Γ_{IB}. Some IB methods have a confined non-zero F_i on Γ_{IB} by using an approximated, smooth Dirac delta function. In practical terms in the implicit forcing technique, the forcing term is never computed (e.g., [9,47]); however, the mathematical formulation for these methods still can be derived with the forcing term [37], so that Eq. (4) does not loose generality. This set of governing equations has been used in most of the previous studies and is referred to as the standard formulation of the IB method.

While Eqs. (4)–(6) have been used widely in IB methods, there are a few issues worth clarifying. The first is related to the pressure accuracy at the boundary Γ_{IB}. By using Eqs. (4)–(6) for determining the pressure, it is assumed that the unmodified

[1] CDP is named after Charles David Pierce (1969–2002).

governing equations are satisfied in Ω_{fluid}, such that the pressure gradient satisfies the following relationship on the fluid side of Γ_{IB} (namely, $\Gamma_{IB-fluid}$):

$$\left.\frac{\partial p}{\partial x_i}\right|_{\Gamma_{IB-fluid}} = \left[-\frac{\partial u_i^{fluid}}{\partial t} - \frac{\partial (u_i u_j)^{fluid}}{\partial x_j} + \frac{1}{Re}\frac{\partial^2 u_i^{fluid}}{\partial x_j \partial x_j}\right]_{\Gamma_{IB-fluid}} \quad \text{with} \quad \frac{\partial u_i^{fluid}}{\partial x_i} = 0, \tag{7}$$

where u_i^{fluid} denotes the velocity field in Ω_{fluid} including the (eventual) boundary velocity. By solving Eqs. (4)–(6), however, the resulting pressure gradient satisfies:

$$\left.\frac{\partial p}{\partial x_i}\right|_{\Gamma_{IB}} = \left[-\frac{\partial u_i}{\partial t} - \frac{\partial u_i u_j}{\partial x_j} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i\right]_{\Gamma_{IB}} \quad \text{with } \frac{\partial u_i}{\partial x_i} = 0. \tag{8}$$

In practice, the terms in Eq. (8) are evaluated using velocity fields in both Ω_{fluid} and Ω_{solid}. There is no evidence that the difference between two pressure gradients in Eqs. (7) and (8) is always negligible, unless f_i is zero at Γ_{IB} or the velocity gradient is discontinuous.

Another example demonstrating the need for an additional treatment of the pressure is the interface problem shown in Fig. 2. There is a thin solid boundary (interface) with virtually zero thickness between two channels with steady laminar flows in the opposite directions. The pressure at the boundary (Γ_{IB}) between two channels increases in the x-direction for the lower channel and decreases for the upper channel. This example is inspired by a flow around a thin airfoil where the pressure distribution on the pressure and suction sides must remain decoupled. For the very thin interface in Fig. 2, the solution requires a discontinuous pressure profile across the interface. This can be achieved by separating the discretized operators across the interface. For example, the cut-cell method [50] modifies the operator near the IB so that the solution on one side of the interface is not affected by the other side. This is equivalent to enforcing Eq. (7) at the boundary (Γ_{IB}) on each side of the interface, thus resulting in solutions across Γ_{IB} independent of each other. More details are given in Section 3.4.

This decoupling process allows discontinuous solutions across the interface and is similar to the "Jump condition" used in the immersed interface method ([34,56], among others) and the ghost fluid method [11]. In the present study, Eq. (7) will be referred to as the decoupling constraint for the pressure. This constraint is satisfied when the flow field in the fluid domain is decoupled from other physical domains. In other words, both the momentum equation and the continuity equation are satisfied using flow variables in the fluid domain and the extrapolated variables inside the solid body. Then, Eq. (7) is automatically satisfied.

The second issue is related to the velocity boundary condition at Γ_{IB} (Eq. (5)). After discretization, the positions where we want to enforce Eq. (5) are not necessarily located on the grid points. A relationship that approximates Eq. (5) is then necessary. One method of accomplishing this is to "spread" the effect of Eq. (5) to nearby velocity points in Ω_{fluid} by modifying their discrete momentum equation, e.g. Fadlun et al. [9]. Another widely used method is to satisfy Eq. (5) using interpolation of neighboring velocity points:

$$\sum_{nb} w_{nb} u_{i,nb} = u_{i,IB} \quad \text{at } \Gamma_{IB}, \tag{9}$$

where w_{nb} is the interpolation coefficient and nb denotes the index of neighboring points. A linear interpolation has been employed in several previous studies. In Mohd-Yusof [38], this method was described as mirroring the velocity field across Γ_{IB}. An example is shown in Fig. 3(a) where the velocity field across the IB is mirrored such that the no-slip boundary condition is satisfied using a linear interpolation.

Methods based on mirroring satisfy the velocity boundary condition with accuracy of the interpolation method. In practice, the accuracy may be lower because of incompatibility with the continuity equation Eq. (6). This issue was previously reported by Kim et al. [28] and schematically represented for a simple case in Fig. 3(b). Fig. 4 shows tangential and normal coordinates local to the immersed boundary Γ_{IB}. Assume that $u_\tau(\tau,\eta)$ and $u_\eta(\tau,\eta)$ are the tangential and normal velocity fields in Ω_{fluid}. The velocity field in the mirrored region Ω_{mirror} can be expressed as:

$$\begin{cases} u_\tau(\tau,\eta) = -u_\tau(\tau,-\eta) + 2u_\tau(\tau,0) \\ u_\eta(\tau,\eta) = -u_\eta(\tau,-\eta) + 2u_\eta(\tau,0) \end{cases} \quad \text{in } \Omega_{mirror}. \tag{10}$$

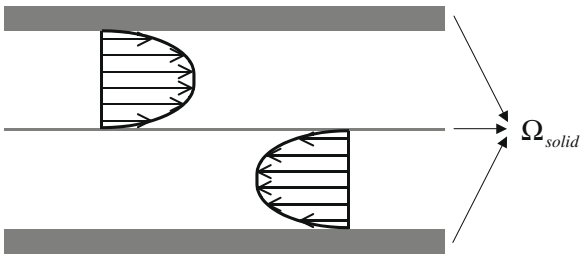


Fig. 2. A very thin solid object between two channels with flows in the opposite directions.

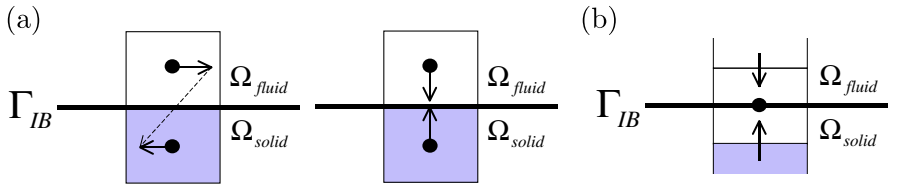


Fig. 3. Examples of velocity field mirrored across the IB: (a) collocated grid; (b) staggered grid. A square denotes a grid.

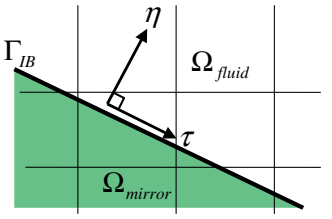


Fig. 4. A local coordinate at the immersed boundary.

We assume that the continuity equation is satisfied in Ω_{fluid}:

$$\frac{\partial}{\partial\tau}u_\tau(\tau,\eta)+\frac{\partial}{\partial\eta}u_\eta(\tau,\eta)=0 \quad \text{in } \Omega_{fluid}.$$

For simplicity, assume the no-slip condition ($u_\tau(\tau,0)=u_\eta(\tau,0)=0$). Then, it is easily proved that

$$\frac{\partial}{\partial\tau}u_\tau(\tau,\eta)+\frac{\partial}{\partial\eta}u_\eta(\tau,\eta)=-\frac{\partial}{\partial\tau}u_\tau(\tau,-\eta)-\frac{\partial}{\partial\eta}u_\eta(\tau,-\eta)=-\frac{\partial u_\tau}{\partial\tau}(\tau,-\eta)+\frac{\partial u_\eta}{\partial\eta}(\tau,-\eta)\neq 0 \quad \text{in } \Omega_{mirror}.$$

The mirrored velocity field across Γ_{IB} does not satisfy the continuity equation unless $\partial u_\eta/\partial\eta=0$ near Γ_{IB}. This is a direct consequence of mass conservation. However, enforcing the mirrored velocity and continuity equation together can reduce the degree of freedom for the velocity interpolation and affect the accuracy especially on coarse meshes. The accuracy is not affected when $\partial u_\eta/\partial\eta\approx 0$ is satisfied, e.g., streamlines are parallel near Γ_{IB}. In the worst situation, however, the accuracy is reduced to the first order. For example, the case in Fig. 3(b) requires the y-velocity components to be zero in order to satisfy the no-slip x-velocity boundary condition, regardless of the grid size. This situation typically occurs near a stagnation point. Kim et al. [28] addressed this issue by satisfying the continuity equation with a non-zero right-hand side (RHS) term, which reduced the velocity error at least by 60%. However, it should be emphasized that $\partial u_\eta/\partial\eta=0$ at Γ_{IB} is the correct condition, which implies that the accuracy will be recovered as the mesh is refined. Interestingly, the mirrored velocity field produces a less serious problem in the collocated mesh, as discussed in [26].

