Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 283 (2015) 106–129

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Accurate Cartesian-grid simulations of near-body flows at intermediate Reynolds numbers

Audrey P. Maertens[a,*], Gabriel D. Weymouth[b]

[a] Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge 02139, USA
[b] University of Southampton, Southampton SO17 1BJ, UK

Received 16 October 2013; received in revised form 6 September 2014; accepted 9 September 2014
Available online 28 September 2014

Abstract

An accurate Cartesian-grid treatment for intermediate Reynolds number fluid–solid interaction problems is described. We first identify the inability of existing immersed boundary methods to handle intermediate Reynolds number flows to be the discontinuity of the velocity gradient at the interface. We address this issue by generalizing the Boundary Data Immersion Method (BDIM, Weymouth and Yue (2011)), in which the field equations of each domain are combined analytically, through the addition of a higher order term to the integral formulation. The new method retains the desirable simplicity of direct forcing methods and smoothes the velocity field at the fluid–solid interface while removing its bias. Based on a second-order convolution, it achieves second-order convergence in the L_2 norm, regardless of the Reynolds number. This results in accurate flow predictions and pressure fields without spurious fluctuations, even at high Reynolds number. A treatment for sharp corners is also derived that significantly improves the flow predictions near the trailing edge of thin airfoils. The second-order BDIM is applied to unsteady problems relevant to ocean energy extraction as well as animal and vehicle locomotion for Reynolds numbers up to 10^5.

Keywords: Cartesian-grid; Immersed boundary method; Fluid–body interactions; Intermediate Reynolds numbers; Airfoil; Sharp edge

1. Introduction

Immersed Boundary (IB) methods have become popular in the last ten years for simulating flows with complex geometries and moving boundaries. IBs remove the effort needed to generate a body-fitted grid and enable the use of efficient numerical methods that can be easily solved in parallel (see [1] for a review of IB methods). This relative simplicity makes IBs particularly attractive for engineering applications and the study of animal locomotion. However, these applications are often characterized by large Reynolds numbers, which we will show are particularly challenging for IB methods.

Introduced by Peskin in the 1970s [2] to simulate heart valves, IB methods were first developed to solve the coupled motion of an elastic boundary immersed in a viscous fluid on a fixed Cartesian-grid. The effect of the IB on the surrounding fluid is simulated by the addition of a force density (which represents the force of the surface of the

* Corresponding author. Tel.: +1 6173787825.
E-mail addresses: maertens@mit.edu (A.P. Maertens), G.D.Weymouth@soton.ac.uk (G.D. Weymouth).

http://dx.doi.org/10.1016/j.cma.2014.09.007
0045-7825/© 2014 Elsevier B.V. All rights reserved.

object on the fluid) to the Navier–Stokes equations [3]. These methods have then been extended to fluids with solid boundaries by defining artificial body forces [4]. Many options have been explored for defining the forcing (structure attached to an equilibrium with a spring [5], explicit feedback controller [4], porous medium [6]) but all require user specified parameters and are subject to severe stability constraints due to their stiffness [7].

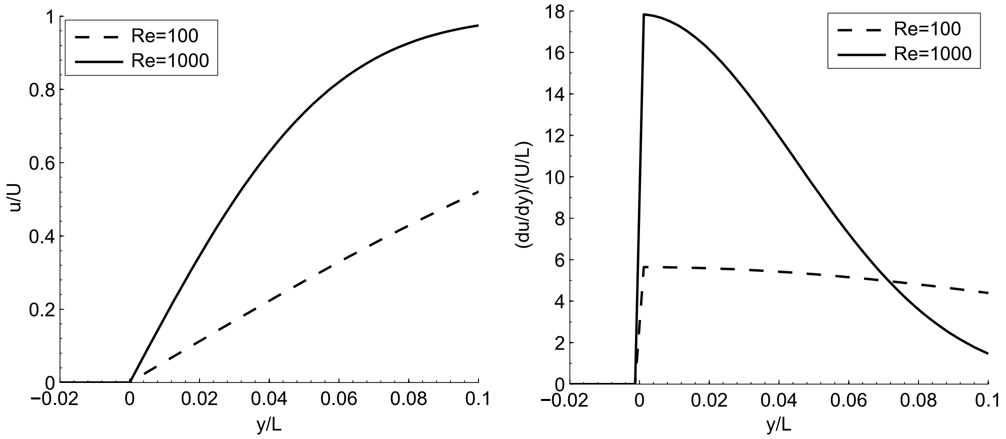
To overcome these limitations, Fadlun et al. [7] proposed a formulation in which the forcing is directly estimated from the discrete problem such as to impose the desired velocity on the boundary. This method and the many variations that have subsequently appeared in the literature are direct forcing methods and have been widely used for flows in which the motion of the boundary is prescribed. A well known issue with this class of algorithms is their tendency to introduce large non-physical pressure oscillations (see [8] for example). Muldoon [9] even showed that the pressure could locally increase without bound as the time step goes to zero. These oscillations are caused by the lack of smoothness of the velocity across the boundary before the projection step [10]. A related issue is that these methods account for the boundary in the momentum conservation equations but not in the mass conservation equation. Uhlmann [11] proposed an alternative direct-forcing formulation in which the forcing is first computed on Lagrangian markers, then spread onto the neighboring Eulerian nodes. While not directly addressing the mass conservation issue, this formulation later generalized by [12] and [13] has been shown to significantly reduce undesirable force oscillations.

In sharp-interface approaches, the communication between the moving boundary and the flow solver is usually accomplished by explicitly modifying the computational stencil near the IB. Unlike forcing methods, sharp-interface approaches alter both conservation equations, usually using a ghost-fluid [14] or ghost-cell method [15,16]. But Seo et al. [17] showed that even with such treatment, local mass conservation is violated which produces pressure fluctuations. Instead, they suggest the use of cut-cell finite volumes that reshape the cells in the vicinity of IBs [18–20]. However, cut-cells in three dimensions can produce seven different polyhedral control volumes and arbitrarily small cells. The small cells need to be merged to avoid stability problems and “freshly cleared” cells that appear with moving boundaries need a careful treatment in order to avoid pressure fluctuations [19,20]. Due to all these considerations, sharp-interface approaches, and especially cut-cell methods have lost the simplicity which was the main appeal of IB methods.

An alternative approach, called the Boundary Data Immersion Method (BDIM), has been proposed by Weymouth and Yue [21]. Similarly to Uhlmann’s direct forcing formulation [11–13], BDIM relies on convolving the equation governing the motion of the immersed boundary with the Navier–Stokes equations. The boundary, however, is represented by a distance function rather than Lagrangian points. The fundamental difference between BDIM and direct forcing methods, though, is an additional modification to the pressure term analogous to the discrete operator adjustments of sharp-interface methods such as [19] and [14], avoiding the projection issues discussed in [10]. Unlike sharp-interface methods, BDIM alters the analytic equations near the embedded boundary (and not the discrete operators), which makes the method easy to implement in existing flow solvers regardless of the geometry being simulated. This enables BDIM to predict a smooth pressure field even for flow featuring moving boundaries, while retaining the simplicity that makes IBs attractive. This method has proved its versatility by successfully simulating a variety of low Reynolds number and multi-phase flows [22–25].

BDIM can also be compared with the volume-penalization IB method of Kajishima [26], wherein the interpolating function represents the volumetric fraction of the fluid in the computational cell. BDIM could reproduce this property using a linear kernel, but in practice we use a smoother kernel to help avoid spurious force oscillations as discussed in [27]. Additionally, because in BDIM the interpolation coefficient is only a function of the distance to the boundary and is independent of the grid, its calculation is trivial compared to the volume fraction. Finally, like direct forcing methods, Kajishima’s method estimates the pressure without taking the solid into account whereas, as will be discussed in Section 2.4, BDIM also modifies the pressure equation.

The next big challenge for IB methods lies in moderate to high Reynolds number flows, which give rise to fundamental problems for existing approaches [1,28]. The source of these problems is illustrated here by considering a one-dimensional unsteady channel flow in which flow of kinematic viscosity ν with uniform x-velocity U suddenly enters a channel with opening $0 < y < L$ at time $t = 0$. Fig. 1(a) shows the velocity profile $u(y, t)/U$ near the boundary for two Reynolds numbers $Re = L^2/(\nu t) = 100$ and 1000 computed on a body-fitted grid and Fig. 1(b) shows the corresponding derivatives. The solution is uniformly zero in the solid domain ($y < 0$) whereas the solution in the fluid ($y > 0$) has a non-zero slope at the interface. Therefore, even though the velocity field is continuous across the boundary, its first derivative is not. Guy [10] showed that the pressure fluctuations in direct forcing methods are caused by the incompatibility of smooth IB methods with this discontinuity in the first derivative of the velocity. The



(a) Velocity profile.

(b) Velocity derivative.

Fig. 1. Velocity profile and its derivative at the wall in a one-dimensional channel flow of height L for $Re = L^2/(\nu t) = 100$ and 1000.

higher the Reynolds number, the larger the jump in the velocity derivative, exacerbating this problem and requiring special techniques for accurate simulation.

In all direct forcing methods as well as in BDIM, a weighted average between the fluid and solid velocities is used to estimate the fluid velocity near a solid boundary. Such a treatment will be referred to as *1st order* in the rest of this paper. While a *1st order* treatment of the boundary can allow accurate predictions at low Reynolds numbers, they are not appropriate for flows characterized by a thin boundary layer.

In this work we extend [21] by using the analytic BDIM formulation to establish a higher order formulation of the near-boundary interaction between the fluid and solid domains. The addition of the higher order term improves the accuracy of the method at high Reynolds number while generating a smoother velocity profile that reduces pressure fluctuations. We show through high resolution simulations that this analytically derived first-moment correction enables accurate simulations of high speed flows without introducing any new model parameters. Section 2 develops the new second-order BDIM approach. Specifically, Section 2.2 proposes an analytical equation that generalizes the convolution evaluation of [21] through the addition of higher order terms. Comparison of the new formulation with that of [21] and a direct forcing method is detailed in Section 2.3 in the context of a one-dimensional channel flow. Finally, a finite volume implementation of the proposed method is presented in Section 2.4. The generalized BDIM formulation is then applied to two and three-dimensional fluid–solid systems in Section 3, which demonstrate its improved accuracy for intermediate Reynolds number flows. Two-dimensional flow past a cylinder is used in Section 3.1 to assess the numerical properties of the method as well as present and validate the force calculation method. The new formulation is then tested on flows relevant to applications of practical interest: unsteady two and three-dimensional viscous flows past a stationary and a moving airfoil in Sections 3.2 and 3.3, as well as a three-dimensional multi-body application in Section 3.4. This last example illustrates a case for which an Immersed Boundary method is more appropriate than a body-fitted one. An improved treatment of sharp edges, essential for thin airfoils, is also derived in Appendix A.

2. The boundary data immersion method revisited

In this work we consider a two-domain interaction problem in which the domain Ω_f is occupied by an incompressible viscous fluid and the domain Ω_b by a solid or deforming body with prescribed velocity $\vec{V}(\vec{x}, t)$. The governing equation in the solid body is simply given by

$$\vec{u} = \vec{V} \tag{1}$$

whereas the fluid is governed by the incompressible Navier–Stokes equation

$$\frac{\partial \vec{u}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla}\right) \vec{u} + \frac{1}{\rho} \vec{\nabla} p - \nu \nabla^2 \vec{u} = 0. \tag{2}$$

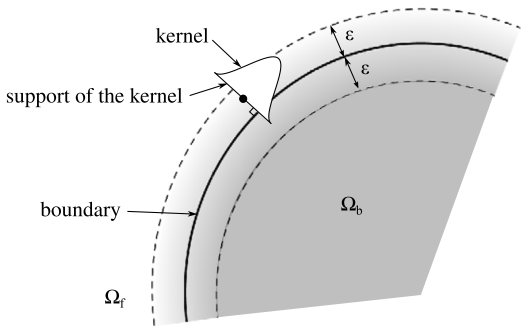


Fig. 2. Smoothing across the immersed boundary. The equations valid in each domain are convolved with a kernel of radius ϵ and added together. The gradient of gray illustrates how the contribution of b_ϵ and f_ϵ to the smoothed equation changes in the boundary region. The kernel at a point (marked by a dot) that belongs to the boundary region is represented.

After integration of Eq. (2) over a time step Δt, the fluid and body equations can be written in the form

$$\begin{cases} \vec{u} = \vec{b}, & \text{for } \vec{x} \in \Omega_b \\ \vec{u} = \vec{f}(\vec{u}), & \text{for } \vec{x} \in \Omega_f \end{cases} \tag{3}$$

with

$$\vec{b} = \vec{V} \tag{4a}$$

$$\vec{f}(\vec{u}, t_0 + \Delta t) = \underbrace{\vec{u}(t_0)}_{\vec{u}^0} + \underbrace{\int_{t_0}^{t_0+\Delta t} \left[-\left(\vec{u}\cdot\vec{\nabla}\right)\vec{u} + \nu\nabla^2\vec{u}\right] \mathrm{d}t}_{\vec{R}_{\Delta t}(\vec{u})} - \underbrace{\int_{t_0}^{t_0+\Delta t} \frac{1}{\rho}\vec{\nabla} p \, \mathrm{d}t}_{\partial\vec{P}_{\Delta t}} \tag{4b}$$

where $\partial\vec{P}_{\Delta t}$ is the pressure impulse over Δt and $\vec{R}_{\Delta t}(\vec{u})$ accounts for all the non-pressure terms.

2.1. Smooth multi-domain coupling

IB methods aim at solving Eq. (3) using a grid that does not conform to the boundary between Ω_f and Ω_b. The approach proposed in [21] to solve Eq. (3) consists in convolving the continuous equations with a nascent delta kernel in order to combine them in a smooth meta-equation.

