

# Boundary data immersion method for Cartesian-grid simulations of fluid-body interaction problems

G.D. Weymouth, Dick K.P. Yue\*

Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge 02139, USA

## ARTICLE INFO

### Article history:

Received 18 December 2009

Received in revised form 8 April 2011

Accepted 14 April 2011

Available online 28 April 2011

### Keywords:

Cartesian-grid

Immersed boundary method

Multi-phase flows

Fluid-body interactions

## ABSTRACT

A new robust and accurate Cartesian-grid treatment for the immersion of solid bodies within a fluid with general boundary conditions is described. The new approach, the Boundary Data Immersion Method (BDIM), is derived based on a general integration kernel formulation which allows the field equations of each domain and the interfacial conditions to be combined analytically. The resulting governing equation for the complete domain preserves the behavior of the original system in an efficient Cartesian-grid method, including stable and accurate pressure values on the solid boundary. The kernel formulation allows a detailed analysis of the method, and it is demonstrated that BDIM is consistent, obtains second-order convergence relative to the kernel width, and is robust with respect to the grid and boundary alignment. Formulation for no-slip and free slip boundary conditions are derived and numerical results are obtained for the flow past a cylinder and the impact of blunt bodies through a free surface. The BDIM predictions are compared to analytic, experimental and previous numerical results confirming the properties, efficiency and efficacy of this new boundary treatment for Cartesian grid methods.

© 2011 Elsevier Inc. All rights reserved.

## 1. Introduction

Cartesian-grid methods are an important advancement in the simulation of general fluid flows because they afford the capability of computing flows with ocean engineering applications without the limitations and difficulties associated with non-orthogonal or unstructured grids. Cartesian-grid solvers have the potential to generate solutions to complicated problems orders of magnitude faster than conventional fitted-grid solvers with comparatively simple, accurate and robust numerical methods.

Two primary methods exist in the literature to enforce the effects of solid bodies in Cartesian-grid simulations: sharp-interface methods and Immersed-Boundary methods. The Immersed Boundary (IB) methods were introduced by Peskin in the 1970s for the simulation of heart valves [1,2] and have since been applied extensively in biological fluid dynamics and other fluid–structure interaction problems such as a flapping filament in a cross flow ([3], and see [4] for a detailed review). In the original IB methods, the elastic body equations are discretized on a surface mesh, while the fluid equations are discretized on a Cartesian grid. The two simulations are linked by applying the body reaction force to the fluid and advecting the body in the resulting flow. The localized forces and body velocities are evaluated using a regularized delta function.

Further work on immersed boundaries sought to address modeling of rigid bodies by adding artificially defined body forces to the fluid equations to drive the velocity field near the body to match the boundary value data. Many options for

\* Corresponding author.

E-mail addresses: [weymouth@mit.edu](mailto:weymouth@mit.edu) (G.D. Weymouth), [yue@mit.edu](mailto:yue@mit.edu) (D.K.P. Yue).

the forcing were explored such as setting the force to exponentially dampen the error in the velocity field [5] or using an explicit Integral/Position feedback controller [6]. However, these methods only enforce their boundary conditions approximately in time. Fadlun et al. [7] defined the force such that it directly canceled the error at each time step, which has subsequently been dubbed the direct forcing method. Since then work has been done establishing the proper form of the boundary data [8] and improved kernels for distributing the boundary data to the local grid points [9]. However, the application of a simple forcing function to the original equations of motion does not address the concerns of proper physical behavior of the equations in the kernel region, specifically in regards to the pressure terms [10]. Additionally, such methods are not easily extended to general interfacial conditions, such as jump conditions or Neumann conditions.

Sharp-interface approaches have been explored as an alternative to smooth Immersed Boundary methods. In these methods the interface effects are included in the discrete spatial operators instead of through a forcing function. An advantage of this approach is the ability to include general types of interfacial conditions above and beyond the no-slip condition, such as in the Immersed Interface method of Leveque and Li [11] which included jump conditions in creeping fluid–fluid free-interfaces. Similarly, Gibou et al. [12], Udaykumar et al. [10], Marella et al. [13] take particular care to ensure proper treatment of the pressure interfacial conditions at the immersed interface in their altered discrete system of equations. By doing so they show increased accuracy and speed-up in the solution of the pressure equation. There are a great variety of methods for adjusting the local numerical operators; adopting a local normal-tangential coordinate system [11], including Ghost Nodes in the interfacial region [12,14], modifying the discrete finite volume equations to account for volumes ‘clipped’ by the interface [15] and others. However, by adjusting the discrete operators instead of the analytic equations much of the simplicity and ease of analysis of the direct forcing methods is lost.