3.2. The "standard" IB reconstruction method

The starting point for the present IB method is the method of Fadlun et al. [9]. In the literature, this approach has been referred to as the reconstruction or the interpolation method. An interpolation formula replaces Eq. (4) near the IB. In addition to its simplicity, it has several advantages. Since the velocity boundary condition is enforced with implicit forcing, there is no severe limit on the time step. The velocity components from the regions across the IB are decoupled. And this approach does not rely on the mirrored velocity field in the solid region.

We used a linear interpolation method similar to Fadlun et al. [9] and Gilmanov et al. [16]. A linear interpolation formula is used to determine a velocity component that is in the fluid region and has one of the neighboring points in the discretization stencil outside of the fluid region. In Gilmanov and Sotiropoulos [15] and Choi et al. [6], higher order interpolation methods were used. Choi et al. stated that a power-law based interpolation is better suited to high Reynolds number flows than a linear interpolation. For the velocity, we considered the following interpolation method:

$$u_{i,c}=\sum_{nb}w_{i,nb}u_{i,nb}+w_{IB}u_{i,IB}, \tag{11}$$

where w_{nb} is the interpolation coefficient, nb denotes the index of neighboring points, and subscript IB denotes the point on the IB that is the boundary-normal projection of the velocity node c. Assuming a local coordinate whose center is located at a point on the IB, we can restate the linear interpolation method as:

$$u_i(x_1,x_2,x_3)=a_{1,i}x_1+a_{2,i}x_2+a_{3,i}x_3+u_{i,IB}, \tag{12}$$

where $(a_{1,i},a_{2,i},a_{3,i})$ are coefficients determined by the local IB geometry and velocity.

A special treatment for the mass conservation was not mentioned in Fadlun et al. [9]. Thus, we assumed that the continuity equation is enforced in the entire computational domain, which results in no modification to the mass conservation procedure. We refer to the combination of the linear interpolation method and mass conservation enforced in the entire domain as the standard reconstruction method (SRM).

One concern of SRM is that the treatment of the velocities in the solid region which contribute to mass conservation in the grid cells crossed by the IB is notionally undefined. Without any special treatment, the original method for enforcing mass conservation results in coupling between the solutions across the IB via discretized operators, which disobeys the pressure decoupling constraint (Eq. (7)). In our numerical tests with the flow with a very thin wall in Section 4.2 shows that the SRM fails to predict the flow fields correctly.

In order to address this issue, a few options for enforcing mass conservation near the IB are introduced and compared in the next section. Then, the most viable one is selected and discussed further in the following sections.

3.3. Treatment of mass conservation near immersed boundary

Fig. 5 shows different grid configurations near the IB. The shaded area is a control volume where mass is conserved in each configuration.

In Fig. 5(a), mass is conserved for every control volume in the domain as it is without the IB method. This has been a standard method for several IB methods with Eqs. (4)–(6) as their governing equations. For a control volume crossed by Γ_{IB}, mass conservation is described by velocity values in both fluid (Ω_{fluid}) and solid (Ω_{solid}) domains, which implies that the pressure decoupling constraint Eq. (7) may not be satisfied. In the present study, this method is referred to as standard mass conservation.

In Fig. 5(b), rectangular control volumes (CVs) crossed by Γ_{IB} are divided into the flow region where solutions to the Navier–Stokes equations are desired, and boundary regions (i.e., white area in the figure) where no solution is needed. Γ_{IB} separates these two regions. Mass conservation is satisfied for the reshaped CVs formed by existing CV faces in Ω_{fluid} and Γ_{IB}. Similar methods have been used by cut-cell approaches ([48,50,31], among others) and Kim et al. [28]. This method satisfies the pressure decoupling constraint, however, is very complicated. As noted by Kirkpatrick et al. [31], another problem is that the matrix condition number increases significantly when the size of the reshaped CV is very small.

As a final option, in Fig. 5(c), CVs crossed by Γ_{IB} and in Ω_{solid} are excluded from the computational domain. By satisfying mass conservation only for CVs in Ω_{fluid}, this method does not suffer from the incompatibility with the mirrored velocity field. Also, it is numerically less complex than the reshaped CV approach by maintaining the original CV shape for mass conservation. Satisfying the pressure decoupling constraint with this method is discussed in greater detail in the next section.

3.4. Immersed boundary-approximated domain method

The immersed boundary-approximated domain method (IB-ADM) introduced in this section satisfies the pressure decoupling constraint, and the computational overhead is marginal. Although the IB-ADM was implemented in CDP – a solver for node-based collocated meshes, the method is applicable to both staggered and collocated meshes.

3.4.1. Enforcing mass conservation for approximated domain

In Fig. 5(c), mass conservation is satisfied for CVs in Ω_{fluid} but not for CVs in Ω_{solid} and CVs crossed by the immersed boundary Γ_{IB}. The first step of this method is to identify CVs entirely in Ω_{fluid} (the shaded area in Fig. 5(c)). A group of contiguous CVs identified in this way is referred to as an approximated domain. The boundary of the approximated domain facing the IB is then identified and referred to as an approximated boundary. Fig. 6 shows an example of the approximated domain (shaded area, Ω_a) and approximated boundary (thick gray line, Γ_a).

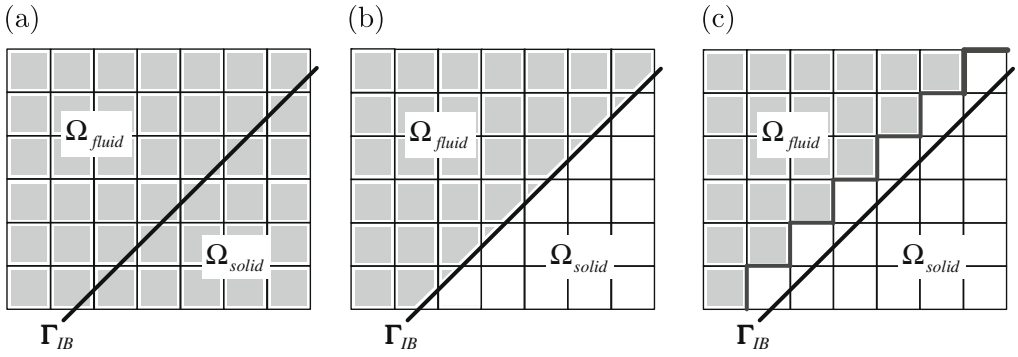


Fig. 5. Different schemes for defining control volumes (CVs) for mass conservation near IB: (a) standard scheme; (b) mass conservation for reshaped CVs; (c) mass conservation for fluid-side CVs. The shaded area denotes a control volume where mass conservation is enforced.

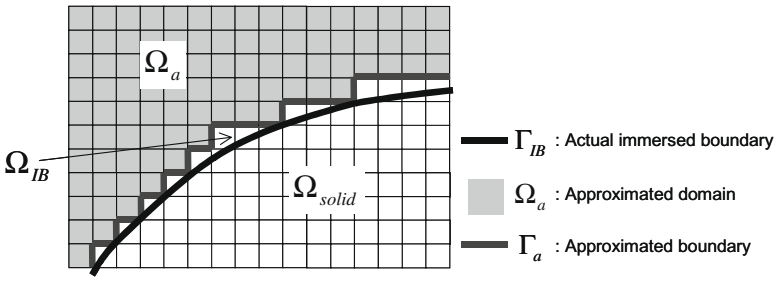


Fig. 6. Examples of the approximated domain and boundary.

The idea of the revised mass conservation method is that the discretized momentum equation and continuity equation are satisfied without modification inside the approximated domain Ω_a. We then need to provide the boundary condition to the approximated domain. The velocity boundary condition at Γ_a is reconstructed using one of the interpolation methods. Imposing boundary conditions at Γ_a results in all flow variables (velocity, pressure and scalar) at CVs in Ω_{solid} and CVs crossed by the IB being excluded from the computation. Fig. 7 shows the interpolated velocity components on Γ_a for the staggered, CV-collocated and node-collocated arrangements of the velocity variables.

Γ_a is a grid-based (stair-step) approximation to Γ_{IB} in the fluid domain. The domain between Γ_a and Γ_{IB} is composed of reshaped CVs in the fluid domain and denoted by Ω_{IB}. Our strategy to satisfy mass conservation only in the approximated domain Ω_a leads to an additional condition of global mass conservation in Ω_{IB}:

$$\int_{\Gamma_a} u \cdot dA = \int_{\Gamma_{IB}} u \cdot dA, \tag{13}$$

where dA is directed outward from Ω_a. In order to avoid complexity of satisfying mass conservation for the reshaped CVs, mass conservation between Ω_a and Γ_{IB} (Eq. (13)) is alternatively used as an additional constraint; this is satisfied by modifying the interpolation method for the velocity. A least-square method is used to account for the constraint of mass conservation.