Eq. (3) can be written as a single equation

$$\vec{u}(\vec{x}, t) = \vec{b}(\vec{x}, t)\mathbf{1}_{\Omega_b}(\vec{x}) + \vec{f}(\vec{u}, \vec{x}, t)\mathbf{1}_{\Omega_f}(\vec{x}) \quad \text{for } \vec{x} \in \Omega \tag{5}$$

where $\mathbf{1}_A$ is the indicator function of subspace A. Convolving both sides with a nascent delta kernel K_ϵ with spherical support of radius ϵ yields the following smoothed equation

$$\vec{u}_\epsilon(\vec{x}, t) = \int_\Omega \vec{u}(\vec{x}', t) K_\epsilon(\vec{x}, \vec{x}') \, \mathrm{d}\vec{x}' = \vec{b}_\epsilon(\vec{x}, t) + \vec{f}_\epsilon(\vec{u}_\epsilon, \vec{x}, t) \quad \text{for } \vec{x} \in \Omega \tag{6}$$

where

$$\vec{b}_\epsilon(\vec{x}, t) = \int_{\Omega_b} \vec{b}(\vec{x}_b, t) K_\epsilon(\vec{x}, \vec{x}_b) \, \mathrm{d}\vec{x}_b \tag{7a}$$

$$\vec{f}_\epsilon(\vec{u}_\epsilon, \vec{x}, t) = \int_{\Omega_f} \vec{f}(\vec{u}_\epsilon, \vec{x}_f, t) K_\epsilon(\vec{x}, \vec{x}_f) \, \mathrm{d}\vec{x}_f. \tag{7b}$$

Thus the general equation (6) smoothly transitions from the fluid equation to the solid equation as illustrated in Fig. 2. In the dark gray area, $\vec{f}_\epsilon = 0$, such that $\vec{u}_\epsilon = \vec{b}_\epsilon$. Similarly, in the white area, $\vec{b}_\epsilon = 0$, so $\vec{u}_\epsilon = \vec{f}_\epsilon$. The black dot is within distance ϵ of the boundary, therefore the kernel centered at that point intersects both Ω_f and Ω_b. At that point, both $\vec{b}_\epsilon$ and $\vec{f}_\epsilon$ contribute to $\vec{u}_\epsilon$.

2.2. Evaluation of the convolution

Eq. (6) is very general and can be applied to any multi-domain problem by replacing Eq. (4) with the appropriate equation for each domain. In order to solve Eq. (6) numerically, we need to estimate the integrals on a grid. We wish a formulation that is grid independent, so that it can easily be implemented on any grid with little computational overhead, even for moving three-dimensional objects. In order to minimize the dissipative effects on the solution, it is necessary to ensure smoothing only occurs where it is needed to alleviate the discontinuity discussed in the introduction, i.e. on the boundary region in the normal direction. Therefore, two requirements will be kept in mind while discretizing Eq. (6): (i) smoothing only occurs near the boundary and (ii) smoothing occurs across the boundary but not along it.

The naive way of discretizing Eq. (6) would be to approximate the continuous convolution by its discrete counterpart

$$\vec{b}_\epsilon(\vec{x},t) = \int_{\Omega_b} \vec{b}(\vec{x}_b,t) K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b = \sum_{\vec{x}_b\in\Omega_b} \vec{b}(\vec{x}_b,t) K_\epsilon(\vec{x},\vec{x}_b)\Delta x_b^d + \mathcal{O}(\Delta x^2) \tag{8}$$

where Δx denotes the grid size and d the dimensionality (2 or 3). However, this formulation violates the first requirement. Since the kernel $K_\epsilon(\vec{x},\vec{y})$ has a finite support, the integral can alternatively be evaluated using a Taylor expansion

$$\vec{b}_\epsilon(\vec{x},t) = \int_{\Omega_b} \vec{b}(\vec{x}_b,t) K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b \tag{9a}$$

$$= \int_{\Omega_b} \left(\vec{b}(\vec{x},t) + \vec{\nabla}\vec{b}(\vec{x},t)\cdot(\vec{x}_b-\vec{x})\right) K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b + \mathcal{O}(\epsilon^2) \tag{9b}$$

$$= \vec{b}(\vec{x},t)\int_{\Omega_b} K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b + \vec{\nabla}\vec{b}(\vec{x},t)\cdot\int_{\Omega_b}(\vec{x}_b-\vec{x})K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b + \mathcal{O}(\epsilon^2) \tag{9c}$$

where $\mathcal{O}(\epsilon^2)$ appears on the right hand side to indicate the order of the error introduced by this linearization. Note that if the velocity within the support of the kernel is not smooth enough for the Taylor expansion to provide a valid approximation, local grid refinement (as in [28]) can be used, which will also greatly improve the accuracy of the discrete differential operators.

In order to compute Eq. (9) in the boundary region, the body velocity is extended into the fluid domain. In the case of non-uniform body velocity, this is done using simple linear extrapolation. Note that this means the prescribed velocity may not be divergence free, but the modified pressure equation maintains divergence free flow in the fluid domain. Since we now have a smooth velocity field $\vec{u}_\epsilon$, the fluid equation can also be extended into the body part of the boundary region. Defining $\hat{n}$ and $\hat{\tau}$, respectively the normal and tangent to the closest point on the fluid–solid interface, we express Eq. (9c) as:

$$\begin{aligned}\vec{b}_\epsilon(\vec{x},t) \approx\ & \vec{b}(\vec{x},t)\int_{\Omega_b} K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b + \frac{\partial\vec{b}}{\partial n}(\vec{x},t)\int_{\Omega_b}(\vec{x}_b-\vec{x})\cdot\hat{n}\ K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b \\ & + \frac{\partial\vec{b}}{\partial\tau}(\vec{x},t)\int_{\Omega_b}(\vec{x}_b-\vec{x})\cdot\hat{\tau}\ K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b.\end{aligned} \tag{10}$$

In order to satisfy the second requirement, a kernel that only depends on the distance to the boundary is used

$$K_\epsilon(\vec{x},\vec{y}) = K_\epsilon(\vec{x}\cdot\hat{n}\,\hat{n}, \vec{y}\cdot\hat{n}\,\hat{n}). \tag{11}$$

If the radius of curvature of the interface is large compared to the grid size, the boundary can be locally approximated by its tangential plane [26], which significantly simplifies the calculation. Indeed, assuming the boundary is locally flat ($\hat{n}$ is constant across the support of the kernel), the tangential component of the integration can be eliminated and the kernel $K_\epsilon(\vec{x},\vec{y})$ be replaced by a one-dimensional kernel $\phi_\epsilon(\vec{x}\cdot\hat{n}, \vec{x}_b\cdot\hat{n})$

$$\int_{\Omega_b} K_\epsilon(\vec{x},\vec{y})\mathrm{d}\vec{x}_b = \int_{\Omega_b\cdot\hat{n}} \underbrace{K_\epsilon(\vec{x}\cdot\hat{n}\,\hat{n}, \vec{y}\cdot\hat{n}\,\hat{n})\left(\int_{\Omega_b\cdot\hat{\tau}}\mathrm{d}\vec{x}_b\cdot\hat{\tau}\right)}_{\phi_\epsilon(\vec{x}\cdot\hat{n},\vec{x}_b\cdot\hat{n})}\ \underbrace{\mathrm{d}\vec{x}_b\cdot\hat{n}}_{\mathrm{d}x_b^{\hat{n}}}. \tag{12}$$

The convolution then simplifies to:

$$\vec{b}_\epsilon(\vec{x},t) \approx \vec{b}(\vec{x},t) \underbrace{\int \phi_\epsilon(\vec{x}\cdot\hat{n}, \vec{x}_b\cdot\hat{n})\, \mathrm{d}x_b^{\hat{n}}}_{\mu_0^{\epsilon,B}} + \frac{\partial b}{\partial n}(\vec{x},t) \underbrace{\int (\vec{x}_b - \vec{x})\cdot\hat{n}\; \phi_\epsilon(\vec{x}\cdot\hat{n}, \vec{x}_b\cdot\hat{n})\, \mathrm{d}x_b^{\hat{n}}}_{\mu_1^{\epsilon,B}} \tag{13a}$$

$$\approx \vec{b}(\vec{x},t)\; \mu_0^{\epsilon,B} + \frac{\partial \vec{b}}{\partial n}(\vec{x},t)\; \mu_1^{\epsilon,B} \tag{13b}$$

where $\mu_0^{\epsilon,B}$ and $\mu_1^{\epsilon,B}$ are respectively the zeroth and first moments of the one-dimensional kernel ϕ_ϵ over Ω_b. Similar expressions can be obtained when the boundary is not locally flat. For example, the derivation in the presence of a sharp corner can be found in Appendix A.

The same simplification holds for $\vec{f}_\epsilon(\vec{u}_\epsilon, \vec{x}, t)$:

$$\vec{f}_\epsilon(\vec{u}_\epsilon,\vec{x},t) \approx f(\vec{u}_\epsilon,\vec{x},t)\; \mu_0^{\epsilon,F} + \frac{\partial f}{\partial n}(\vec{u}_\epsilon,\vec{x},t)\; \mu_1^{\epsilon,F} \tag{14}$$

where $\mu_0^{\epsilon,F}$ and $\mu_1^{\epsilon,F}$ are the moments over Ω_f. The kernel is chosen symmetric such that $\mu_1^{\epsilon,F} = \mu_1^{\epsilon,B} = 0$ outside of the boundary region. Consequently, $\vec{u}_\epsilon = \vec{f}_\epsilon = \vec{f}$ in the fluid, away from the boundary and vice versa in the solid. It is also chosen positive in order to guarantee the convergence of a broad spectrum of algorithms traditionally used to solve the Navier–Stokes equations. Note that extending the method to higher than second-order will require dealing with the non-zero second moment of positive kernels outside the smoothing region.

Unlike distributed forcing methods [27], the solution proposed here is rather insensitive to the exact form of the kernel. In particular, as long as the kernel is continuous, changing it does not affect the possible pressure fluctuations or convergence properties. The following kernel will be used in the rest of this paper:

$$\phi_\epsilon(x,y) = \begin{cases} \big(1+\cos(|x-y|\pi/\epsilon)\big)/(2\epsilon) & \text{if } |x-y| < \epsilon \\ 0 & \text{if } |x-y| \geq \epsilon. \end{cases} \tag{15}$$

Using this kernel and defining $d(\vec{x})$ the signed distance from $\vec{x}$ to the fluid–body boundary ($d > 0$ in the fluid, $d < 0$ in the body), we find:

$$\mu_0^\epsilon(d) = \begin{cases} \dfrac{1}{2}\left[1 + \dfrac{d}{\epsilon} + \dfrac{1}{\pi}\sin\left(\dfrac{d}{\epsilon}\pi\right)\right] & \text{for } |d| < \epsilon \\ 0 & \text{for } d \leq -\epsilon \\ 1 & \text{for } d \geq \epsilon \end{cases} \tag{16a}$$

$$\mu_1^\epsilon(d) = \begin{cases} \epsilon\left[\dfrac{1}{4} - \left(\dfrac{d}{2\epsilon}\right)^2 - \dfrac{1}{2\pi}\left(\dfrac{d}{\epsilon}\sin\left(\dfrac{d}{\epsilon}\pi\right) + \dfrac{1}{\pi}\left(1+\cos\left(\dfrac{d}{\epsilon}\pi\right)\right)\right)\right] & \text{for } |d| < \epsilon \\ 0 & \text{for } |d| \geq \epsilon \end{cases} \tag{16b}$$

where $\mu_0^\epsilon = \mu_0^{\epsilon,F}$ and $\mu_1^\epsilon = \mu_1^{\epsilon,F}$. For the complementary domain Ω_b, we simply have $\mu_0^{\epsilon,B}(d) = \mu_0^\epsilon(-d) = 1 - \mu_0^\epsilon(d)$ and $\mu_1^{\epsilon,B}(d) = -\mu_1^\epsilon(-d) = -\mu_1^\epsilon(d)$. The one-dimensional kernel ϕ_ϵ and its zeroth and first order moments are shown in Fig. 3. Since these have been calculated analytically in the continuous domain, BDIM would belong to the 'continuous forcing' approach according to Mittal's definition [1]. This also means that the formulation derived here can as easily be used on a Cartesian grid (uniform or not) as on a tetrahedral or unstructured mesh.

Combining the simplified convolution Eqs. (13) and (14) with Eq. (6) we finally obtain the new meta-equation

$$\vec{u}_\epsilon(\vec{x}) = \mu_0^\epsilon\, \vec{f} + (1-\mu_0^\epsilon)\, \vec{b} + \mu_1^\epsilon\, \frac{\partial}{\partial n}(\vec{f} - \vec{b}). \tag{17}$$

This meta-equation generalizes the one from [21] by adding the first-order term in the expansion of the convolution in Eq. (9). As noted in [21], μ_0^ϵ can be interpreted as an interpolating function acting on the governing equations. The new μ_1^ϵ term increases the order of interpolation, improving accuracy in the presence of a large discontinuity in the velocity gradient. As will be illustrated in Section 2.3, the μ_1 term further smoothes the transition and results in

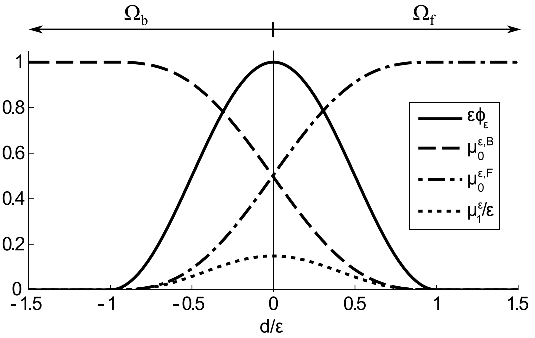


Fig. 3. Integration kernel and its zeroth and first order moments. Outside of the boundary region ($|d| > \epsilon$), $\mu_1^\epsilon = 0$. In the fluid for $d > \epsilon$, $\mu_0^{\epsilon,B} = 0$ and $\mu_0^{\epsilon,F} = 1$. Similarly, in the body for $d < -\epsilon$, $\mu_0^{\epsilon,F} = 0$ and $\mu_0^{\epsilon,B} = 1$. Within the smoothing region ($|d| < \epsilon$) all values are non-zero.

quadratic convergence in the external flow. Therefore, the present formulation, based on a second-order convolution, will be referred to as *2nd order* BDIM, whereas the formulation presented in [21] will be referred to as *1st order* BDIM.