The complexity of sharp-interface approaches can lead to a loss of second-order behavior near the interface [14], and removes much of the appeal of Cartesian-grid methods as being well-suited to solving complex engineering systems with simple numerical approaches. Guy and Hartenstine [16] shows that direct forcing methods are in general first-order accurate in the  $L_\infty$  norm and generate spurious pressure fluctuations in the region of the immersed body. However, the correct physical behavior of a system must be preserved, particularly the values of pressure on rigid solid bodies, which are required for estimating fluid forces in engineering applications. It is particularly difficult to predict pressure forces accurately in applications with free-interfaces interacting with immersed bodies, such as slamming, sloshing, and water entry problems in ocean engineering.

A new immersed boundary formulation is introduced to meet these requirements of simplicity and accuracy in simulating immersed bodies in difficult multi-phase flows. By altering the analytic form of the fluid equations the method ensures exact enforcement of the boundary data and maintain physically consistent behavior near the smoothed interface. Weymouth et al. [17] briefly introduces some of these ideas, and expanding upon that work, Section 2.1 of the current paper introduces a formal approach for analytically convolving the governing equations for fluid and solid systems with general interface conditions. This boundary data immersion method thereby creates a meta-equation valid over the full domain including the transition region. Analytic simplifications to these equations presented in Section 2.2 produce a functional more suitable for use in large-scale simulations. An introductory channel flow example in Section 2.3 is used to establish the consistency, convergence properties, and sensitivity of the new approach.

Because the initial formulation is general, it can be applied to a variety of solid/fluid systems. To demonstrate this, the governing meta-equations for general fluid systems with immersed no-slip (Section 3.1) and free-slip (Section 3.2) bodies are derived. The resulting no-slip formulation matches the direct forcing method closely, but includes an additional modification to the pressure term analogous to the discrete operator adjustments of sharp interface method such as [12,10], avoiding the projection issues discussed in [16]. The difference is due to the complete transition from the fluid equations to the solid equations, as opposed to the addition of a forcing term as in the direct forcing method. The free-slip condition is enforced through an analytic term on the left-hand side of the meta-equation, which is otherwise identical to the no-slip formulation. A simple to implement second-order solution algorithm for the general form of the meta-equations is presented in Section 3.3.

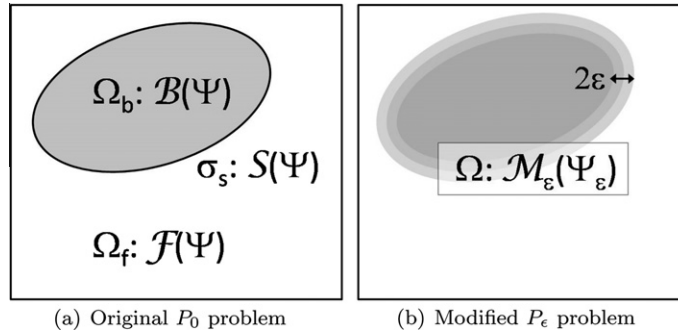
Finally, the new approach is tested using a set of canonical fluids problems: unsteady 2D viscous flow in Section 4.1 and unsteady 3D multi-phase flows in Section 4.2. The solutions are compared to existing numerical and experimental results and grid resolution tests are run to confirm the convergence properties of the boundary data immersion method. These simulations validate the new methodology and verify its ability to obtain efficient solutions to problems of engineering interest easily and robustly.

## 2. The boundary data immersion method

Our eventual goal is the development of governing equations for solid–fluid systems, but for the time being we consider a general two-domain boundary value problem problem  $P_0$  defined as: Given domains  $\Omega_b$  and  $\Omega_f$  and their interface  $\sigma_s$  (see Fig. 1(a)), find the solution  $\Psi_0(\vec{x}, t)$  which satisfies

$$\mathcal{B}(\Psi_0, \vec{x}, t) = 0 \quad \vec{x} \in \Omega_b \quad (1a)$$

$$\mathcal{F}(\Psi_0, \vec{x}, t) = 0 \quad \vec{x} \in \Omega_f \quad (1b)$$



**Fig. 1.** Conceptual sketch of the problem. Figure (a) is the original two-domain problem  $P_0$  with domains  $\Omega_b$  and  $\Omega_f$  and their interface  $\sigma_s$  and the respective governing equations  $\mathcal{B}$ ,  $\mathcal{F}$ , and  $\mathcal{S}$ . Figure (b) is the equivalent single-domain problem  $P_\epsilon$  with governing meta-equation  $\mathcal{M}_\epsilon$  constructed by the boundary data immersion method by convoluting and assembling the equations over a width  $\epsilon$ . The transition between the equations is smooth and located within the small distance  $\epsilon$  of the interface  $\sigma_s$ .