For simplicity, the basic linear interpolation method Eq. (11) is considered. The continuous form Eq. (12) is rewritten as:

$$\vec{u}_m = \vec{a}_{1,m}x_{1,m} + \vec{a}_{2,m}x_{2,m} + \vec{a}_{3,m}x_{3,m} + \vec{u}_{IB,m} = \sum_k^3 \vec{a}_{k,m}x_{k,m} + \vec{u}_{IB,m}, \tag{14}$$

where $\vec{u}_m$ denotes a velocity vector on an approximated boundary face and m is the index of the approximated boundary face. The local coordinate system is centered at a point on the IB and the interpolated velocity is located at $(x_{1,m}, x_{2,m}, x_{3,m})$. $\vec{a}_{k,m}$ is the interpolation coefficient, and $\vec{u}_{IB,m}$ is the velocity value at the IB. Note that the repeated indices do not imply summation. A correction to the coefficient $\vec{a}_{k,m}$ is considered to determine $\vec{u}_m$:

$$\vec{u}_m = \sum_k^3 (\vec{a}_{k,m} + \delta\vec{a}_{k,m})x_{k,m} + \vec{u}_{IB,m} = \vec{u}_m^* + \sum_k^3 \delta\vec{a}_{k,m}x_{k,m}, \tag{15}$$

where $\delta\vec{a}_{k,m}$ is the correction to the interpolation coefficient, and $\vec{u}_m^*$ is the velocity from the unmodified interpolation formula Eq. (14).

A minimum amount of correction is computed while the constraint Eq. (13) is satisfied. The objective function to be minimized and the equality constraint are:

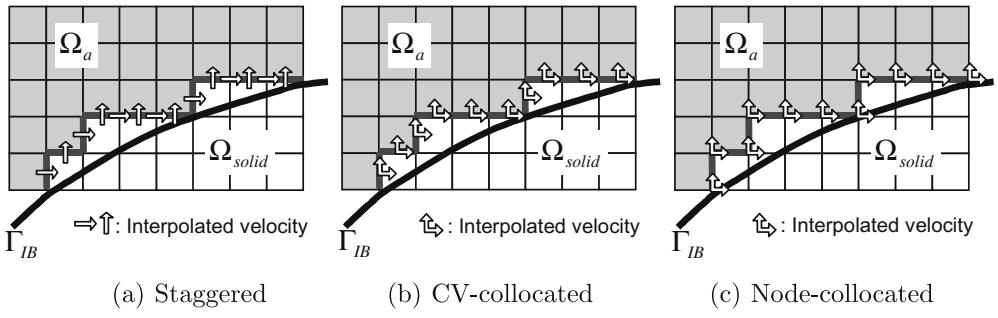


Fig. 7. Examples of the interpolated velocity components on the approximated boundary Γ_a in different arrangements of the velocity variables.

$$J = \sum_m^{N_a} \left(\frac{1}{\omega_m} \sum_k^3 \left| \vec{\delta a}_{k,m} \right|^2 \right), \tag{16}$$

$$Q_{IB} = \int_{\Gamma_{IB}} u \cdot dA = \sum_m^{N_a} \vec{u}_m \cdot \vec{A}_m, \tag{17}$$

where N_a denotes the total number of the approximated boundary faces, and ω_m is the weight factor of the face. $\omega_m = 1$ is used in the present study. $\vec{A}_m$ is the outward-normal face-area vector at the approximated boundary face where $\vec{u}_m$ is located. Q_{IB}, which is the mass flux through Γ_{IB}, is available by integrating the velocity boundary condition. Eq. (17) is a discrete version of the constraint Eq. (13). $|\vec{\delta a}_{k,m}|^2$ is minimized instead of $|\vec{\delta a}_{k,m} x_{k,m}|^2$ in order to achieve spatial convergence of the boundary condition.

In order to obtain an expression for $\vec{\delta a}_{k,m}$, the minimization problem is solved by introducing the Lagrange multiplier λ:

$$J_\lambda = \sum_m^{N_a} \left(\frac{1}{\omega_m} \sum_k^3 |\vec{\delta a}_{k,m}|^2 \right) + \lambda \left[\left\{ \sum_m^{N_a} \left(\vec{u}_m^* + \sum_k^3 \vec{\delta a}_{k,m} x_{k,m} \right) \cdot \vec{A}_m \right\} - Q_{IB} \right]. \tag{18}$$

Using $\partial J_\lambda / \vec{\delta a}_{k,m} = 0$ yields an expression for $\vec{\delta a}_{k,m}$:

$$\vec{\delta a}_{k,m} = -\frac{\lambda}{2} x_{k,m} \omega_m \vec{A}_m. \tag{19}$$

Substituting this into the constraint Eq. (17) and solving for λ yields:

$$\frac{\lambda}{2} = -\frac{Q_{IB} - \sum_m^{N_a} \vec{u}_m^* \cdot \vec{A}_m}{\sum_m^{N_a} \left(\omega_m |\vec{A}_m|^2 \sum_k^3 x_{k,m}^2 \right)}. \tag{20}$$

Letting $Q_e = Q_{IB} - \sum_m^{N_a} \vec{u}_m^* \cdot \vec{A}_m$, we can get an expression for $\vec{u}_m$:

$$\vec{u}_m = \vec{u}_m^* + \frac{Q_e \sum_k^3 x_{k,m}^2}{\sum_m^{N_a} \left(\omega_m |\vec{A}_m|^2 \sum_k^3 x_{k,m}^2 \right)} \omega_m \vec{A}_m = \vec{u}_m^* + \vec{G}_m. \tag{21}$$

Q_e is the error from $\vec{u}_m^*$ relative to the mass constraint Eq. (17). In Eq. (21), the mass flux error Q_e is corrected in $\vec{u}_m$ with the second term in the RHS. With an iterative solver for the velocity, $\vec{u}_m$ is updated every iteration using Eq. (21) until the error vanishes. Note that the size of the correction is weighted by the square of the distance between the locations of $\vec{u}_m$ and $\vec{u}_{IB,m}$. The correction term is of second-order error because of the weight and the fact that Q_e is proportional to the error of the velocity reconstruction which is second-order.

Since this correction process redistributes the error from the interpolation method, applying to a situation where the error is very localized may not be appropriate. For example, when different boundary conditions are used in different regions of Γ_a, it is recommended that the Γ_a is divided such that the boundary condition at the faces in a Γ_a is homogeneous. In case of an impinging jet, as another example, the interpolation error is locally large near the jet due to high velocity gradient. In this case, either dividing the wall (Γ_a) to close and distant regions or using $\omega_m = \sqrt{\sum_k^3 (\vec{a}_{k,m} \cdot \vec{A}_m / |\vec{A}_m|)^2}$ instead of $\omega_m = 1$ is more reasonable. Considering that this ω_m is the size of the gradient of the face-normal component of $\vec{u}_m^*$, the size of the correction is proportional to the velocity gradient as well as the distance. For simplicity, however, $\omega_m = 1$ is used in the numerical tests of the present study.

3.4.2. Implementation for the fractional step method

With the fractional step method in Section 2.1 and the IB-ADM in this section, the following relationships are used for the velocity projection at the k-th time step:

$$\nabla \cdot \nabla \phi = \frac{1}{\Delta t} \nabla \cdot \hat{u}^k \quad \text{in } \Omega_a, \tag{22}$$

$$u_i^k = \hat{u}_i^k - \Delta t \frac{\partial \phi}{\partial x_i} = \hat{u}_i^k + \mathcal{O}(\Delta t^2), \tag{23}$$

where $\phi = p^k - p^{k-1}$ and $\hat{u}_i^k$ is the intermediate velocity. Eq. (22) is solved by an algebraic multigrid solver [20]. The standard mass conservation technique produces a finite error in the velocity B.C. during the projection step. This is because $u_i = \hat{u}_i$ is not enforced at the immersed boundary Γ_{IB}. In the IB-ADM, $u_i = \hat{u}_i$ is enforced at the approximated boundary Γ_a instead of Γ_{IB}. It is very important to satisfy the pressure decoupling constraint. This assumption leads to $\partial \phi / \partial n = 0$ at Γ_a. If Γ_a is very close to Γ_{IB}, the original condition is recovered.

The interpolation formula, Eq. (15), is satisfied for $\hat{u}_i^k$, not u_i^k. Thus, when evaluated for u_i^k, Eq. (15) will have a finite error. This error is of order Δt^2 and $u_i^k = \hat{u}_i^k$ at Γ_a guarantees that u_i^k converges to the velocity B.C. as the velocity point becomes close to Γ_{IB}. It is possible to satisfy Eq. (15) exactly for u_i^k, for example by employing the method by Ikeno and Kajishima [24] and Taira and Colonius [46].

3.4.3. Interpolation method for the velocity

In the standard reconstruction method, the linear interpolation method is used for a velocity component that is in the fluid region and has one of the neighboring points in the discretization stencil unavailable in the fluid region. In the immersed boundary-approximated domain method (IB-ADM) presented in this section, however, the velocity components at Γ_a are determined from the interpolation method. Except for this, various interpolation methods can be used with the IB-ADM. Although the linear interpolation (Eq. (11)) is simplest, we found that the velocity error is slightly reduced when the effect of the pressure gradient term is included following the approach of [27]:

$$u_{i,c}^k = \sum_{nb} w_{i,nb} u_{i,nb}^k + w_{IB} u_{i,IB}^k - \Delta t \left(\left. \frac{\partial p^{k-1}}{\partial x_i} \right|_c - \sum_{nb} w_{i,nb} \left. \frac{\partial p^{k-1}}{\partial x_i} \right|_{nb} - w_{i,IB} \left. \frac{\partial p^{k-1}}{\partial x_i} \right|_{IB} \right). \tag{24}$$

Thus, this interpolation method is used in test cases presented hereafter.