2.3. Application to a one-dimensional channel flow

In order to illustrate the role played by the first moment correction, the simplistic example of the one-dimensional unsteady channel flow presented in the Introduction is considered again. This example has been chosen because, except for the mass conservation equation that is absent, its treatment is very similar to that of the full three-dimensional unsteady Navier–Stokes equations and an exact solution can be calculated (see Appendix B.1). A forward Euler scheme is used such that the transition from time t_0 to time $t_0 + \Delta t$ have a very simple expression:

$$b = 0 \tag{18a}$$

$$f(u, y, t_0 + \Delta t) = u(y, t_0) + \Delta t\, \nu\, \frac{\partial^2}{\partial y^2} u(y, t_0). \tag{18b}$$

The transition equation for *2nd order* BDIM is given by

$$u_\epsilon(y, t_0 + \Delta t) = \left[\mu_0^\epsilon + \mu_1^\epsilon \frac{\partial}{\partial y}\right]\left[1 + \Delta t\, \nu\, \frac{\partial^2}{\partial y^2}\right] u_\epsilon(y, t_0), \tag{19}$$

which we write in matrix form

$$u_\epsilon(n\Delta t) = \left(\left[\mu_0^\epsilon + \mu_1^\epsilon \mathbf{D}\right]\left[1 + \Delta t\, \nu\, \mathbf{D}^{(2)}\right]\right)^n U, \tag{20}$$

where $\mathbf{D}$ and $\mathbf{D}^{(2)}$ are tridiagonal matrices resulting from the second-order central differencing of the first- and second-order derivatives respectively. The moments μ_0^ϵ and μ_1^ϵ are given by Eq. (16) where the distance function is

$$d(y) = \frac{L}{2} - \left|\frac{L}{2} - y\right| \tag{21}$$

for this channel geometry.

We will compare the error in the *2nd order* BDIM solution to two formulations from the literature; the *1st order* BDIM from [21] and a direct forcing method adapted from the approach in [27]. The *1st order* BDIM transition matrix can be recovered by setting μ_1^ϵ to zero in Eq. (20). The direct forcing formulation, which we derive in Appendix B.2, is

$$u_\epsilon(n\Delta t) = \left(\left[1 - \zeta_\epsilon \zeta_\epsilon^T\right]\left[1 + \Delta t\, \nu\, \mathbf{D}^{(2)}\right]\right)^n U, \tag{22}$$

where ζ_ϵ is a column vector defined by $\zeta_\epsilon(y) = (\phi_\epsilon(d(y), 0) + \phi_\epsilon(d(y - L), 0))\, dy$ for the kernel ϕ_ϵ defined by Eq. (15). Calling u_0 the exact solution and u_g an approximate solution calculated on grid g, we define the L_∞ error

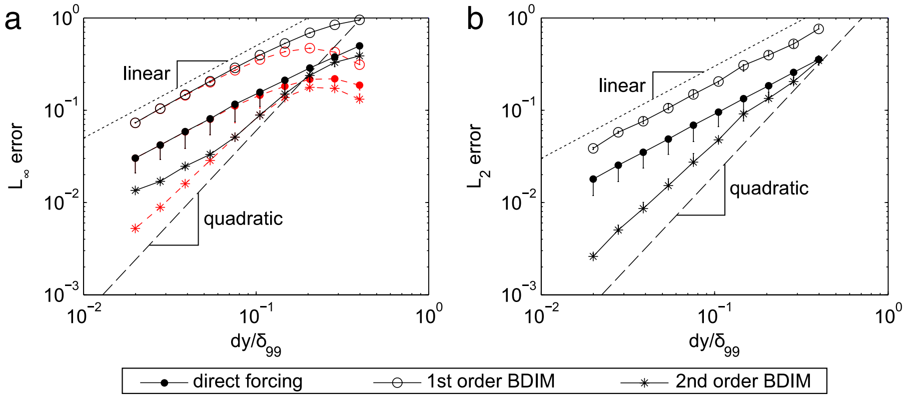


Fig. 4. L_∞ (a) and L_2 (b) norms of the velocity error in the channel as a function of the grid spacing normalized by δ_{99}, the 99% boundary layer thickness as defined in Appendix B.1. Error bars show the spread of the error calculated for Reynolds numbers $Re = [100,\ 1000,\ 10\,000]$. In (a), the black solid curves show the L_∞ norm across the whole channel ($p = 0$), whereas the red dashed curves show the error away from the transition region ($p = \epsilon$). Note that as dy increases, so does the region excluded from the $e_\infty(dy, \epsilon)$ error, causing the value to decrease artificially when the grid is so coarse that most of the boundary layer is excluded.

with parameter p for grid spacing dy

$$e_\infty(dy, p) = \max_{g \in \mathcal{G}(dy)} \left[\max_{y \in [p, L-p]} |u_0 - u_g| \right], \tag{23}$$

where p is the location of the first point away from the wall included in the error metric, and the L_2 error

$$e_2(dy) = \max_{g \in \mathcal{G}(dy)} \sqrt{\frac{1}{2\delta_{99}} \int_{y=0}^{L} (u_0 - u_g)^2 \mathrm{d}y}, \tag{24}$$

where δ_{99} is the 99% boundary layer thickness (see Appendix B.1) and 10 grids of similar spacing but different offset are used in $\mathcal{G}(dy)$. For all cases, we ensure that the simulations are converged in time by using 10^7 time steps.

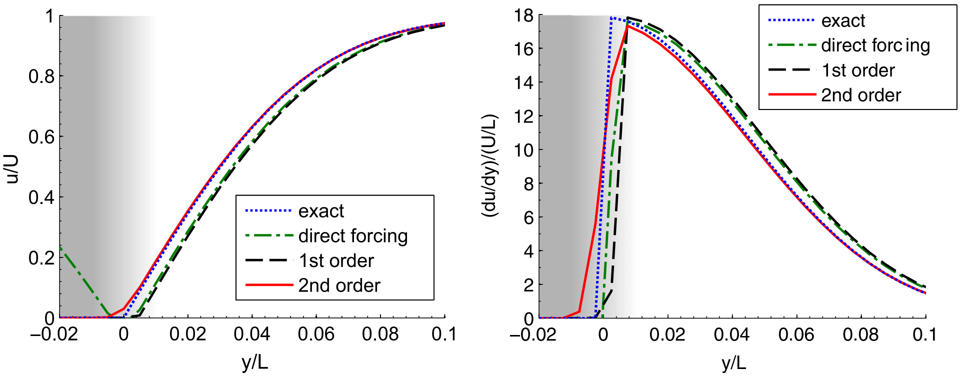
Fig. 4 shows that for Reynolds number $Re = L^2/(\nu t)$ between 100 and 10 000, e_∞ and e_2 are only functions of the ratio between the grid spacing dy and δ_{99}. The L_∞ norm of all three methods converges at first order when the points in the smoothing region are included. However, excluding the first point off the boundary (setting $p = \epsilon$ in Eq. (23)), or using the L_2 norm, the *2nd order* BDIM shows quadratic convergence.

Analysis of the limiting case $\nu = 0$, detailed in Appendix B.3, can help understand these convergence results. *1st order* BDIM and direct forcing have the same fixed point solution $u_\epsilon(y) = 0$ for $|d(y)| < \epsilon$. The smoothed solution calculated with these methods is as sharp as the exact solution, with the discontinuity displaced ϵ into the fluid. This phenomenon can also be observed at $Re = 1000$ in Fig. 5. At large Reynolds number, interpolation is not enough to ensure appropriate smooth transition from the solid velocity to the fluid one, which results in linear convergence even outside of the transition region. Increasing the width of the kernel would not provide much additional smoothing, as these first order methods cannot take advantage of the whole transition region, and would increase the kernel dependent error.

For the proposed *2nd order* BDIM, the fixed point solution to the infinite Reynolds number case is

$$u_\epsilon(y) = \exp\left(\frac{1 - \mu_0^\epsilon}{\mu_1^\epsilon} y - 1\right) \quad \text{for } |d(y)| < \epsilon. \tag{25}$$

This solution transitions smoothly from $u_\epsilon(y) = 0$ at $y = -\epsilon$ to $u_\epsilon(y) = 1$ at $y = \epsilon$. The normal derivative term allows *2nd order* BDIM to fully take advantage of the 2ϵ wide transition region, resulting in a smoother velocity at the interface. This results in a significantly reduced error and second-order convergence in the external flow. The smoothness also ensures that discrete differential operators still approximate their continuous counterpart, which helps prevent artificial pressure fluctuations and flow instabilities as will be illustrated in Section 3.2.



(a) Velocity profile.

(b) Velocity derivative.

Fig. 5. Exact, direct forcing, *1st order* and *2nd order* BDIM velocity profiles at the wall (located at $y = 0$) in an unsteady one-dimensional channel flow of height L for $Re = L^2/(\nu t) = 1000$. The solid region is colored in gray, the fluid region in white. The smoothing region that extends from $y = -0.01L$ to $y = 0.01L$ is represented by a gradient of gray. The new *2nd order* method predicts a velocity that very closely matches the exact solution outside of the smoothing region. The boundary is halfway between two grid points.

Finally, we note that, in this one-dimensional flow, *1st order* BDIM can be considered a type of direct forcing method (though different from that described in [27]) because this example is missing a mass conservation equation. Indeed, Fig. 5 shows that they result in very similar velocity profiles, whereas the *2nd order* BDIM profile is much closer to the exact solution, especially as $\nu \to 0$.

2.4. Flow solver

We will now apply the new *2nd order* BDIM formulation to our fluid–solid interaction problem. Substituting Eq. (4) into Eq. (17) results in the momentum conservation equation for a general fluid–solid interaction problem

$$\vec{u}_\epsilon(t_0 + \Delta t) = \mu_0^\epsilon \left(\vec{u}_\epsilon^0 + \vec{R}_{\Delta t}(\vec{u}_\epsilon) - \partial\vec{P}_{\Delta t}\right) + (1 - \mu_0^\epsilon)\vec{V} + \mu_1^\epsilon \frac{\partial}{\partial n}\left(\vec{u}_\epsilon^0 - V + \vec{R}_{\Delta t}(\vec{u}_\epsilon) - \partial\vec{P}_{\Delta t}\right). \tag{26}$$

Since the fluid is incompressible, the velocity field has to be divergence-free in the fluid region at all time

$$\vec{\nabla} \cdot \vec{u} = 0 \quad \text{for } \vec{x} \in \Omega_f. \tag{27}$$

The corresponding equation in the body is trivial

$$\vec{\nabla} \cdot \vec{u} = \vec{\nabla} \cdot \vec{b} \quad \text{for } \vec{x} \in \Omega_b. \tag{28}$$

If the divergence of the velocity is zero inside the body, then the full meta-equation automatically enforces a divergence-free $\vec{u}_\epsilon$. However, for general deforming bodies (like the shrinking cylinder example from [29]), the smoothing equation needs to be applied in order to resolve the discontinuity in $\vec{\nabla} \cdot \vec{u}$. Applying Eq. (17) to $\vec{\nabla} \cdot \vec{u}$ leads to the following generalized equation

$$\vec{\nabla} \cdot \vec{u}_\epsilon = (1 - \mu_0^\epsilon)\, \vec{\nabla} \cdot \vec{b} - \mu_1^\epsilon \frac{\partial}{\partial n}\left(\vec{\nabla} \cdot \vec{b}\right) \quad \text{for } \vec{x} \in \Omega. \tag{29}$$

Note that the mass conservation equation is applied to $\vec{\nabla} \cdot \vec{u}_\epsilon$, whereas direct forcing methods usually apply it to $\vec{f}$. Substituting in Eq. (26) gives the following mass conservation equation

$$\vec{\nabla} \cdot \left(\mu_0^\epsilon \partial\vec{\tilde{P}}_{\Delta t}\right) = \vec{\nabla} \cdot \left(\mu_0^\epsilon \left(\vec{u}_\epsilon^0 + \vec{R}_{\Delta t}(\vec{u}_\epsilon)\right) + \mu_1^\epsilon \frac{\partial}{\partial n}\left(\vec{u}_\epsilon^0 + \vec{R}_{\Delta t}(\vec{u}_\epsilon)\right)\right) - \left(\vec{V} \cdot \vec{\nabla}\right)\mu_0^\epsilon - \left(\frac{\partial \vec{V}}{\partial n} \cdot \vec{\nabla}\right)\mu_1^\epsilon \tag{30}$$

where we define a modified pressure $\tilde{P}$ implicitly as

$$\vec{\nabla} \cdot \left(\mu_0^\epsilon \partial\vec{\tilde{P}}_{\Delta t}\right) = \vec{\nabla} \cdot \left(\mu_0^\epsilon \partial\vec{P}_{\Delta t} + \mu_1^\epsilon \frac{\partial}{\partial n}\partial\vec{P}_{\Delta t}\right). \tag{31}$$

We use $\tilde{P}$ instead of P on the left hand side of Eq. (30) to avoid inversion of a non-symmetric third-order pressure equation. As this variable substitution does not affect the right hand side, it does not introduce any error in the velocity field divergence and the error in the pressure field will not propagate in time. Eq. (31) shows that P only differs from $\tilde{P}$ through a second-order term. The difference P_1 between P and $\tilde{P}$ can be approximately estimated from $\tilde{P}$ by solving the following equation

$$\vec{\nabla} \cdot \left(\mu_0^\epsilon \partial \vec{P}_{1\,\Delta t} \right) = \vec{\nabla} \cdot \left(\mu_1^\epsilon \frac{\partial}{\partial n} \partial \vec{\tilde{P}}_{\Delta t} \right). \tag{32}$$

As will be shown in Section 3.1, P_1 is indeed quadratically driven to zero with grid refinement.