$$\mathcal{S}(\Psi_0, \vec{x}, t) = 0 \quad \vec{x} \in \sigma_s \quad (1c)$$

where  $\mathcal{B}$  and  $\mathcal{F}$  are the governing field equations in domains  $\Omega_b$  and  $\Omega_f$  respectively, and  $\mathcal{S}$  are the interfacial conditions on  $\sigma_s$ . Solid–fluid interaction problems are a special case of Eq. (1), where  $\mathcal{F}$  are the governing equations in the fluid domain  $\Omega_f$ ,  $\mathcal{S}$  the boundary condition on the body boundary  $\sigma_s$ , and  $\Omega_b$  is the body governed by its own elastic or rigid-body equations  $\mathcal{B}$ .

In the present work we seek to reformulate Eq. (1) into a single ‘meta’ governing equation with analytically varying coefficients which describes the complete system. Thus a  $P_0$  system with a possibly complex interface and interfacial conditions is converted into a  $P_\epsilon$  problem with a single meta-equation governing the unified domain. Because the boundary data is integrated into the governing field equation, the  $P_\epsilon$  problem is simpler to implement and more computationally efficient to solve.

Significantly, through concrete numerical examples, we show three important properties of this boundary data immersion method under general smoothness conditions: (i)  $P_\epsilon$  is consistent with  $P_0$  as  $\epsilon \rightarrow 0$ ; (ii)  $P_\epsilon$  has second-order convergence, i.e.,  $\Psi_\epsilon = \Psi_0 + O(\epsilon^2)$ ; and (iii) the solution is insensitive to the relative locations of the numerical grid and the boundary interface.

### 2.1. Boundary immersion through kernel integration

To derive the meta-equation, we must first formally extend the range of Eqs. 1a, 1b, and 1c to the full domain  $\Omega \equiv \Omega_b \cup \Omega_f$ . This is done through the use of a nascent delta integration kernel  $K_\epsilon$ , which has unit zeroth moment and compact support  $\epsilon$ , i.e.,  $\int_\Omega K_\epsilon(\vec{x}, \vec{y}) d\vec{y} \equiv 1$ , and  $K_\epsilon(\vec{x}, \vec{y}) \equiv 0$  for  $|\vec{x} - \vec{y}| > \epsilon$ . To illustrate the properties of  $K_\epsilon$ , we define the zeroth moments of the kernel over the sub-domains (which are unsteady for systems with moving interfaces):

$$\delta_\epsilon^B(\vec{x}, t) = \int_{\Omega_b} K_\epsilon(\vec{x}, \vec{x}_b) d\vec{x}_b \quad (2a)$$

$$\delta_\epsilon^F(\vec{x}, t) = \int_{\Omega_f} K_\epsilon(\vec{x}, \vec{x}_f) d\vec{x}_f \quad (2b)$$

$$\delta_\epsilon^S(\vec{x}, t) = \int_{\sigma_s} K_\epsilon(\vec{x}, \vec{x}_s) d\vec{x}_s. \quad (2c)$$