4. Accuracy test and verification

In this section, results of a few two-dimensional cases are presented to show the accuracy of the present approach in comparison with the standard reconstruction method (the linear interpolation method + standard mass conservation).

4.1. Decaying vortex problem

The problem of decaying vortices in a periodic domain [7] is used to test the accuracy of the IB-ADM. The velocity and pressure fields are given as:

$$u_1(x_1, x_2, t) = -\cos \pi x_1 \, \sin \pi x_2 \, e^{-2\pi^2 t/Re}, \tag{25}$$

$$u_2(x_1, x_2, t) = \sin \pi x_1 \, \cos \pi x_2 \, e^{-2\pi^2 t/Re}, \tag{26}$$

$$p(x_1, x_2, t) = -0.25(\cos 2\pi x_1 + \cos 2\pi x_2) e^{-4\pi^2 t/Re}. \tag{27}$$

In order to test the accuracy of a passive scalar with the IB method, a manufactured solution [32] is employed. The governing equations for the passive scalar T and the solution are:

$$\frac{\partial T}{\partial t} + \frac{\partial u_1 T}{\partial x_1} + \frac{\partial u_2 T}{\partial x_2} = \alpha \left[\frac{\partial^2 T}{\partial x_1^2} + \frac{\partial^2 T}{\partial x_2^2} \right] + S(x_1, x_2, t), \tag{28}$$

$$S(x_1, x_2, t) = \frac{\pi}{2} (u_1 \sin 2\pi x_1 + u_2 \sin 2\pi x_2) e^{-4\pi^2 \alpha t}, \tag{29}$$

$$T(x_1, x_2, t) = -0.25(\cos 2\pi x_1 + \cos 2\pi x_2) e^{-4\pi^2 \alpha t}, \tag{30}$$

where α is the diffusivity of the scalar. The flow field outside the rotated square (Fig. 8(a)) is solved from the exact flow field at $t = 0$. The Reynolds number and $1/\alpha$ are set to 10^5. The Dirichlet B.C. at the IB and the linear reconstruction are used for the velocity and passive scalar. The computed flow field at $t = 0.2$ is compared with the analytic solution. The order of accuracy is investigated by computing the maximum (L_∞) error of the flow variables with different grid sizes and time steps. A set of grid sizes ($\Delta x = 0.125, 0.0625$ and 0.03125) is chosen with a set of time steps ($\Delta t = 0.02, 0.01$ and 0.005) as parameters. Both grid size and time step are halved to simultaneously test the order accuracy in space and time.

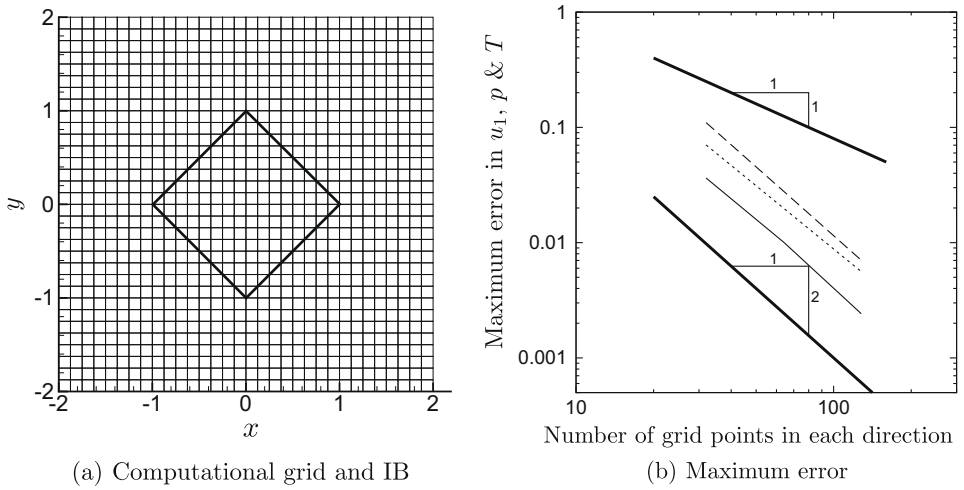
Fig. 8(b) shows the accuracy of the velocity, pressure and passive scalar for two IB geometries. The order of accuracy is shown to be second order for the velocity and pressure, and near second order ($\simeq 1.88$) for the passive scalar. The accuracy loss for the scalar does not result from the IB method but from the discretization of the advection term (WENO) that uses a stencil modified near the boundary.

4.2. A very thin wall between two channel flows in opposite directions

This test case was introduced in Section 3.1 and illustrated in Fig. 2. Steady laminar flows in opposite directions in two channels separated by a very thin rigid wall are considered. As previously mentioned, satisfying the pressure decoupling constraint ensures that the flow field in one fluid domain is not affected by the other unconnected domains. This case is therefore useful to verify whether the IB method satisfies the pressure decoupling constraint.

The Reynolds number based on the channel half-width δ and the centerline velocity U_o is 100. The computational domain size is $-10 < x/\delta < 10$ and $-3 < y/\delta < 3$, respectively. The channels are aligned in the x-direction. The lower channel extends from $-2 < y/\delta < 0 - \varepsilon$, and the upper channel from $0 + \varepsilon < y/\delta < 2$. The value of ε is set to 10^{-6}. All boundaries in the y-direction, including the thin wall at $y = 0$, are treated using the IB method. The no-slip condition is enforced at the immersed boundaries. A uniform grid spacing of $(\Delta x/\delta, \Delta y/\delta) \simeq (0.156, 0.093)$ is used for the mesh.

By using different boundary conditions in the x-direction, this problem can be cast into two scenarios that produce an identical velocity field but different pressure fields. First, if a parabolic velocity profile is imposed at the inlet ($x/\delta = -10$) and outlet ($x/\delta = 10$), the pressure drops constantly in the streamwise direction because the wall shear stress is balanced



(a) Computational grid and IB (b) Maximum error

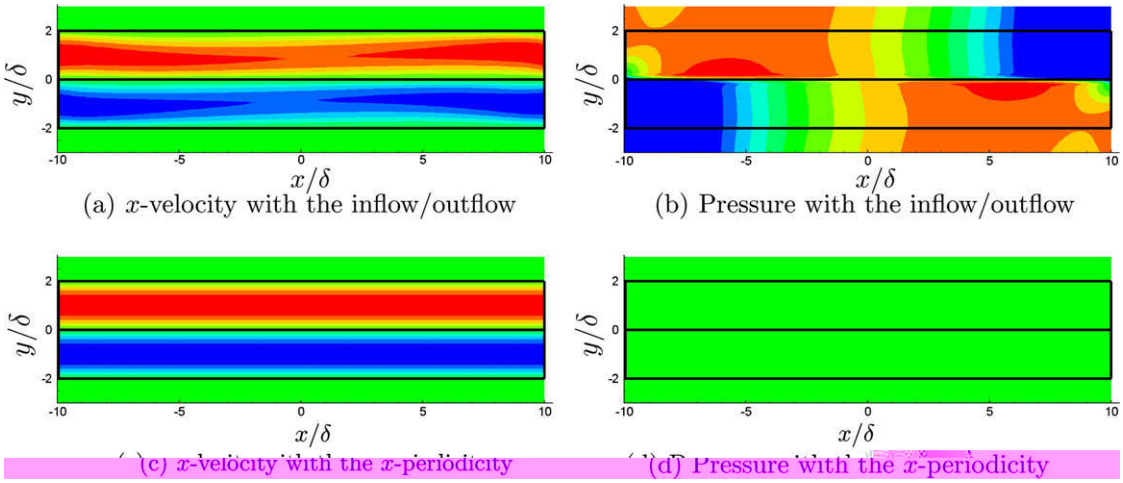
Fig. 8. Grid and IB configuration (a) and maximum error in u_1, p and T at $t = 0.2$ (b) for a decaying vortex problem: —, u_1; - - - -, p; ····, T.

by the pressure gradient. Because flow directions are opposite in two channels, a pressure profile across the thin wall shows a discontinuity which requires the pressure decoupling constraint. In the second case, the periodic B.C. is imposed in the x-direction and a constant momentum forcing term is added to the x-momentum equation. Then, the computed pressure has zero gradient in the entire domain, since the wall shear stress is balanced by the forcing term. This case therefore does not require the pressure decoupling constraint.

Fig. 9 shows contours of the x-velocity and pressure with the standard reconstruction method (SRM). With the SRM, the velocity field in the solid region is set to zero at every time step. In Fig. 9(a) and (b), the SRM produces a distorted flow field with the parabolic inflow/outflow B.C. The velocity is smeared across the two channels, especially near $x/\delta = -10$ and 10. The wall-normal velocity at the thin wall is set to zero at every time step. However, the divergence of the velocity at the thin wall is computed using the velocity fields in both channels, which results in the coupling. In Fig. 9(c) and (d), however, the SRM produces no distortion of the velocity field. Since there is no pressure gradient, the IB treatment for the velocity is sufficient to obtain the correct solution. This issue exists also for IB methods based on a discretization.