Eqs. (26) and (30) form the smoothed governing equations of the coupled fluid–solid system. They ensure exact mass and momentum conservation as well as the smoothness necessary for the discrete operators to approximate the continuous ones. The governing equations (26) and (30) can be implemented with any computational scheme. We have implemented them using an Euler explicit integration scheme with Heun's corrector. The following equations are solved in order to calculate $\vec{u} = \vec{u}_\epsilon(t_0 + \Delta t)$ from $\vec{u}_0 = \vec{u}_\epsilon(t_0)$ and $\vec{V} = \vec{V}(t_0 + \Delta t)$.

Euler integration with pressure correction

$$\vec{u}' = \mu_0^\epsilon \; (\vec{u}_0 + \vec{r}_{\Delta t}(\vec{u}_0)) + (1 - \mu_0^\epsilon)\; \vec{V} + \mu_1^\epsilon \; \frac{\partial}{\partial n} \left(\vec{u}_0 + \vec{r}_{\Delta t}(\vec{u}_0) - \vec{V} \right) \tag{33a}$$

$$\Delta t \; \vec{\nabla} \cdot \left(\frac{\mu_0^\epsilon}{\rho} \; \vec{\nabla} p_0 \right) = \vec{\nabla} \cdot \vec{u}' - (1 - \mu_0^\epsilon)\; \vec{\nabla} \cdot \vec{V} \tag{33b}$$

$$\vec{u}_1 = \vec{u}' - \frac{\Delta t}{\rho} \; \mu_0^\epsilon \; \vec{\nabla} p_0. \tag{33c}$$

Heun's corrector

$$\vec{u}_1' = \mu_0^\epsilon \; (\vec{u}_0 + \vec{r}_{\Delta t}(\vec{u}_1)) + (1 - \mu_0^\epsilon)\; \vec{V} + \mu_1^\epsilon \; \frac{\partial}{\partial n} \left(\vec{u}_0 + \vec{r}_{\Delta t}(\vec{u}_1) - \vec{V} \right) \tag{34a}$$

$$\Delta t \; \vec{\nabla} \cdot \left(\frac{\mu_0^\epsilon}{\rho} \; \vec{\nabla} p_1 \right) = \vec{\nabla} \cdot \vec{u}_1' - (1 - \mu_0^\epsilon)\; \vec{\nabla} \cdot \vec{V} \tag{34b}$$

$$\vec{u}_2 = \vec{u}_1' - \frac{\Delta t}{\rho} \; \mu_0^\epsilon \; \vec{\nabla} p_1 \tag{34c}$$

$$\vec{u} = \frac{1}{2} \left(\vec{u}_1 + \vec{u}_2 \right) \tag{34d}$$

where for incompressible Navier–Stokes,

$$\vec{r}_{\Delta t}(\vec{u}) = \Delta t \left[- \left(\vec{u} \cdot \vec{\nabla} \right) \vec{u} + \nu \nabla^2 \vec{u} \right]. \tag{35}$$

These equations have been implemented in an Implicit Large Eddy Simulation (ILES) code (see [30] for a discussion on ILES). They are posed on a staggered mesh and central differences are used for all spacial derivatives except in the convective term in $\vec{r}_{\Delta t}$ which uses a flux limited QUICK scheme for stability. When the local flow is well resolved, these equations automatically revert to a non-dissipative (central difference) scheme. The only novel computations required by the *2nd order* formulation are in steps (33a) and (34a) to add the μ_1^ϵ derivative term on the right hand side. This normal derivative term is computed by calculating the gradient using a second-order central difference at all points. The gradient is then projected on the outward normal to the closest boundary. Our experience is that this makes up less than 1% of the simulation cost and, as shown in the next section, enables accurate predictions of high Reynolds number flows.

3. Application to fluid–solid systems

In this section, two and three-dimensional flows relevant to animal and vehicle locomotion are used to demonstrate the versatility and accuracy of the *2nd order* BDIM formulation and its suitability for practical intermediate Reynolds number applications.

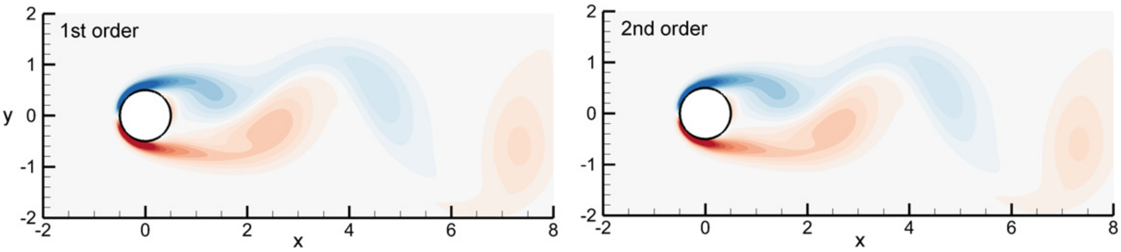


Fig. 6. Flow past a stationary cylinder at $Re = 100$, instantaneous vorticity for the *1st* and *2nd order* BDIM formulations.

First a grid convergence study is carried out on the canonical case of two-dimensional flow past a static cylinder at low Reynolds number in Section 3.1. A method to calculate the forces on a body is also presented and tested on that flow. The two following examples focus on airfoils at Reynolds numbers between 10^4 and 10^5, since many potential applications of IBs (from industrial applications to the study of animal locomotion) involve airfoils for producing lift or thrust. The first example in Section 3.2 consists in a SD7003 airfoil at 4° angle of attack and Reynolds numbers $Re = 10\,000$ and 22 000. In this very challenging example we show that a careful treatment of sharp edges dramatically improves the flow predictions near the trailing edge. In Section 3.3, a heaving and pitching NACA0012 airfoil at Reynolds number $Re = 10^5$ is simulated. Finally, in Section 3.4, the method is applied to a three dimensional example for which a body-fitted simulation would be highly impractical.

3.1. Two-dimensional flow past a stationary cylinder at low Reynolds number

The canonical case of two-dimensional flow past a static cylinder is first considered in order to assess the numerical properties of the proposed method. The flow is simulated in a 29 × 29 diameter D domain, constant velocity $u = U$ on the inlet, upper and lower boundaries, and a zero gradient exit condition with global flux correction. The grid is uniform near the cylinder with spacing $dx/D = 1/120$ and uses a 1% geometric expansion ratio for the spacing in the far-field. In this example and in the rest of the paper, the radius of the smoothing kernel is chosen as twice the grid size $\epsilon = 2\,dx$. The following studies in this section show that this level of smoothing is ideal.

As shown in Fig. 6, both *1st* and *2nd order* BDIM methods show the same characteristic vortex shedding pattern on this simple example. In order to quantify the error evolution with grid refinement, the grid size is parametrized by parameter h such that the spacing is $dx/D = h/120$. Since an exact solution for this flow does not exist, we use the solution computed on a highly resolved grid ($h = 1$) as a baseline for computing the error. The same flow is then computed for $h = [3,\ 4,\ 6,\ 8,\ 12]$, and the velocity and pressure errors are shown on log–log plots in Fig. 7 for *1st* and *2nd order* BDIM, as well as direct forcing. Also included in the figure are dotted lines denoting linear convergence and dashed lines denoting quadratic convergence. On all plots, the errors for *1st order* BDIM and direct forcing are at least twice as large as the error for *2nd order* BDIM. The order of convergence of the direct forcing method is between 1 and 1.5 for both velocity and pressure in the L_2 and L_∞ norms. For *1st order* BDIM, the order of convergence of the velocity in L_∞ norm is also between 1 and 1.5, whereas in L_2 norm and for the pressure in both norms, the convergence is close to second-order. Note also that whereas *1st order* BDIM and direct forcing have almost the same velocity error, the pressure error decreases faster with *1st order* BDIM than direct forcing. Finally, the *2nd order* BDIM errors all converge quadratically but for the pressure error in L_2 norm that converges linearly in the range of grid spacing used. In Fig. 7(d) and (e), the norm of the pressure correction P_1 defined by Eq. (32) is also plotted and converges quadratically for both norms. We remark here that in this practical case the orders of convergence do not exactly match those estimated in Section 2.3 on the one-dimensional channel example. The addition of the pressure term, the smaller range of grid spacing and the impact of time convergence are potential reasons for the observed differences.

We have also tested how the width of the kernel affects the accuracy of the computed solution. For $h = 4$, Fig. 7(c) and (f) show the L_2 norm of the velocity and pressure error for ϵ/dx from 0.5 to 4. For ϵ/dx greater than 2, the error decreases quadratically with the kernel radius. Decreasing ϵ/dx from 2 to 1 hardly reduces pressure error and increases the velocity error; further decreasing ϵ/dx results in an increased error. These plots show that the choice of ϵ/dx is a trade-off between limiting the diffusion caused by a large kernel radius while ensuring enough smoothness for the discrete differential operators to be accurate. A radius of two grid points is the best compromise and has been used throughout this paper.

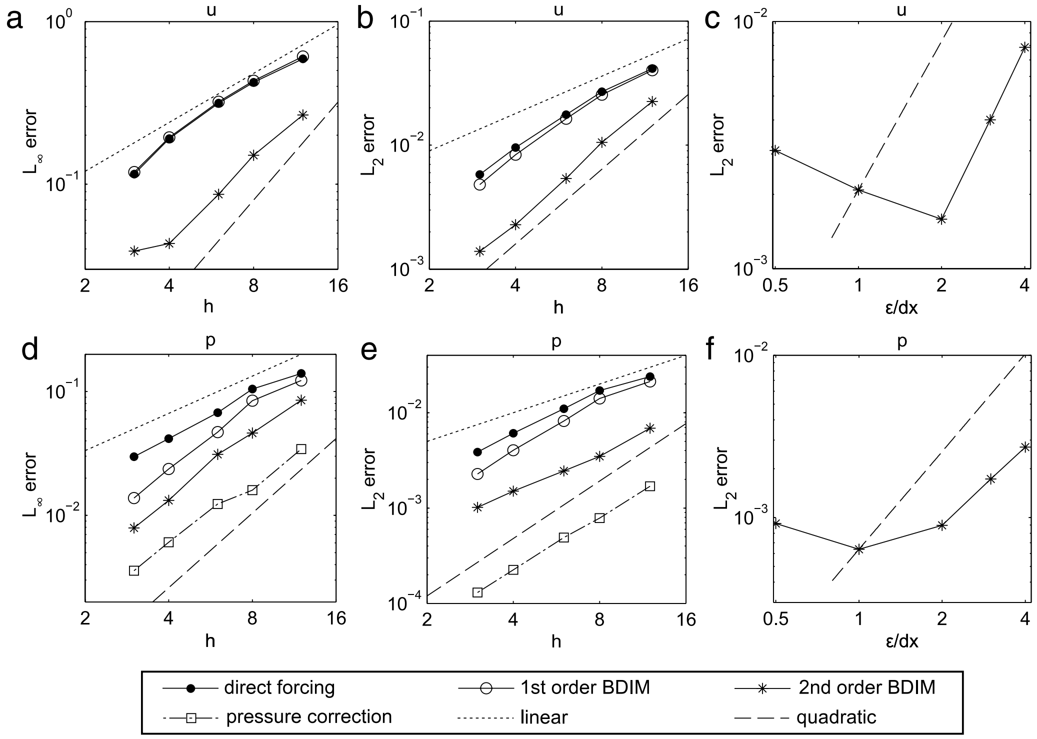


Fig. 7. L_∞ and L_2 norms of the velocity (a–c) and pressure (d–f) error versus grid size (a–b, d–e) and kernel radius (c, f). The grid spacing is $dx/D = h/120$.

Forces are calculated using a one-sided Derivative Informed Kernel (DIK) derived in [37]. The expression for the pressure force uses the Neumann pressure boundary condition and has the form

$$\vec{F}_p = \int_\Omega \left(p - d\frac{\partial p}{\partial n}\right) \hat{n}\delta^+_\epsilon \mathrm{d}\vec{x} \tag{36}$$

where δ^+_ϵ is a kernel designed to sample in the fluid near the surface. For this work we used $\delta^+_\epsilon(d) = \phi_\epsilon(d,\epsilon)/(1 + d/R)$ with ϕ_ϵ the kernel defined in Eq. (15) and R the radius of curvature. Similarly, the friction force is estimated using

$$\vec{F}_f = \int_\Omega \rho\, \nu\, d\, \hat{n} \cdot \left(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^{\,T}\right) \delta^+_\epsilon \mathrm{d}\vec{x}. \tag{37}$$

The advantage of the DIK method is that it evaluates the unsteady forces on the body in one step without a surface grid. Note however that the forces are evaluated at a distance ϵ from the body. While this does not affect the accuracy of the pressure force, the friction force will be underestimated if the width of the linear region of the boundary layer is less than ϵ. However, as noted by Pourquie [38], this limitation is common to most IB methods. The mean drag (C_D) and lift (C_L) coefficients calculated using the DIK method, as well as the individual contribution from pressure and friction (indicated by subscripts p and f respectively) are compared to body-fitted simulations from Park [31] and Henderson [32] in Table 1. In the same table, results are also compared to other Cartesian-grid methods [33,34] and experimental measurements [35,36]. For this simple geometry low Reynolds number example, both *1st* and *2nd order* BDIM compare well with documented results, validating the force calculation approach. The mean forces have also been calculated using a control volume, and the results match the DIK average values exactly.