Fig. 10 shows contours of the x-velocity and pressure with the IB-ADM and the parabolic inflow/outflow B.C. in the x-direction. It is shown that the IB-ADM maintains a parabolic x-velocity profile and a discontinuous pressure profile across the thin wall. This is confirmed in Fig. 11 showing the velocity profiles at $x = 0$ with the IB-ADM and SRM with the parabolic inflow/outflow B.C. Notably, the distorted profile does not improve significantly when the number of mesh points is doubled in both directions.

In summary, satisfying the pressure decoupling constraint is important to numerically decouple domains that are physically unrelated, and this condition is most effective in handling a thin solid object.



(a) x-velocity with the inflow/outflow (b) Pressure with the inflow/outflow

(c) x-velocity with the x-periodicity (d) Pressure with the x-periodicity

Fig. 9. Contours of the x-velocity and pressure with the standard reconstruction method. Bold lines denote boundaries of the channels.

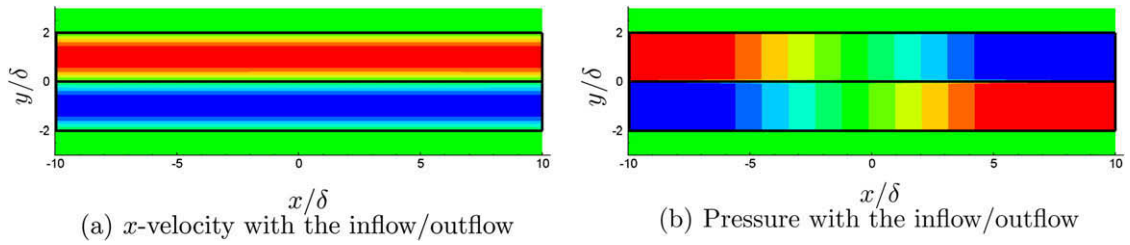


Fig. 10. Contours of the x-velocity and pressure with the IB-ADM. Bold lines denote boundaries of the channels.

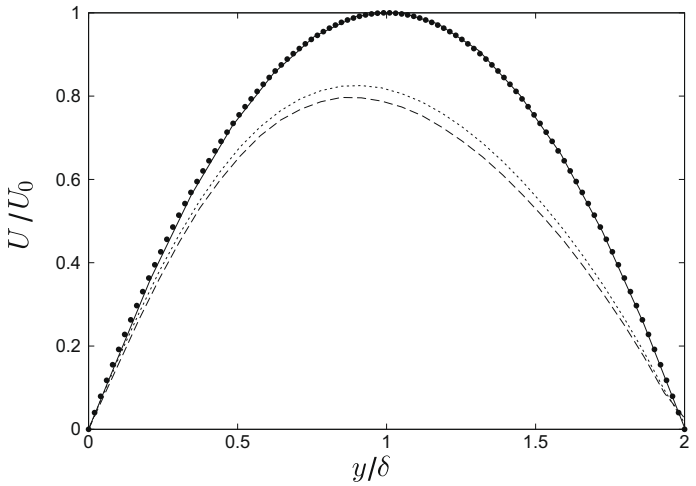


Fig. 11. Profiles of the x-velocity at $x = 0$ with the IB-ADM and standard reconstruction method (SRM) with the parabolic inflow/outflow B.C.: •, the exact solution; —, IB-ADM; - - - -, SRM; ····, SRM with doubled the number of mesh points in x- and y-directions.

4.3. Laminar flow around a circular cylinder

A laminar flow around a circular cylinder was used to verify the wall-pressure predicted with the IB-ADM. The Reynolds number based on the diameter of the cylinder (d) and the freestream velocity (U_∞) is 160. The overall flow conditions are the same as those of Park et al. [40] which used a body-fitted finite difference method for the simulations.

In this section, a locally refined mesh (shown in Fig. 12) is used. The number of mesh points around the circular cylinder is approximately 128. The flow field is averaged in time using about 9000 samples with CFL = 1.6 and about 18,000 samples with CFL = 0.4.

We compared the IB-ADM with the standard reconstruction method, which is the combination of the linear interpolation method and standard mass conservation. With the latter, the velocity field inside the cylinder is artificially set to zero at

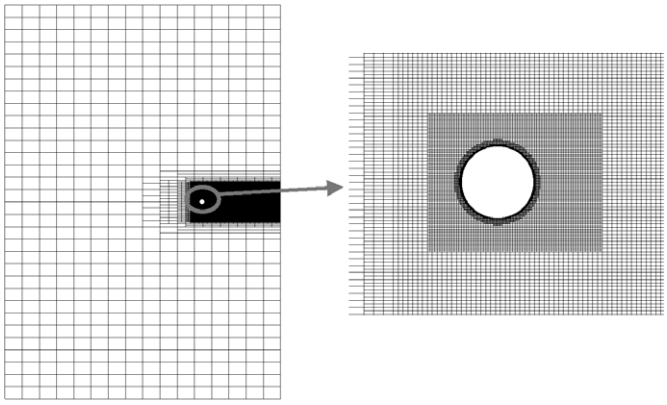


Fig. 12. Locally refined mesh for a laminar flow around a circular cylinder.

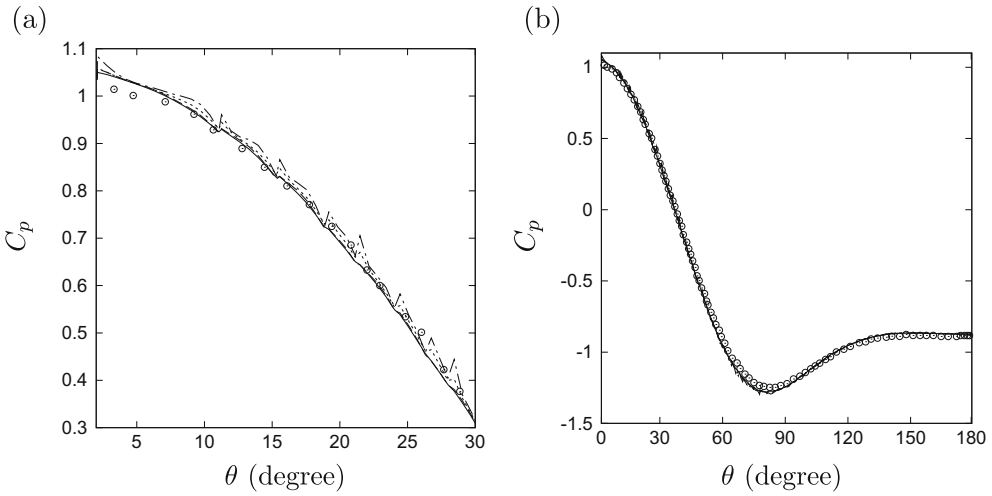


Fig. 13. Time-averaged wall-pressure coefficients around a circular cylinder: ⊙, Park et al. [40]; —, IB-ADM with CFL=1.6; - - - -, IB-ADM with CFL = 0.4; · · ·, standard reconstruction with CFL=1.6; —·—, standard reconstruction with CFL = 0.4: (a) near the stagnation point (b) the whole range.

every time step. Otherwise, the flow field becomes unphysical, as noted in [26], because of coupling of the pressure fields across the IB and the velocity error from the projection step.

Fig. 13(a) shows the time-averaged pressure coefficients near the stagnation point with the two methods. It is shown that wiggles in the wall pressure with the standard reconstruction method are reduced with the IB-ADM. Interestingly, with the standard reconstruction method, the size of wiggles increases as the time step Δt decreases. This is due to the decoupling between local velocity and pressure fields, which results from the fact that the linear interpolation method for the velocity has no contribution from the local pressure gradient [26]. This problem is not observed with the IB-ADM. Except for the wiggles, overall agreement with the previous study is found to be good, as shown in Fig. 13(b).

5. Validation studies

5.1. DNS of a turbulent channel flow at $Re_\tau = 180$

An objective of this study is to assess that the IB method correctly predicts the wall pressure fluctuations in a turbulent flow because of the interest in computing flow-generated noise. In this section, results from a DNS study of a turbulent channel flow at Re_τ=180 are presented. In order to validate the IB method, several statistical quantities including the mean velocity, RMS velocity, pressure spectra, and pressure space–time correlation are compared with previous DNS results. Kim et al. [30] and Moser et al. [39] reported various statistics, including the velocity and pressure spectra in space. Also, Choi and Moin [5] reported the power spectrum and space–time correlation of the wall pressure.

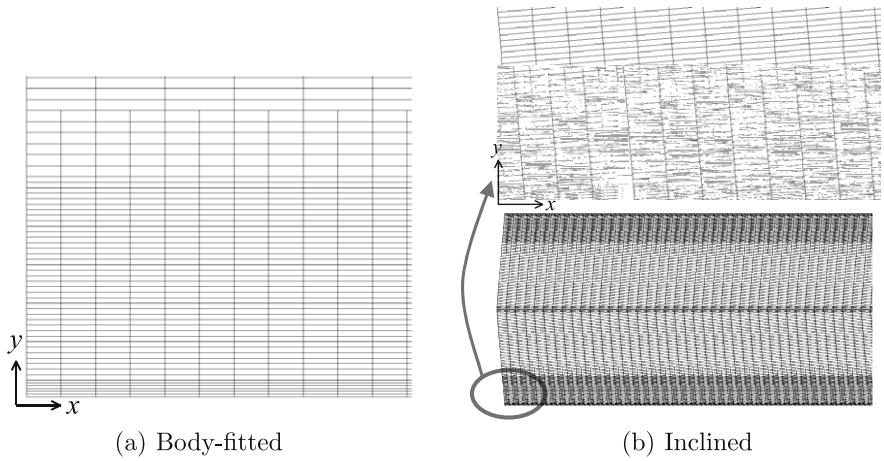


Fig. 14. Locally refined meshes for a turbulent channel flow.

The domain size in the streamwise (x), wall-normal (y), and spanwise (z) direction is $(L_x, L_y, L_z) = (4\pi\delta, 2\delta, 4\pi\delta/3)$, respectively, with δ being the channel half-width. Fig. 14 shows two meshes used in the present study. The first mesh is a body-fitted mesh with local mesh refinement. The IB reconstruction is not required with this mesh as the boundaries are perfectly aligned. The second mesh is generated by inclining the first mesh by 4.8 degrees in the counter-clockwise direction. The mesh is then mirrored across the centerline of the channel to achieve symmetry. This results in stair-step boundaries at the top and bottom of the domain where the IB method is applied. This case is referred to hereafter as the inclined case. For both meshes, the grid spacing in the wall unit is $(\Delta x^+, \Delta y^+, \Delta z^+) = (8.9, 0.7, 5.8)$ at the wall and $(\Delta x^+, \Delta y^+, \Delta z^+) = (17.8, 2.8, 5.8)$ at the channel centerline, respectively. These meshes are slightly under-resolved for DNS with a low-order numerical scheme. The total number of mesh points is approximately 4.4 million.

The flow conditions are the same as in [30,5]. In order to obtain the wall-pressure data for computing the power spectra and space–time correlation, the same time step Δt and sampling rate as [5] were used. The statistical sample was obtained by averaging in the streamwise and spanwise directions, as well as in time.

Fig. 15 shows the mean streamwise velocity profile from the present study. The agreement with the previous DNS result is satisfactory. The difference between the body-fitted and inclined cases is shown to be minor. Fig. 16 shows profiles of the RMS streamwise and wall-normal velocity. Again, agreement with the previous DNS is good. The RMS profiles of the wall-normal velocity exhibit a kink at the locations of grid transitions due to localized discretization errors.

Previous studies also reported the ratio of the RMS wall pressure (p_{rms}) and wall shear stress (τ_w). The result from the present study is shown in Table 1; agreement with the previous DNS results is acceptable. When the previous IB method (SRM) was used with the inclined mesh, it produced about 3.3% larger error (relative to [39]) than the IB-ADM.

Fig. 17 shows the wall-pressure power spectra ϕ as a function of the discrete frequency ω_t compared with the result of [5]. The overall agreement is good; however, there are discrepancies in both the low- and high-frequency regions. The discrepancy in the high-frequency region is due to the difference in the numerical schemes: the present calculations are only

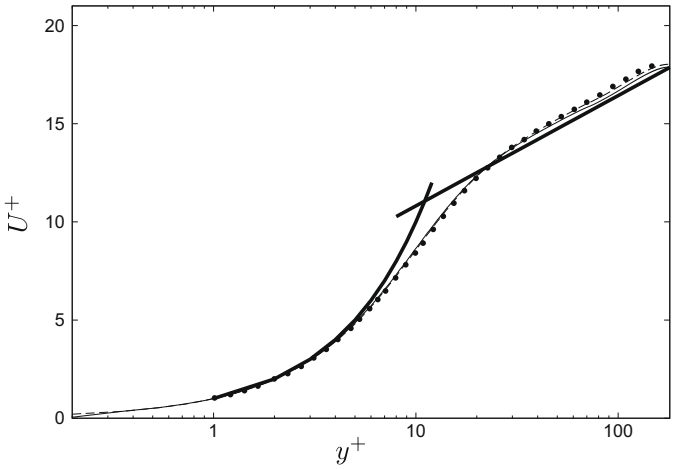


Fig. 15. Mean streamwise velocity profiles in wall units: •, Kim et al. [30]; —, inclined (IB) case; - - - -, body-fitted case.

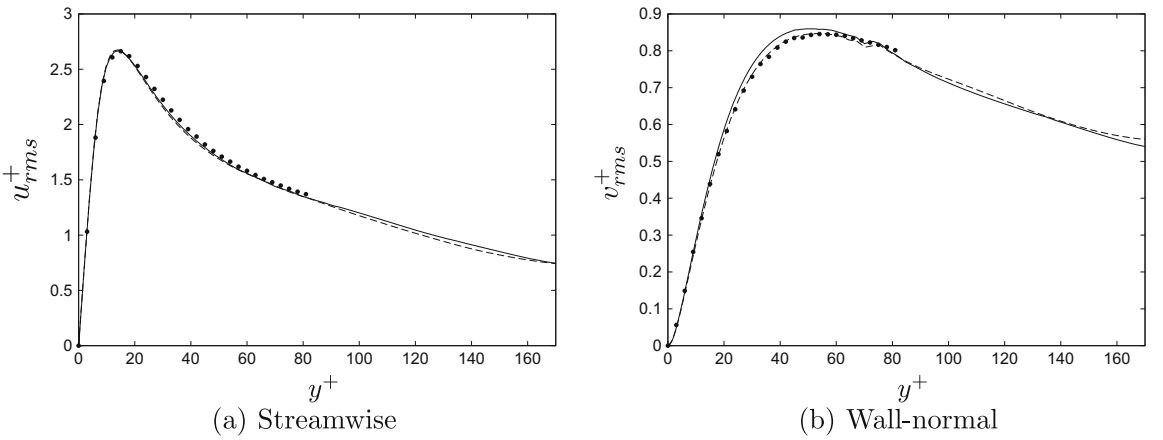


Fig. 16. RMS velocity profiles in wall units: •, Kim et al. [30]; —, inclined (IB) case; - - - -, body-fitted case.

al. [39]	Inclined (IB)	Body-fitted	Inclined (SRM)
	1.556	1.552	1.607

$\rho(\omega_t)$

9 M mesh.

Choi

) mir

× 192

mine

effec

or con

n of th

and c

ra for

ved onl

cy

these observations, we can conclude that the number of samples used in [5] and in the present study might not have been sufficient to achieve fully-converged statistics in the low-frequency region.

Another quantity of interest for the study of turbulence transport is the convection velocity, U_c, which can be interpreted as the translation speed of the Galilean frame in which the integral time scale is maximized [55]. The space–time correlation of the wall pressure $R(r_x, r_t)$ is a function of the streamwise spatial separation (r_x) and temporal separation (r_t). According to Choi and Moin [5], a common definition of U_c is the ratio of r_x and r_t that maximizes $R(r_x, r_t)$:

$$U_c(r_x) = r_x/r_{tc},$$

where r_{tc} is defined as:

$$\left(\frac{\partial R(r_x, r_t)}{\partial r_t}\right)_{r_t = r_{tc}} = 0.$$

In order to find r_{tc}, we can select a discrete r_t value that results in the maximum $R(r_x, r_t)$ for a given r_x. Alternatively, a quadratic polynomial can be used to find r_{tc} for a given r_x. These two approaches are referred to as linear and quadratic polynomial searches, respectively. Fig. 19 shows the convection velocity of the wall pressure computed with these two search methods. The convection velocity based on the quadratic polynomial search produces a smooth function of r_x, while the convection velocity based on the linear search shows closer agreement with the result of [5].

In summary, the statistical data of the present study show a good agreement with results from the previous DNS studies; this indicates that the proposed IB method can correctly predict the wall pressure dynamics associated with the turbulent channel flow.

The present DNS study produces promising results, but raises questions about how effective the IB method is in the presence of a very fine mesh near the wall as compared to a simpler wall treatment, such as the stair-step approximation. In a LES or DNS, the grid spacing in the wall-normal direction is typically less than one in wall units. In case of the no-slip condition, the stair-step approximation has first order accuracy in space. A test was performed for the channel flow with the inclined mesh in Fig. 14(b). The grid spacing in the wall-normal direction is 0.7 at the wall. With the stair-step approximation, the face boundary can be regarded as a rough wall with the roughness height k^+ less than 1. According to Jimenez [25], the effect of this small roughness on the logarithmic layer is negligible. And the effect on turbulent fluctuations is not documented in literature for such small roughness. On the other hand, the similarities between the stair-step approximation and a rough wall are only partially justified, as in the present grid only 1–2 meshes are used to resolve the surface roughness.