3.2. Flow around a stationary SD7003 airfoil

Next the *1st* and *2nd order* formulations of BDIM are applied to a more challenging high Reynolds number streamlined body case. The two-dimensional flow past a stationary SD7003 airfoil at 4° angle of attack is computed in

Table 1
Simulated and experimental measurements of the shedding frequency (St) and the mean drag (C_D) and lift (C_L) on a circular cylinder at $Re = 100$ compared to *1st* and *2nd order* BDIM with $dx/D = 1/120$ and $\epsilon/dx = 2$. Experimental St is from [35] with an estimated uncertainty of 0.8%; experimental C_D is from [36], with an estimated error of 6%.

Source	St	C_D	C_L	C_{Dp}	C_{Df}	C_{Lp}	C_{Lf}
Exp.	0.164	1.25	–	–	–	–	–
[31]	0.165	1.33±0.009	±0.332	0.99±0.0082	0.34±0.0010	±0.295	±0.042
[32]	–	1.33	–	1.00	0.32	–	–
[33]	0.164	1.34±0.011	±0.315	–	–	–	–
[34]	0.167	1.35±0.012	±0.303	–	–	–	–
1st order	0.167	1.31±0.009	±0.321	1.01±0.0085	0.30±0.0008	±0.292	±0.035
2nd order	0.167	1.31±0.009	±0.313	1.00±0.0081	0.30±0.0007	±0.285	±0.034

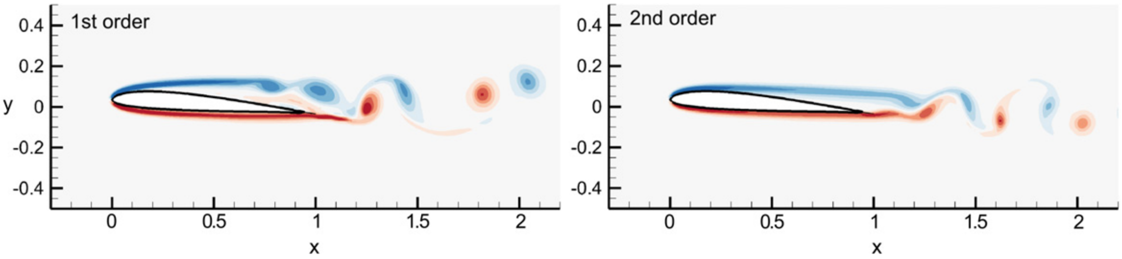


Fig. 8. Flow past a stationary SD7003 airfoil at 4° angle of attack and $Re = 10\,000$ with grid size $dx = 1/100$ ($h = 2$). Instantaneous vorticity for the *1st* and *2nd order* BDIM formulations. Only the *2nd order* method successfully predicts the regular vortex shedding pattern expected for this test case.

a 15×20 chord lengths domain. Constant velocity $\vec{u} = \vec{U}$ on the inlet, upper and lower boundaries, and a zero gradient exit condition with global flux correction are used. The grid spacing is set to $200/h$ points per chord length near the airfoil (corresponding to $16/h$ points across the thickness of the airfoil) with a 1% geometric expansion ratio for the grid spacing in the far-field. The thin boundary layer, low curvature separation and very sharp trailing edge make this case extremely challenging for Cartesian-grid methods. We first consider a Reynolds number (based on the chord c) $Re = 10\,000$ at which the boundary layer is expected to remain laminar along 94% of the chord and the wake to display periodic vortex shedding [39,40]. Since [39] reported that at this Reynolds number 2D and 3D curves for average pressure and stream-wise skin friction coefficients are indistinguishable, two-dimensional simulations are used.

Fig. 8 shows instantaneous vorticity fields computed by both BDIM formulations for $h = 2$ and $\epsilon/dx = 2$. Whereas the *2nd order* method shows laminar separation and periodic vortex shedding as expected at this Reynolds number (detailed in [39,40]), the *1st order* one shows vortices forming on the upper surface of the foil. This example compared to the previous one illustrates the fact that the local accuracy assumes a much greater importance at high Reynolds number, especially when low curvature separation is involved. At low Reynolds number, Fig. 6 shows that the lower and higher order methods predict similar results. However, on this more challenging high Reynolds number example, the lower order method fails to predict the proper qualitative behavior because a higher order treatment of the boundary is necessary to address the large discontinuity in the velocity derivative illustrated in Fig. 1(b).

A grid refinement study has been performed in order to establish the convergence properties of the methods on this high Reynolds number streamlined body case. The grid spacing parameter h is decreased from $h = 6$ to $h = 0.5$, corresponding to 33 and 400 points per chord length respectively (2.5 and 32 points across the thickness of the airfoil respectively), and $\epsilon/dx = 2$ is used. Fig. 9 shows the time-averaged drag and lift coefficients (respectively C_D and C_L) on the airfoil as well as the reduced vortex shedding frequency k (based on the chord) compared to body fitted ILES results from Uranga [39]. The *1st order* method does not seem to converge to the expected solution until the finest level. Indeed, despite the smoothing introduced by μ_0 the zeroth order moment of the kernel, the solution remains in the wrong regime, even with 200 grid points per body length, due to the large jump in the velocity derivative at the immersed boundary. On the other hand, the *2nd order* method converges steadily towards values that are consistent with Uranga [39] (force coefficients and separation location reported by other authors [40] are within 5% of values from [39]).

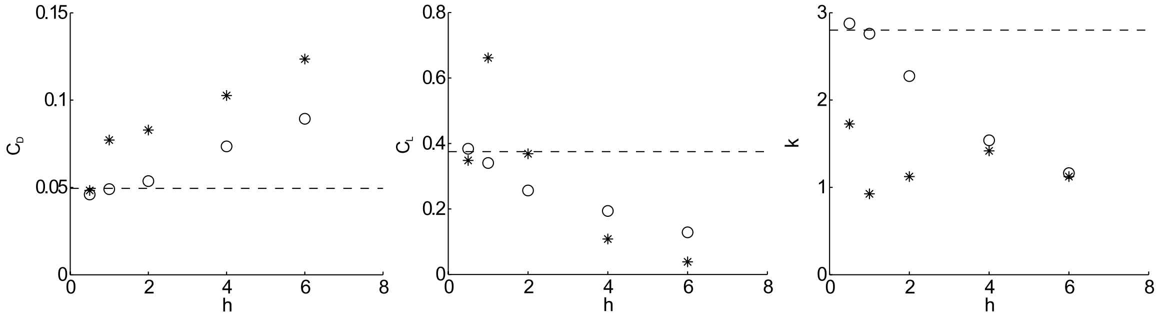


Fig. 9. Convergence of *1st order* ($*$) and *2nd order* ($\circ$) BDIM for flow past a stationary SD7003 airfoil at $Re = 10\,000$. Drag (C_D) and lift (C_L) coefficients, as well as reduced vortex shedding frequency ($k = \omega c/(2\pi U)$) are compared to values from Uranga [39] (dashed lines).

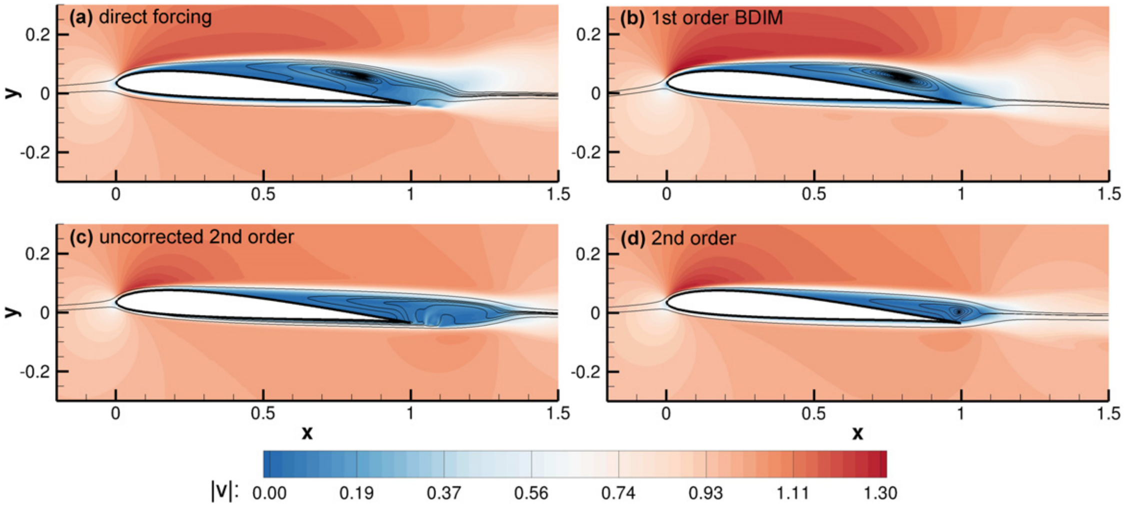


Fig. 10. Flow around a SD7003 airfoil at 4° angle of attack and $Re = 10\,000$ with $h = 1$. Time-averaged velocity magnitude and streamlines.

The grid size $h = 1$ is chosen to further investigate the challenges associated with this example and the importance of carefully treating sharp corners with IB methods. The formulation derived above (Eq. (13)) assumes that the IB is locally flat but it can easily be extended to account for a sharp corner (see derivation in Appendix A). Fig. 10 shows the time-averaged velocity magnitude field and streamlines for four formulations: (a) direct forcing, (b) *1st order* BDIM, (c) the *uncorrected 2nd order* derived in the previous sections, and (d) the *2nd order* with the sharp corner treatment derived in Appendix A. The direct forcing used here applies the *1st order* BDIM equation for the momentum conservation and the unmodified mass conservation equation (Eq. (30) with $\mu_0^\epsilon = 1$ and $\mu_1^\epsilon = 0$). As has been observed earlier, the lower order methods (a and b) are unable to provide appropriate smoothing at this Reynolds number, causing instabilities to develop over the upper surface of the wing instead of further down in the wake. As a result the separation bubble is swollen and the velocity above it remains high until the trailing edge. In contrast, the second-order formulations show that the velocity rapidly recovers after the leading edge, as observed in [39]. However, the velocity field of the *uncorrected 2nd order* method shows that without a careful sharp corner treatment, the low velocity region extends too far downstream with unphysical fluctuations and streamlines. The simple analytic extension for sharp corners derived in Appendix A solves these issues and the *2nd order* velocity fields and streamlines closely match those reported by Uranga [39] and Castonguay [40]. This example makes it clear that the second-order formulation of BDIM significantly improves flow predictions in cases where the grid does not fully resolve the geometry nor the boundary layer.

Fig. 11(a) shows the average pressure coefficient along the airfoil for $h = 1$. The pressure coefficient predicted by the *2nd order* BDIM compares very well with Castonguay [40], whereas for the same grid size, the first order methods

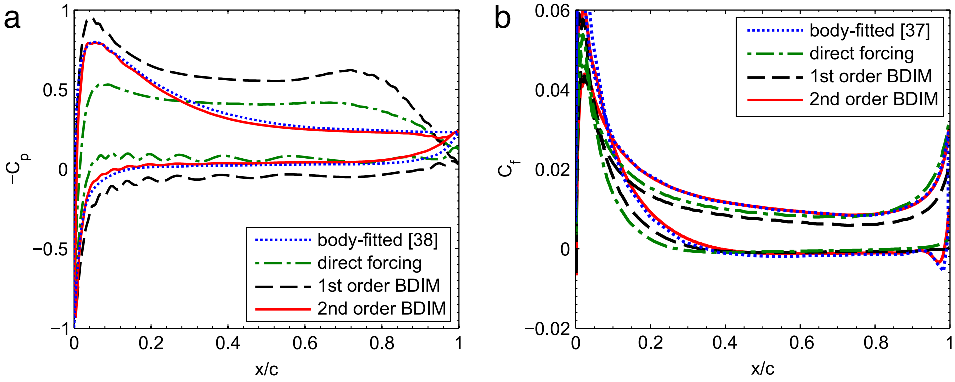


Fig. 11. Average pressure (C_p) and skin friction (C_f) coefficients around a SD7003 airfoil at 4° angle of attack and $Re = 10\,000$ with $h = 1$. $dy = dx/4$ for (b).

(direct forcing and *1st order* BDIM) show large pressure fluctuations on the high pressure side and a plateau around mid-foil on the low pressure side that is not expected at this Reynolds number [39,40].

Since calculation of the skin friction is carried out ϵ away from the boundary, it is accurate only if the viscous sublayer is thicker than ϵ, which is not the case in the simulations above. In order to accurately calculate the skin friction coefficient and separation point for this stationary, low angle of attack airfoil, a finer grid in the cross-flow direction is needed. Here we used the $h = 1$ grid and increased the density in the y direction by a factor of 4. Correcting for the fact that $u_\epsilon(0) = \mu_1^\epsilon \partial u_\epsilon/\partial n$ (from substituting exact f and b solutions in Eq. (17)), we calculated the skin friction as

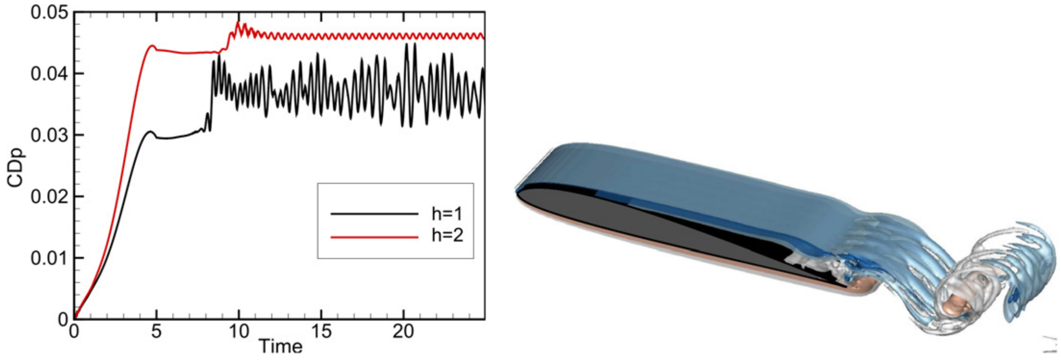
$$C_f(\vec{x}) = \frac{2\nu}{U^2}\frac{\vec{u}_\epsilon(\vec{x}+\epsilon\hat{n})\cdot\hat{n}}{\epsilon+\mu_1^\epsilon(0)}, \tag{38}$$

where $\vec{u}_\epsilon(\vec{x}+\epsilon\hat{n})$ is linearly interpolated from the grid and $\mu_1^\epsilon(0) = 0$ for *1st order* BDIM and direct forcing. As shown in Fig. 11(b), only *2nd order* BDIM compares well with [39]. We are able to accurately predict the skin friction, the location of separation (around $x/c = 0.38$ versus $x/c = 0.37$ for Uranga [39]) and even the transition of the boundary layer to turbulent as indicated by the sudden dip in C_f around $x/c = 0.94$.

This example has been chosen to illustrate the main challenges faced by Immersed Boundary methods: accurately simulating thin boundary layers separating over a low curvature surface and ending with a very sharp edge. We show here that, unlike *1st order* methods, *2nd order* BDIM is able to accurately predict the pressure distribution around the airfoil, without spurious fluctuations, in a way that steadily converges with grid refinement. Using a finer grid in the cross-flow direction, it also provides good prediction of the skin friction and separation. If skin friction is of interest, local grid refinement [28] or a wall-model [41–43] can also be used instead of a global grid refinement in order to reduce the computational cost. However, in practice, IB methods are of interest to simulate moving boundaries, in which case the friction forces are much smaller than the pressure forces.

Next, a Reynolds number $Re = 22\,000$ is considered, at which the separated boundary layer is expected to become turbulent around $x/c = 0.7$ [39]. In this regime, a three-dimensional simulation is required to capture the main flow features and a domain of $0.3c$ is used in the span-wise direction. As illustrated in the time history of pressure drag coefficient for $h = 1$ and $h = 2$ in Fig. 12(a), *2nd order* BDIM naturally predicts the three dimensional wake as soon as the grid spacing is smaller than the viscous sublayer ($dy^+ = 7$ for $h = 1$). Fig. 12(b) shows instantaneous span-wise vorticity iso-surfaces computed using *2nd order* BDIM and the grid size $h = 1$.

Compared to the canonical low Reynolds number flow past a cylinder, the present example combines three additional complexities: (i) a high Reynolds number, (ii) a low curvature and (iii) a sharp edge. This flow presents most of the difficulties encountered in practical fixed or rotating wings applications while being well documented, which makes it an excellent benchmark. We have shown that this flow is very sensitive to the treatment of the IB: the first order IB treatments (either BDIM or direct forcing) are unable to capture the physics of the flow. An appropriate treatment of the sharp trailing edge also dramatically improves the flow predictions and allows *2nd order* BDIM to accurately capture the integrated forces, pressure distribution, flow separation and skin friction for this challenging test case.



(a) Time history of pressure drag coefficient for two grid resolutions.

(b) Instantaneous span-wise vorticity iso-surfaces computed with $h = 1$.

Fig. 12. Flow around a SD7003 airfoil at 4° angle of attack and $Re = 22\,000$.

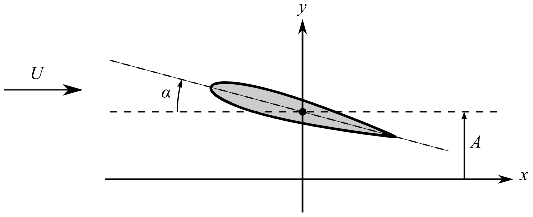


Fig. 13. Definition of heaving and pitching motion.

3.3. Flow around a heaving and pitching NACA0012 airfoil

In order to illustrate the application of BDIM to moving IBs, we apply it to a heaving and pitching NACA0012 airfoil at Reynolds number $Re = 10^5$ (based on the chord c and free-stream velocity U). The combination of heaving and pitching motion of a foil, known as flapping, is at the core of aerial and underwater animal locomotion. The position of the foil at time t is defined by A the vertical position of its pitch axis located at mid-chord of the foil and α its angular displacement (see Fig. 13), which are expressed as

$$\begin{cases} A/c = A_0 \cos kt \\ \alpha = \alpha_0 \cos(kt + \phi) \end{cases} \tag{39}$$

where the reduced frequency k is expressed in radians and the time t nondimensionalized by U and c. A_0 represents the amplitude of the heaving motion, α_0 the pitching amplitude and ϕ the phase between the two.

A high thrust producing case investigated by [44] is chosen:

$$A_0 = 1, \quad \alpha_0 = 10°, \quad \phi = \pi/2, \quad k = 1. \tag{40}$$

The grid spacing is set to 67 points per chord length (8 points across the thickness of the foil) in the region swept by the flapping foil with a 1% geometric expansion ratio for the grid spacing in the far-field. The domain extends 6 chord lengths upstream, 8 downstream and 9 chord lengths on either side in the cross flow direction. As in the previous examples, constant velocity $\vec{u} = \vec{U}$ on the inlet, upper and lower boundaries, and a zero gradient exit condition with global flux correction are used.

The mean drag coefficient (drag force normalized by $\rho U^2 c/2$) estimated with the *1st* and *2nd order* BDIM are compared to direct forcing and [44] for this case and two other parameter sets in Table 2. The role of the pressure equation is more important in dynamic cases than in static cases, therefore BDIM performs much better on this example than direct forcing. However, as noticed by Isogai [45], the detailed treatment of the boundary is much less important for this high frequency flow than for the steady case studied in the previous section. Indeed, Fig. 14 shows that the (dimensionless) drag and lift predictions from the *1st* and *2nd order* formulations are very similar. There are however two main discrepancies between the two: (i) in the lift around $t = 22$ and (ii) in the mean drag as shown in

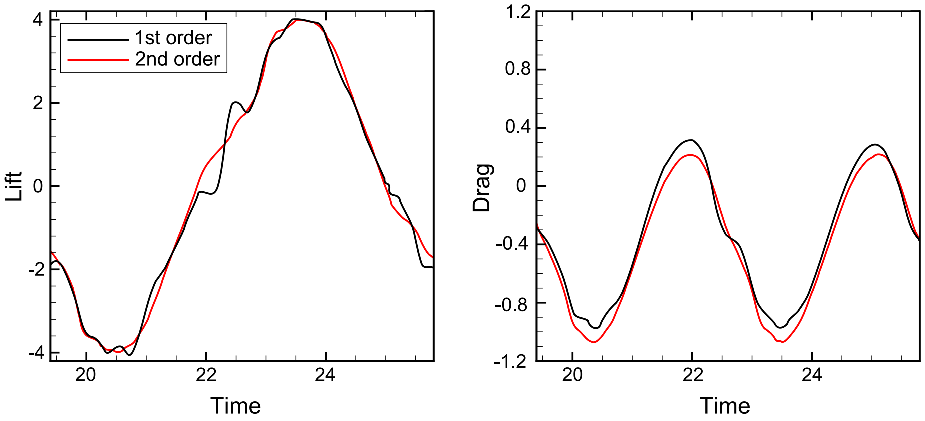


Fig. 14. Lift and drag coefficients on the heaving and pitching NACA0012 at $Re = 10^5$.

Table 2
Mean drag coefficient on the heaving and pitching NACA0012 at $Re = 10^5$ for various combinations of the flapping parameters defined in Eq. (39). Based on comparisons with experiments and other simulations, the error of the results reported in [44] is estimated to be within 15%.

Flapping parameters				Mean drag coefficient			
k	ϕ	α_0	A_0	Direct forcing	1st order	2nd order	Tuncer [44]
1	90°	10°	1	−0.174	−0.358	−0.418	−0.446
1	75°	7°	0.75	−0.061	−0.173	−0.224	−0.29
1.34	75°	10°	0.75	−0.214	−0.338	−0.394	−0.446

Table 2. Whereas the *1st order* formulation underestimates the drag by 20%, the value from the *2nd order* formulation is within 5% of the drag reported by Tuncer [44].

A more detailed analysis of the flow structures is necessary in order to assess the quality of the simulations. To do so, Fig. 15 shows instantaneous vorticity fields. During the upstroke (Fig. 15, $t = 21$) both the *1st* and *2nd order* formulations show the formation of a positive starting vortex due to the high angle of attack of the foil with respect to the flow. However, as the foil reaches the top of its trajectory, the *1st* and *2nd order* predictions diverge (Fig. 15, $t = 22$ and 22.5). Only the *2nd order* formulation is able to predict that the second (positive) vortex remains attached until it reaches the tail of the foil (as observed in [44] and [45]). This accounts for the slight discrepancy in the lift between the two methods shown in Fig. 14.

We have demonstrated the ability of the *2nd order* formulation of BDIM to capture the main flow features generated by a flapping airfoil at Reynolds number $Re = 10^5$. The method has proved to provide accurate force predictions free of spurious fluctuations. This example validates the use of BDIM for the study of flapping foils and more generally highly unsteady flows.

3.4. Multi-body example inspired by fish sensing

The validation cases presented above are typically solved using body-fitted simulations, and therefore do not feature complexities such as interfering bodies in the fluid domain. In this section we demonstrate the ability of our method to handle several three-dimensional bodies with different velocities by simulating a fish-like body passing a circular cylinder as shown in Fig. 16. Fish can sense pressure changes due to passing objects through an organ called the lateral line [46] and use the information to detect and identify obstacles. Similar to [47], our 'fish' is represented by an axisymmetric body of revolution based on a NACA0013 airfoil and a Reynolds number of 6000 is chosen (Reynolds number based on the airfoil length L and free stream velocity U). The computational frame is attached to the fish (whose axis of rotation is $\{y = 0,\ z = 0\}$) such that the fish is represented as stationary on the grid and the cylinder moves with the free stream. The minimum separation distance between the cylinder and the vehicle as well as the radius of the cylinder are chosen to be equal to the thickness of the fish ($0.13L$). The grid spacing is set to 100

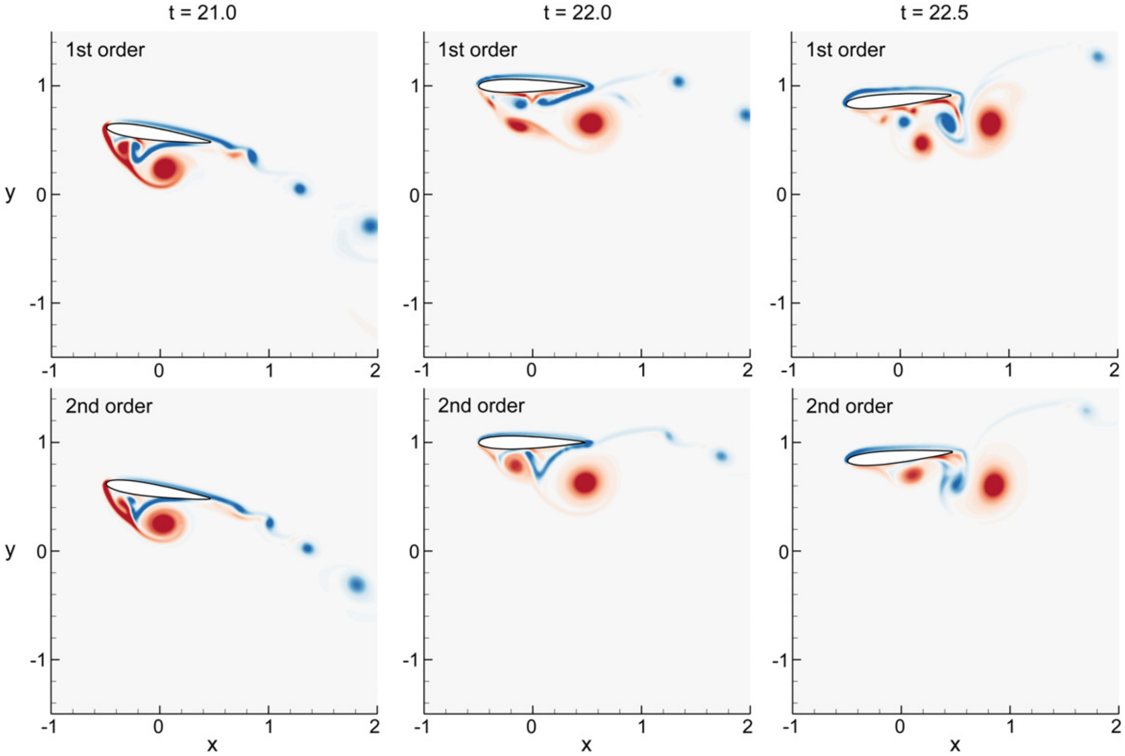


Fig. 15. Instantaneous vorticity fields during heaving and pitching motion of a NACA0012 at $Re = 10^5$ for the *1st* and *2nd order* formulations ($A_0 = c, \alpha_0 = 10°, \phi = \pi/2, k = 1$).