Notable differences between the cases with the IB method and stair-step approximation were observed in the mean streamwise velocity drawn in log scale and the RMS wall pressure. Fig. 20 shows the mean streamwise velocity profiles. Compared to the IB method, the viscous sub-layer with the stair-step approximation is shifted up by the roughness height, k. When the coordinate is shifted down by the roughness height, both methods show almost identical results. The same trend is observed in the RMS velocity. The ratio of the RMS wall pressure and wall shear stress is 1.556 with the IB method, while the stair-step approximation results in 1.635 – about 5% increase. In summary, the effect of the stair-step approximation in DNS is appreciable exclusively in the viscous sub-layer.

5.2. LES of the turbulent flow around an airfoil

In this section, results from LES of turbulent flow around a thin airfoil are presented. The airfoil is a cross-section of a commercial low-speed fan blade (its shape was provided by VALEO Inc.). The objective of this test is to validate the IB

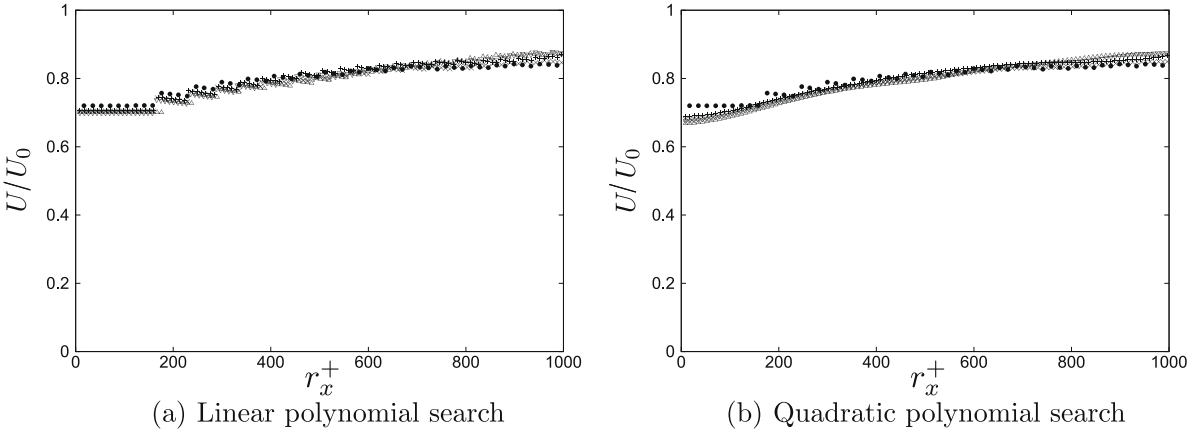


Fig. 19. Convection velocity $U_c(r_x)$ scaled by the centerline velocity U_0 : •, Choi and Moin [5]; +, inclined (IB) case; ×, body-fitted case; △, Cartesian collocated 10M mesh; ▽, Cartesian staggered 10M mesh.


15
10
5

method by comparing turbulent statistics with previous studies. The Reynolds number based on the chord length (c) and the freestream velocity (U_∞) is 150,000. The angle of attack is 8 degrees.

Fig. 21 shows the computational domain. In order to make the simulation comparable to a reference experiment with a reasonable computational cost, a RANS simulation was performed for the full geometry (Fig. 21, left). The RANS solution was used as the far-field boundary condition for a sub-domain where more accurate simulations using LES are carried out. The computational domain in x and y directions is $-1.9 < x/c < 2.4, -1.2 < y/c < 1.3$, respectively. The trailing edge of the airfoil is located at $(x/c, y/c) = (0, 0)$. The domain size in the spanwise direction is $0.1c$, with periodic boundary conditions enforced. Wang et al. [54] mentioned that the improvement is marginal when the spanwise domain size is doubled. The spanwise domain size in wall unit ranges from 1150 to 700 for $-0.8 < x/c < 0.2$.

Local mesh refinement is used to cluster mesh points efficiently near the airfoil surface and in the downstream region where the shear layer develops. Fig. 22 shows the mesh used in the present study. The grid spacing in the wall-normal

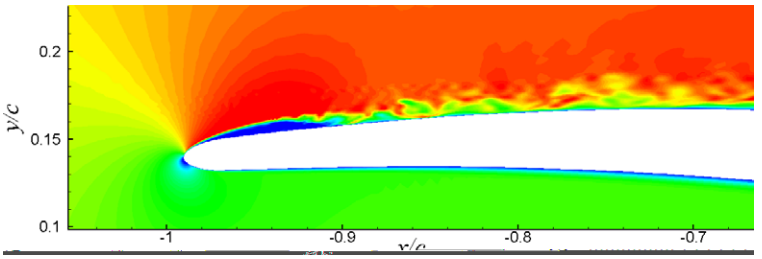


Fig. 23. Contours of the instantaneous x-velocity.

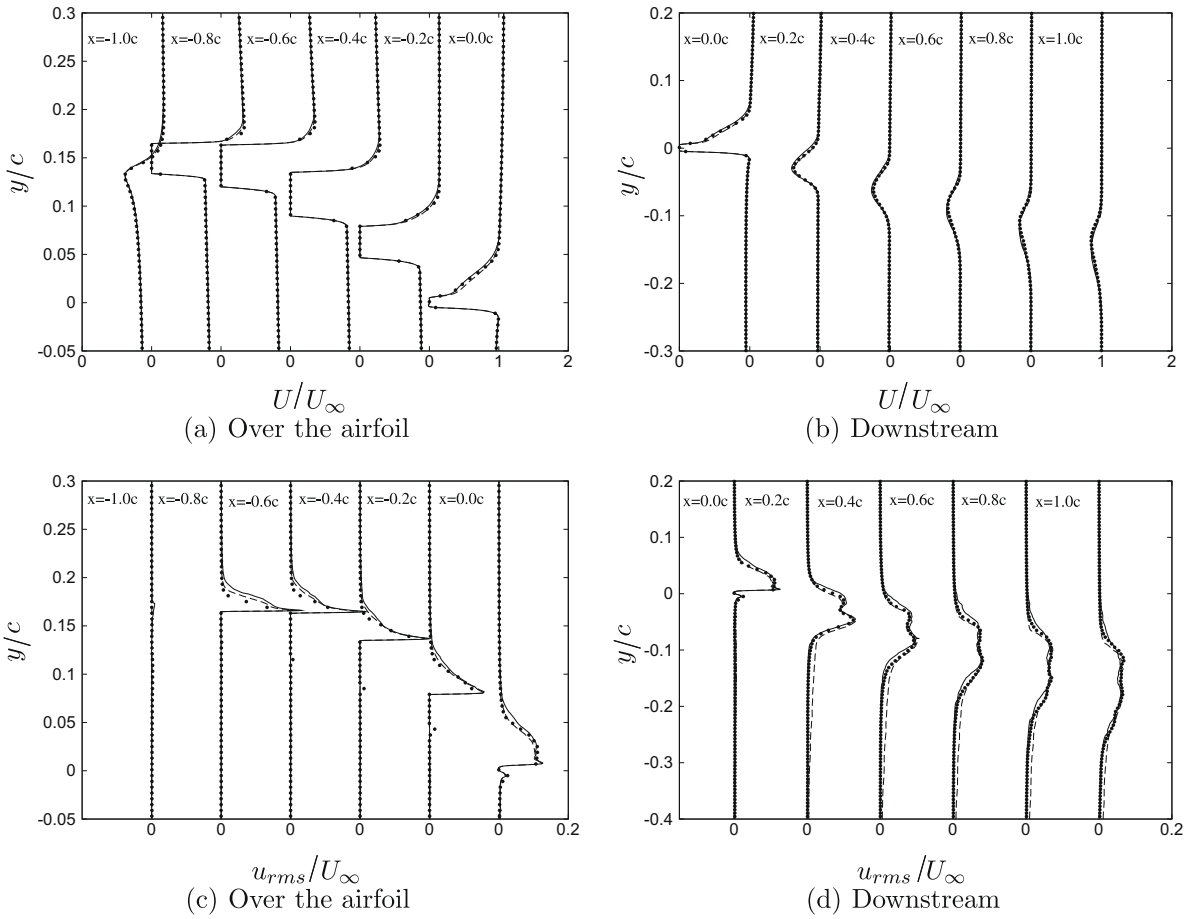


Fig. 24. Averaged and RMS x-velocity profiles at several x-locations: •, Wang et al. [54]; —, IB method; - - - -, body-fitted.

Fig. 26 shows the pressure power spectrum at the trailing edge. Compared to the experiment, the result with the IB method shows a better agreement than the body-fitted simulation at frequencies less than 400 Hz but a worse agreement at frequencies at 400–1500 Hz. The agreement is similar between two cases at frequencies higher than 1500 Hz. Overall agreement is found to be acceptable.

6. Conclusions

IB methods have been developed as an alternative to conventional body-fitted approaches to reduce the difficulties of mesh generation. The objective of the present study was to assess the predictive capabilities of IB method for the near-wall flow dynamics in turbulent flows.

Existing formulations of the IB method were analyzed in order to develop an accurate approach for turbulent flows. First, a pressure decoupling constraint was derived to guarantee the accuracy of the IB formulation by decoupling the flow domain from other physically unrelated domains. In addition, the incompatibility between the interpolated velocity boundary condition and mass conservation was investigated. These lead to the introduction of the immersed boundary-approximated domain method (IB-ADM). The discretized momentum equation and continuity equation are satisfied inside an approximated domain where the reconstructed velocity is used as the boundary condition. In numerical tests, presented IB-ADM shows second-order accuracy and correctly predicts the near-wall velocity, pressure and scalar fields. By satisfying the pressure decoupling constraint, the IB-ADM is shown to successfully handle very thin solid objects.