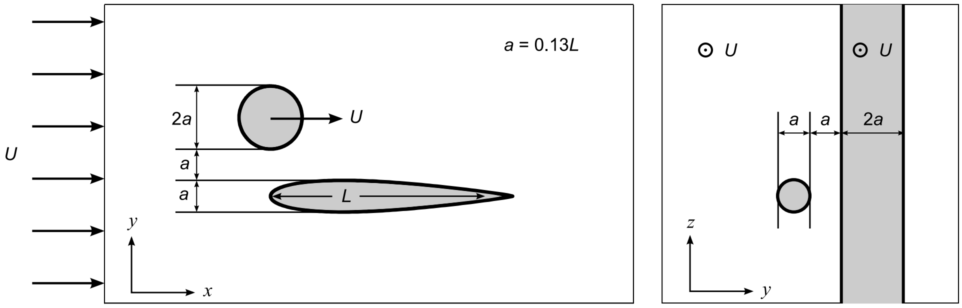
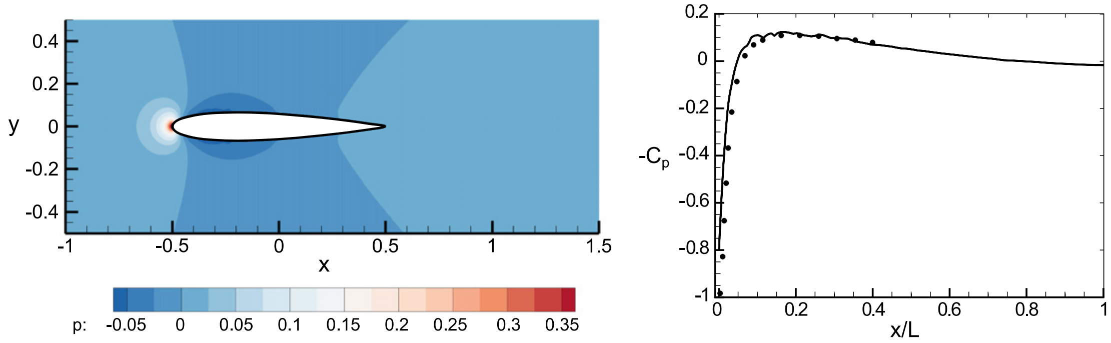


Fig. 16. Three dimensional flow geometry: projection onto the $z = 0$ and $x = 0$ planes.

points per chord length near the fish with a 1% geometric expansion ratio for the grid spacing in the far-field and the computational domain has $10L \times 4L \times 4L$ size. Constant velocity $\vec{u} = \vec{U}$ on the inlet and y boundaries, periodic boundary conditions in z and a zero gradient exit condition with global flux correction are used. The method easily generalizes to multiple immersed bodies by applying Eq. (17) with $\vec{b}(\vec{x})$ the velocity of the closest body and $d(\vec{x})$ the distance to it (used to calculate the μ^ϵ terms).

Fig. 17(a) shows the pressure field around the axisymmetric fish in open water. For easier comparison with body fitted simulations, the pressure coefficient C_p along the surface of the fish are compared to Windsor [47] in Fig. 17(b). Our Cartesian-grid method slightly underestimates the stagnation pressure at the front of the fish but the agreement with the body fitted simulations from Windsor [47] (dots in the Figure) after the front 2% of the fish is very good. Our method also shows very small pressure fluctuations where the boundary crosses the Cartesian grid but, as shown in Fig. 11 on a two-dimensional example, these fluctuations are significantly smaller than with *1st order* methods.



(a) Pressure field in the slice $z = 0$.

(b) Time-averaged pressure coefficient. *2nd order* BDIM (solid line) is compared to Windsor [47] (dots) which has a reported estimated error of 7%.

Fig. 17. Pressure around the axisymmetric fish in open water at $Re = 6000$.

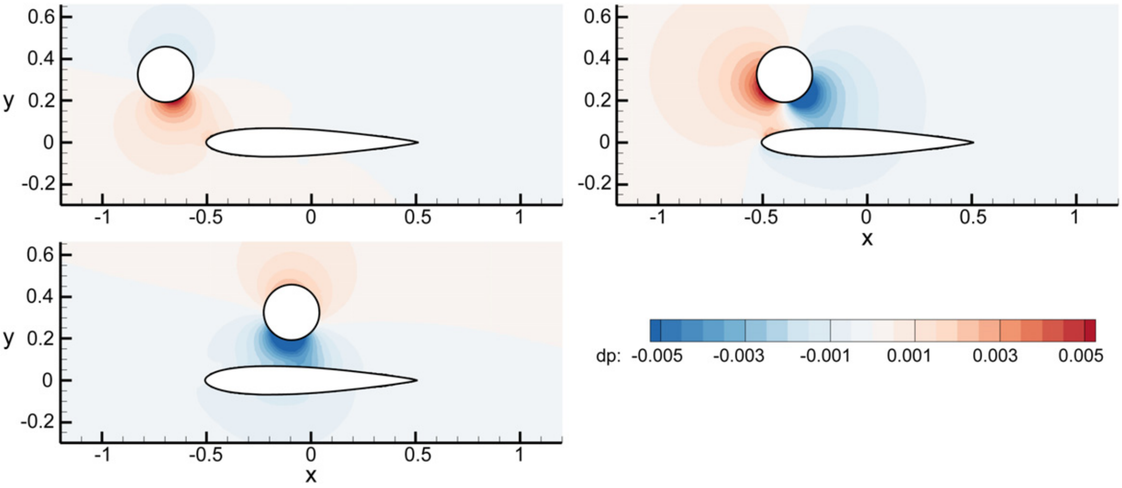


Fig. 18. Instantaneous pressure perturbation field around the axisymmetric fish passing the cylinder (compared with the open water pressure) in the slice at $z = 0$. Pressure is normalized by ρU^2.

Fig. 18 shows the instantaneous pressure perturbation field around the fish passing the cylinder (compared with the open water pressure) in the slice at $z = 0$. The figure shows that as the fish passes the cylinder distinctive pressure changes can be felt by the fish.

This kind of simulation is extremely challenging for body-fitted algorithms and the cost of deforming and/or regenerating the grid as the computational domain undergoes large deformations can exceed the cost of solving the Navier–Stokes equations. BDIM avoids these issues and the complexity and deformation of the geometry do not affect the efficiency of the new second-order simulation method.

4. Conclusion

This paper generalizes the Boundary Data Immersion Method proposed in [21] by establishing a higher order analytic meta-equation. *2nd order* BDIM provides a robust and accurate treatment of IBs in high Reynolds number fluid/solid interaction problems. Our method addresses the issues encountered by first-order methods (including direct forcing methods) at high Reynolds number by adding a higher-order term to the traditional averaging used to estimate velocities near the boundary. The resulting algorithm is both simple to implement in existing Navier–Stokes solvers and computationally efficient.

Applications with Reynolds numbers ranging up to 10^5 are presented: viscous flow past a static SD7003 airfoil and past a flapping NACA0012 airfoil, as well as flow around an axisymmetric fish passing a cylinder. It is shown that the predictions of flow around a slender body with a sharp trailing edge is very sensitive to the IB treatment and that *2nd order* BDIM, with its new sharp edges treatments, can successfully predict it. *2nd order* BDIM has also demonstrated its ability to simulate highly unsteady flows without encountering grid-locking issues as is the case with direct forcing methods [48]. The final examples illustrate the ease of our method to handle three dimensional complex geometries with moving boundaries.

A limitation of the present method lies in the necessity to resolve the boundary layer using a Cartesian-grid in order to accurately predict the skin friction. Considerations of computational cost caused us to limit our studies to $Re \leq 10^5$. Combining BDIM with a local grid refinement technique [28] could improve computational efficiency for the thin shear layers of higher Reynolds number flows. The use of a wall-layer model [41–43] or tangential force model [20] to reduce the required near-wall resolution for very high Reynolds numbers is an active area of research. At higher Reynolds number, the interactions between wall-layer approximation and subgrid-scale model also need to be investigated in the context of immersed boundary as has been done by Temmerman [49] for body-fitted grids.

The ability of *2nd order* BDIM to accurately simulate the viscous flow around complex geometries up to Reynolds numbers of at least $Re = 10^5$ enables a wide range of exciting applications from ocean energy extraction to animal and vehicle locomotion. The robust and smooth simulation of pressures and forces are also especially important in resonant marine systems such as tank sloshing, vortex-induced-vibration reduction, and investigation of biological hydrodynamic sensors such as the lateral line and seal vibressa. As such we believe this simple Cartesian-grid approach based on a strong analytical framework to be a significant contribution to the accurate study of these and other highly non-linear viscous flow systems.

Acknowledgments

This research was supported by the National Research Foundation Singapore through the Singapore MIT Alliance for Research and Technology's Center for Environmental Sensing and Modeling interdisciplinary research program, and by the MIT Sea Grant Program through Professor Michael Triantafyllou. The authors are also grateful to the Link Foundation for their support through the Link Ocean Engineering & Instrumentation Ph.D. Fellowship awarded to Audrey Maertens.

Appendix A. Convolution evaluation at sharp corners

In Section 2.2, kernel moments are analytically evaluated in order to accurately immerse the boundary data from the solid mechanical system onto the fluid mechanical equations of motion. At corners (like sharp trailing edges), the smooth interface assumption introduced in Eq. (13) does not hold. Here, the case of a geometry with sharp corners is locally treated as the intersection of two component planar geometries.

Let us consider a point $\vec{x}$ near a two-dimensional corner defined by two planes. We call w_1 (with normal $\vec{n}_1$ and distance d_1 from $\vec{x}$) the wall closest to $\vec{x}$ and w_2 (with normal $\vec{n}_2$ and distance $d_2 \geq d_1$ from $\vec{x}$) the other wall. The angle between the two planes is θ. Fig. A.19 shows a schematic of the geometry and variables.

Let us define a local coordinate system centered in $\vec{x}$ such that $\hat{j} = \hat{n}_1$ and $\hat{i} \cdot \hat{n}_2 \geq 0$. In this local coordinate system, the equation of w_2 is:

$$\sqrt{1-a^2}x + ay + c = 0 \tag{A.1}$$

with normal $\vec{n} = \left(\sqrt{1-a^2},\ a\right)$. Therefore, $a = \vec{n}_1 \cdot \vec{n}_2$. We also know that the point $-d_2\vec{n}$ belongs to w_2, therefore $c = d_2$.

We can now write:

$$\vec{b}_\epsilon(\vec{u}, \vec{x}) = \int_{\Omega_b} \vec{b}(\vec{u}, \vec{x}_b) K_\epsilon(\vec{x}, \vec{x}_b)\, \mathrm{d}\vec{x}_b \tag{A.2a}$$

$$\approx \int_{\Omega_b} \left(\vec{b}(\vec{u}, \vec{x}) + \vec{\nabla}\vec{b}(\vec{u}, \vec{x}) \cdot (\vec{x}_b - \vec{x})\right) K_\epsilon(\vec{x}, \vec{x}_b)\, \mathrm{d}\vec{x}_b \tag{A.2b}$$

$$\approx \vec{b}(\vec{u}, \vec{x}) \int_{\Omega_b} K_\epsilon(\vec{x}, \vec{x}_b)\, \mathrm{d}\vec{x}_b + \vec{\nabla}\vec{b}(\vec{u}, \vec{x}) \cdot \int_{\Omega_b} (\vec{x}_b - \vec{x})\ K_\epsilon(\vec{x}, \vec{x}_b)\, \mathrm{d}\vec{x}_b \tag{A.2c}$$

$$\approx \vec{b}(\vec{u}, \vec{x})\ \mu_0^{\epsilon,B}(\vec{x}) + \vec{\nabla}\vec{b}(\vec{u}, \vec{x}) \cdot \vec{\mu}_1^{\epsilon,B}(\vec{x}). \tag{A.2d}$$

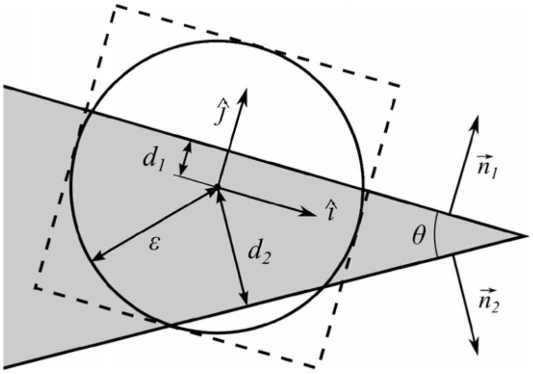


Fig. A.19. Schematic showing the variables used in the derivation of the convolution evaluation at sharp corners.

Using the previously defined local coordinates and assuming a 2D problem (and kernel), we can write (use $d = d_1$):

$$\mu_0(\vec{x}) = \mu_0^{\epsilon,B}(\vec{x}) = \int_{\Omega_b} K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b \tag{A.3a}$$

$$= \int_{y=-\epsilon}^{-d_1}\int_{x=-\sqrt{\epsilon^2-y^2}}^{\sqrt{\epsilon^2-y^2}} \mathbf{1}_{d_2(x,y)<0}\frac{\phi_\epsilon(y)}{2\sqrt{\epsilon^2-y^2}}\,\mathrm{d}x\,\mathrm{d}y \tag{A.3b}$$

$$= \int_{-\epsilon}^{-d_1} \psi_\epsilon(y)\phi_\epsilon(y)\,\mathrm{d}y \tag{A.3c}$$

where

$$\mathbf{1}_{d_2(x,y)<0} = \mathbf{1}_{x<-(ay+c)/\sqrt{1-a^2}} \tag{A.4}$$

therefore

$$\psi_\epsilon(y) = \int_{\min\left(-\sqrt{\epsilon^2-y^2},\ -(ay+c)/\sqrt{1-a^2}\right)}^{\min\left(\sqrt{\epsilon^2-y^2},\ -(ay+c)/\sqrt{1-a^2}\right)} \frac{1}{2\sqrt{\epsilon^2-y^2}}\,\mathrm{d}x \tag{A.5a}$$

$$= \max\left[0,\ \frac{1}{2}+\min\left(\frac{1}{2},\ \frac{-ay-c}{2\sqrt{\epsilon^2-y^2}\sqrt{1-a^2}}\right)\right]. \tag{A.5b}$$

For most functions ϕ, the integral above does not have a closed form solution. To simplify the equation, the integral is simplified by replacing $\sqrt{\epsilon^2-y^2}$ by ϵ. This is equivalent to assuming a kernel with square support instead of circular. Therefore

$$\psi(y) \approx \max\left[0,\ \frac{1}{2}+\min\left(\frac{1}{2},\ \frac{-ay-c}{2\epsilon\sqrt{1-a^2}}\right)\right] \tag{A.6}$$

where

$$\phi(y) = 0.5\,(1+\cos(y\pi/\epsilon))\,. \tag{A.7}$$

Similarly,

$$\vec{\mu}_1(\vec{x}) = \vec{\mu}_1^{\epsilon,B}(\vec{x}) = \int_{\Omega_b} (\vec{x}_b-\vec{x})\,K_\epsilon(\vec{x},\vec{x}_b)\,\mathrm{d}\vec{x}_b \tag{A.8a}$$

$$= \int_{y=-\epsilon}^{-d_1}\int_{x=-\sqrt{\epsilon^2-y^2}}^{\sqrt{\epsilon^2-y^2}} \mathbf{1}_{d_2(x,y)<0}\left(x\hat{\imath}+y\hat{\jmath}\right)\frac{\phi_\epsilon(y)}{2\sqrt{\epsilon^2-y^2}}\,\mathrm{d}x\,\mathrm{d}y \tag{A.8b}$$

$$= \int_{-\epsilon}^{-d_1} \left(\psi_\epsilon^i(y)\hat{\imath}+\psi_\epsilon y\hat{\jmath}\right)\phi_\epsilon(y)\,\mathrm{d}y \tag{A.8c}$$

where

$$\psi_\epsilon^i = \frac{\sqrt{\epsilon^2-y^2}}{2}\min\left[0,\ \frac{(ay+c)^2}{(1-a^2)(\epsilon^2-y^2)}-1\right] \tag{A.9a}$$

$$\approx \frac{\epsilon}{2}\min\left[0,\ \frac{(ay+c)^2}{(1-a^2)\epsilon^2}-1\right]. \tag{A.9b}$$

These equations minimize the modeling error near sharp corners, as shown in Section 3.2.

Appendix B. Derivations for the one-dimensional channel flow

Here we detail some derivations for the unsteady one-dimensional channel flow example analyzed in Section 2.3.

B.1. Exact solution

The exact solution to the unsteady one-dimensional channel flow at Reynolds number $Re = L^2/(\nu t)$ is

$$u(y,t) = U\sum_{k=0}^{\infty} e^{-(2k+1)^2\pi^2/Re}\sin\big((2k+1)\pi y/L\big). \tag{B.1}$$

The sum converges very rapidly and summing over the first 50 terms guaranties an error smaller than 10^{-10}. For Reynolds numbers larger than 100, the velocity in the middle of the channel is not affected by the boundaries, and the 99% boundary layer thickness is given by

$$\delta_{99} = 3.65/\sqrt{Re}. \tag{B.2}$$

B.2. Direct forcing solution

We will now derive a direct forcing formulation (described in [27,11]) of this example. Using the same notations as in Section 2.3, the velocity in direct forcing is expressed as

$$u_\epsilon(y, t_0+\Delta t) = f(u, y, t_0+\Delta t) + g(u, y, t_0+\Delta t) \tag{B.3}$$

where g is a volume force ensuring that $u_\epsilon = 0$ at the boundary ($d(y) = 0$). The volume force is evaluated using a regularized delta function δ_ϵ

$$g(u, y, t_0+\Delta t) = \sum_{Y\in\{0,L\}} G(Y)\delta_\epsilon(y-Y) = -\sum_{Y\in\{0,L\}} F(Y, t_0+\Delta t)\delta_\epsilon(y-Y) \tag{B.4}$$

where capital letters (F and G) refer to values at the boundary. These values are interpolated from the Cartesian grid points using the regularized delta function

$$F(Y, t_0+\Delta t) = \sum_{y'} f(y', t_0+\Delta t)\delta_\epsilon(y'-Y). \tag{B.5}$$

In summary, the direct forcing formulation is

$$u_\epsilon(y, t_0+\Delta t) = f(u, y, t_0+\Delta t) - \sum_{Y\in\{0,L\}} \delta_\epsilon(y-Y)\sum_{y'} f(y', t_0+\Delta t)\delta_\epsilon(y'-Y) \tag{B.6}$$

where the discrete delta function is $\delta_\epsilon(y) = \phi_\epsilon(d(y), 0)\,dy$ for the kernel ϕ_ϵ defined by Eq. (15). Defining the column vector $\zeta_\epsilon = \delta_\epsilon(y) + \delta_\epsilon(y-L)$, the matrix form of the direct forcing formulation is:

$$u_\epsilon(n\Delta t) = \left(\left[1-\zeta_\epsilon\zeta_\epsilon^T\right]\left[1+\Delta t\ \nu\mathbf{D}^{(2)}\right]\right)^n U. \tag{B.7}$$

B.3. Limiting case $\nu = 0$

The fixed point solution of the limiting case $\nu = 0$ for *1st order* BDIM is given by the equation

$$u_\epsilon = \mu_0^\epsilon u_\epsilon. \tag{B.8}$$

Similarly, the fixed point solution for the direct forcing method is given by

$$u_\epsilon = \left[1 - \zeta_\epsilon \zeta_\epsilon^T\right] u_\epsilon. \tag{B.9}$$

In both cases, the fixed point solution verifies $u_\epsilon(y) = 0$ for $|d(y)| < \epsilon$.

For the proposed *2nd order* BDIM, the fixed point solution is given by

$$u_\epsilon = \left[\mu_0^\epsilon + \mu_1^\epsilon \frac{\partial}{\partial y}\right] u_\epsilon. \tag{B.10}$$

In this case, the fixed point solution for $|d(y)| < \epsilon$ is

$$u_\epsilon(y) = \lambda \exp\left(\frac{1 - \mu_0^\epsilon}{\mu_1^\epsilon} y\right), \tag{B.11}$$

where the constant $\lambda = \exp(-1)$ ensures that $u_\epsilon(\epsilon) = 1$.

References

[1] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[2] C.S. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys. 10 (1972) 252–271.

[3] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.

[4] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (1993) 354–366.

[5] R.P. Beyer, R.J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, SIAM J. Numer. Anal. 29 (1992) 332–364.

[6] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math. 81 (1999) 497–520.

[7] E. 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.

[8] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.

[9] F. Muldoon, S. Acharya, A divergence-free interpolation scheme for the immersed boundary method, Internat. J. Numer. Methods Fluids 56 (2008) 1845–1884.

[10] R.D. Guy, D.A. Hartenstine, On the accuracy of direct forcing immersed boundary methods with projection methods, J. Comput. Phys. 229 (2010) 2479–2496.

[11] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448–476.

[12] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, J. Comput. Phys. 228 (2009) 6617–6628.

[13] A. Pinelli, I.Z. Naqavi, U. Piomelli, J. Favier, Immersed-boundary methods for general finite-difference and finite-volume Navier–Stokes solvers, J. Comput. Phys. 229 (2010) 9073–9091.

[14] F. Gibou, R.P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.

[15] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (2008) 4825–4852.

[16] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.

[17] J.H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J. Comput. Phys. 230 (2011) 7347–7363.

[18] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.

[19] H. 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.

[20] J.W. Rottman, K.A. Brucker, D. Dommermuth, D. Broutman, Parameterization of the near-field internal wave field generated by a submarine, in: 28th Symposium on Naval Hydrodynamics, Pasadena, California.

[21] G. Weymouth, D.K. Yue, Boundary data immersion method for Cartesian-grid simulations of fluid–body interaction problems, J. Comput. Phys. 230 (2011) 6233–6247.

[22] G.D. Weymouth, D.K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids, J. Comput. Phys. 229 (2010) 2853–2865.

[23] M.S. Wibawa, S.C. Steele, J.M. Dahl, D.E. Rival, G.D. Weymouth, M.S. Triantafyllou, Global vorticity shedding for a vanishing wing, J. Fluid Mech. 695 (2012) 112–134.

[24] G.D. Weymouth, M.S. Triantafyllou, Global vorticity shedding for a shrinking cylinder, J. Fluid Mech. 702 (2012) 470–487.

[25] G.D. Weymouth, M.S. Triantafyllou, Ultra-fast escape of a deformable jet-propelled body, J. Fluid Mech. 721 (2013) 367–385.

[26] T. Kajishima, S. Takiguchi, H. Hamasaki, Y. Miyake, Turbulence structure of particle-laden flow in a vertical plane channel due to vortex shedding, JSME Int. J. Ser. B 44 (2001) 526–535.

[27] X. Yang, X. Zhang, Z. Li, G.-W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, J. Comput. Phys. 228 (2009) 7821–7836.

[28] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (2003) 331.

[29] G.D. Weymouth, M.S. Triantafyllou, Global vorticity shedding for a shrinking cylinder, J. Fluid Mech. 702 (2012) 470–487.

[30] N.A. Adams, S. Hickel, Implicit large-eddy simulation: theory and application, in: B. Eckhardt (Ed.), Advances in Turbulence XII vol. 132, Springer, Berlin, Heidelberg, 2009, pp. 743–750.

[31] J. Park, K. Kwon, H. Choi, Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160, J. Mech. Sci. Technol. 12 (1998) 1200–1205.

[32] R.D. Henderson, Details of the drag curve near the onset of vortex shedding, Phys. Fluids 7 (1995) 2102–2104.

[33] 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.

[34] P. Chiu, R. Lin, T.W. Sheu, A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier–Stokes equations in time-varying complex geometries, J. Comput. Phys. 229 (2010) 4476–4500.

[35] C. Norberg, An experimental investigation of the flow around a circular cylinder: influence of aspect ratio, J. Fluid Mech. 258 (1994) 287–316.

[36] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, J. Fluid Mech. 6 (1959) 547–567.

[37] G.D. Weymouth, Physics and learning based computational models for breaking bow waves based on new boundary immersion approaches (Ph.D. thesis), Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 2008.

[38] M. Pourquie, Accuracy close to the wall of immersed boundary methods, in: 4th European Conference of the International Federation for Medical and Biological Engineering vol. 22, Springer, Berlin, Heidelberg, 2009, pp. 1939–1942.

[39] A. Uranga, P.-O. Persson, M. Drela, J. Peraire, Implicit large eddy simulation of transition to turbulence at low Reynolds numbers using a discontinuous Galerkin method, Internat. J. Numer. Methods Engg. 87 (2011) 232–261.

[40] P. Castonguay, C. Liang, A. Jameson, Simulation of transitional flow over airfoils using the spectral difference method, in: 40th AIAA Fluid Dynamics Conference, Chicago, IL, American Institute of Aeronautics and Astronautics, 2010.

[41] F. Roman, V. Armenio, J. Fröhlich, A simple wall-layer model for large eddy simulation with immersed boundary method, Phys. Fluids 21 (2009) 101701–101701-4.

[42] Z.L. Chen, A. Devesa, M. Meyer, E. Lauer, S. Hickel, C. Stemmer, N.A. Adams, Wall modelling for implicit large eddy simulation of favourable and adverse pressure gradient flows, in: Progress in Wall Turbulence: Understanding and Modeling: Proceedings of the WALLTURB International Workshop held in Lille, France, April 21–23, vol. 14, 2009, p. 337.

[43] F. Capizzano, Turbulent wall model for immersed boundary methods, AIAA J. 49 (2011) 2367–2381.

[44] I.H. Tuncer, M.F. Platzer, Computational study of flapping airfoil aerodynamics, J. Aircr. 37 (2000) 514–520.

[45] K. Isogai, Y. Shinmoto, Y. Watanabe, Effects of dynamic stall on propulsive efficiency and thrust of flapping airfoil, AIAA J. 37 (1999) 1145–1151.

[46] S. Coombs, J.C. Montgomery, The enigmatic lateral line system, in: R.R. Fay, A.N. Popper (Eds.), Comparative Hearing: Fish and Amphibians, Springer Handbook of Auditory Research, Springer, New York, 1999, pp. 319–362.

[47] S.P. Windsor, S.E. Norris, S.M. Cameron, G.D. Mallinson, J.C. Montgomery, The flow fields involved in hydrodynamic imaging by blind mexican cave fish (*Astyanax fasciatus*). Part I: open water and heading towards a wall, J. Exp. Biol. 213 (2010) 3819–3831.

[48] W.-P. Breugem, A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows, J. Comput. Phys. 231 (2012) 4469–4498.

[49] L. Temmerman, M.A. Leschziner, C.P. Mellen, J. Fröhlich, Investigation of wall-function approximations and subgrid-scale models in large eddy simulation of separated flow in a channel with streamwise periodic constrictions, Int. J. Heat Fluid Flow 24 (2003) 157–180.