DNS of a turbulent channel flow at $Re_\tau = 180$ has been performed, and the statistical data, such as the mean & RMS velocity, pressure spectra, and pressure space–time correlation, show good agreement with previous studies. When applied to the turbulent flow around an airfoil, the predicted flow statistics, including the power spectra of the wall pressure, are again in good agreement with a previous body-fitted LES simulation and experiments. These results indicate that the proposed IB method can correctly predict the dynamics of wall-bounded turbulent flows. Compared to the purely Cartesian mesh retaining similar resolution near the wall, locally refined meshes employed herein are found to reduce the number of grid points significantly, thus making the IB method practical for turbulent flows with complex geometries.

Acknowledgments

Financial support from the Department of Energy under Advanced Simulation and Computing (DOE-ASC) program is gratefully acknowledged. We also thank Dr. Stephane Moreau in VALEO inc. for providing geometry and experimental data.

References

[1] M.J. Aftosmis, M.J. Berger, J.E. Melton, Robust and efficient Cartesian mesh generation for component-based geometry, AIAA J. 36 (6) (1998) 952–960.
[2] P.J. Atzberger, P.R. Kramer, C.S. Peskin, A stochastic immersed boundary method for fluid-structure dynamics at microscopic length scales, J. Comput. Phys. 224 (2007) 1255–1292.
[3] D.S. Balsara, C.D. Norton, Highly parallel structured adaptive mesh refinement using parallel language-based approaches, Parallel Comput. 27 (2001) 37–70.
[4] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.
[5] H. Choi, P. Moin, On the space–time characteristics of wall-pressure fluctuations, Phys. Fluids A 2 (8) (1990) 1450–1460.
[6] J.-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys. 224 (2007) 757–784.
[7] A.J. Chorin, The numerical solution of the Navier–Stokes equations for an incompressible fluid, AEC Research and Development Report, NYO-1480-82, New York University, New York, 1967.
[8] W.J. Coirier, K.G. Powell, An accuracy assessment of Cartesian-mesh approaches for the Euler equations, AIAA Paper (1993) 1993–3335.
[9] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.
[10] L.J. Fauci, R. Dillon, Biofluidmechanics of reproduction, Annu. Rev. Fluid Mech. 38 (2006) 371–394.
[11] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method), J. Comput. Phys. 152 (1999) 457–492.
[12] Y. Ge, L.-S. Fan, Three-dimensional direct numerical simulation for film-boiling contact of moving particle and liquid droplet, Phys. Fluids 18 (2006) 117104.
[13] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A 3 (7) (1991) 1760–1765.
[14] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, J. Comput. Phys. 225 (2007) 528–553.
[15] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D geometrically complex moving bodies, J. Comput. Phys. 207 (2005) 457–492.
[16] A. Gilmanov, F. Sotiropoulos, E. Balaras, A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, J. Comput. Phys. 191 (2003) 660–669.
[17] F. Ham, An efficient scheme for large eddy simulation of low-Ma combustion in complex configurations, Annual Research Briefs (Center for Turbulence Research, Stanford University), 2007, p. 41.
[18] F. Ham, K. Mattsson, G. Iaccarino, Accurate and stable finite volume operators for unstructured flow solvers, Annual Research Briefs (Center for Turbulence Research, Stanford University), 2006, p. 243.
[19] F.E. Ham, F.S. Lien, A.B. Strong, A Cartesian grid method with transient anisotropic adaptation, J. Comput. Phys. 179 (2002) 469–494.
[20] V.E. Henson, U.M. Yang, BoomerAMG: a parallel algebraic multigrid solver and preconditioner, Appl. Numer. Math. 41 (2002) 155–177.
[21] G. Iaccarino, G. Kalitzin, P. Moin, B. Khalighi, Local grid refinement for an immersed boundary RANS solver, AIAA Paper, 2004-0586, 2004.
[22] G. Iaccarino, S. Moreau, Natural and forced conjugate heat transfer in complex geometries on Cartesian adapted grids, J. Fluids Eng. 128 (2006) 838–846.
[23] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (3) (2003) 331–347.
[24] T. Ikeno, T. Kajishima, Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations, J. Comput. Phys. 226 (2007) 1485–1508.

[25] J. Jimenez, Turbulent flows over rough walls, Annu. Rev. Fluid Mech. 36 (2004) 173–196.
[26] S. Kang, An Improved Immersed Boundary Method for Computation of Turbulent Flows with Heat Transfer. PhD Thesis, Stanford University, 2008.
[27] S. Kang, G. Iaccarino, P. Moin, Accurate and efficient immersed-boundary interpolations for viscous flows. Annual Research Briefs (Center for Turbulence Research, NASA Ames and Stanford University), 2004, p. 31.
[28] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.
[29] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[30] J. Kim, P. Moin, R.D. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, J. Fluid Mech. 177 (1987) 133–166.
[31] M.P. Kirkpatrick, S.W. Armfield, J.H. Kent, A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid, J. Comput. Phys. 184 (2003) 1–36.
[32] P. Knupp, K. Salari, Verification of Computer Codes in Computational Science and Engineering, third ed., Chapman and Hall/CRC, Boca Raton, FL, 2003.
[33] C. Lee, Stability characteristics of the virtual boundary method in three-dimensional applications, J. Comput. Phys. 184 (2003) 559–591.
[34] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SI AM J. Sci. Comput. 25 (3) (2003) 832–856.
[35] D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method, Phys. Fluids A 4 (3) (1992) 633–635.
[36] Q. Liu, O.V. Vasilyev, A Brinkman penalization method for compressible flows in complex geometries, J. Comput. Phys. 227 (2007) 946–966.
[37] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[38] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. Annual Research Briefs (Center for Turbulence Research, NASA Ames and Stanford University), 1997, p. 317.
[39] R.D. Moser, J. Kim, N.N. Mansour, Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$, Phys. Fluids 11 (4) (1999) 943–945.
[40] J. Park, K. Kwon, H. Choi, Numerical solutions of flow past a circular cylinder at Reynolds number up to 160, KSME Int. J. 12 (6) (1998) 1200–1205.
[41] C.S. Peskin, The fluid dynamics of heart valves: experimental, theoretical and computational methods, Annu. Rev. Fluid Mech. 14 (1982) 235–259.
[42] C.D. Pierce. Progress-variable Approach for Large-eddy Simulation of Turbulent Combustion. PhD Thesis, Stanford University, 2001.
[43] M. Roger, S. Moreau, Broadband self-noise from loaded fan blades, AIAA J. 42 (3) (2004) 536–544.
[44] M. Roger, S. Moreau, Effect of airfoil aerodynamic loading on trailing-edge noise sources, AIAA J. 43 (1) (2005) 41–52.
[45] P.K. Smolarkiewicz, R. Sharman, J. Weil, S.G. Perry, D. Heist, G. Bowker, Building resolving large-eddy simulations and comparison with wind tunnel experiments, J. Comput. Phys. 227 (2007) 633–653.
[46] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2007) 2118–2137.
[47] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.
[48] P.G. Tucker, Z. Pan, Cartesian cut cell method for incompressible viscous flow, Appl. Math. Modell. 24 (2000) 591–606.
[49] M. Tyagi, S. Acharya, Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method, Int. J. Numer. Meth. Fluids 48 (2005) 691–722.
[50] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (2001) 345–380.
[51] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448–476.
[52] R. Verzicco, M. Fatica, G. Iaccarino, P. Orlandi, Flow in an impeller-stirred tank using an immersed-boundary method, AIChE J. 50 (6) (2004) 1109–1118.
[53] R. Verzicco, J. Mohd-Yusof, P. Orlandi, D. Haworth, Large eddy simulation in complex geometric configurations using boundary body forces, AIAA J. 38 (3) (2000) 427–433.
[54] M. Wang, S. Moreau, G. Iaccarino, M. Roger. LES prediction of pressure fluctuations on a low speed airfoil. Annual Research Briefs (Center for Turbulence Research, NASA Ames and Stanford University), 2004, p. 183.
[55] J.A.B. Wills, On convection velocities in turbulent shear flows, J. Fluid Mech. 20 (1964) 417–432.
[56] S. Xu, Z.J. Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, SIAM J. Sci. Comput. 27 (6) (2006) 1948–1980.
[57] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (2006) 12–40.
[58] Z. Yu, X. Shao, A direct-forcing fictitious domain method for particulate flows, J. Comput. Phys. 227 (2007) 292–314.
[59] Z. Yu, X. Shao, A. Wachs, A fictitious domain method for particulate flows with heat transform, J. Comput. Phys. 217 (2006) 424–452.
[60] G. Yun, D. Kim, H. Choi, Vortical structures behind a sphere at subcritical Reynolds numbers, Phys. Fluids 18 (2006) 015102.
[61] H. Zhao, J.B. Freund, R.D. Moser, A fixed-mesh method for incompressible flow-structure systems with finite solid deformations, J. Comput. Phys. 227 (2008) 3114–3140.