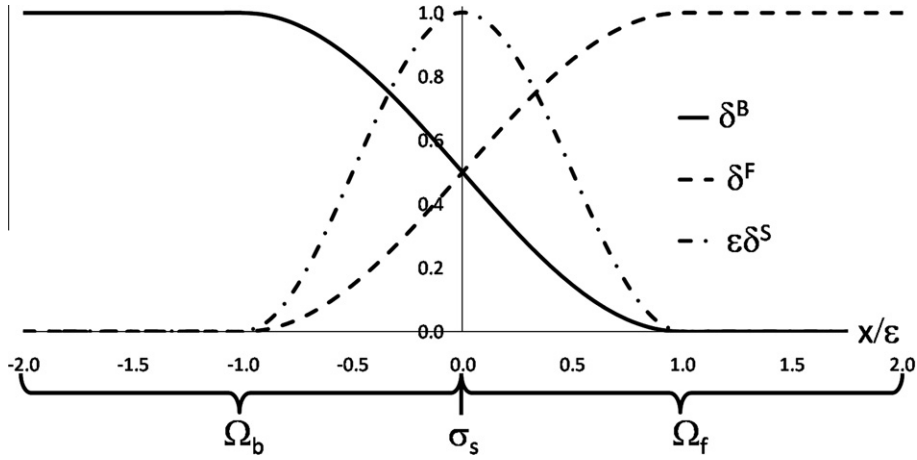
Fig. 2 sketches the behavior of these kernel integrals for a simple 1D domain. Note that: (a) each kernel integral is non-zero only within its own sub-domain plus a distance  $\epsilon$  beyond; (b) the values of all the integrals change smoothly; and (c) while  $\delta_\epsilon^B$  and  $\delta_\epsilon^F$  vary from 1 to 0, the integral over the point-boundary  $\sigma_s$  results in  $\delta_\epsilon^S(x) = K_\epsilon(x, \sigma_s)$  with a peak magnitude of  $1/\epsilon$  at  $x = \sigma_s$ , so that  $\delta_\epsilon^S = O(1/\epsilon)\delta_\epsilon^F = O(1/\epsilon)\delta_\epsilon^B$  in the interface region. These properties allow  $K_\epsilon$  to analytically extend the range of each governing equation and smoothly transition between them.

Convoluting Eq. (1) with  $K_\epsilon$  over their respective domains gives

$$\mathcal{B}_\epsilon(\Psi, \vec{x}, t) \equiv \int_{\Omega_b} \mathcal{B}(\Psi, \vec{x}_b, t) K_\epsilon(\vec{x}, \vec{x}_b) d\vec{x}_b \quad (3a)$$

$$\mathcal{F}_\epsilon(\Psi, \vec{x}, t) \equiv \int_{\Omega_f} \mathcal{F}(\Psi, \vec{x}_f, t) K_\epsilon(\vec{x}, \vec{x}_f) d\vec{x}_f \quad (3b)$$

$$\mathcal{S}_\epsilon(\Psi, \vec{x}, t) \equiv \int_{\sigma_s} \mathcal{S}(\Psi, \vec{x}_f, t) K_\epsilon(\vec{x}, \vec{x}_s) d\vec{x}_s. \quad (3c)$$



**Fig. 2.** Behavior of the kernel integrals  $\delta_\epsilon$  for a 1D case. For points  $x < -\epsilon$  within  $\Omega_b$  we have  $\delta_\epsilon^B = 1$  and  $\delta_\epsilon^S, \delta_\epsilon^F = 0$ . Similarly, for  $x > \epsilon$ , we have  $\delta_\epsilon^F = 1$  and  $\delta_\epsilon^S, \delta_\epsilon^B = 0$ . Within the smoothing region, points less than  $\epsilon$  from  $\sigma_s$ ,  $\delta_\epsilon^F, \delta_\epsilon^B, \epsilon \delta_\epsilon^S = O(1)$ .

As desired, these are defined over the full domain  $\Omega$  while the compact support of  $K_\epsilon$  ensures they are non-trivial only within their respective domains and a small distance  $\epsilon$  beyond. To obtain the solution  $\Psi_\epsilon$  over the full domain we construct the analytic meta-equation  $\mathcal{M}_\epsilon$  by simply assembling Eq. (3) as

$$\mathcal{M}_\epsilon(\Psi_\epsilon) \equiv \mathcal{B}_\epsilon(\Psi_\epsilon) + \mathcal{F}_\epsilon(\Psi_\epsilon) + \mathcal{S}_\epsilon(\Psi_\epsilon) = 0. \quad (4)$$

Note that  $\mathcal{M}_\epsilon$  has the properties

$$\mathcal{M}_\epsilon(\Psi_\epsilon) = \mathcal{B}(\Psi) + O(\epsilon) \quad \vec{x} \in \Omega_b^{-\epsilon} \quad (5a)$$

$$\mathcal{M}_\epsilon(\Psi_\epsilon) = \mathcal{F}(\Psi) + O(\epsilon) \quad \vec{x} \in \Omega_f^{-\epsilon} \quad (5b)$$

$$\epsilon \mathcal{M}_\epsilon(\Psi_\epsilon) = \mathcal{S}(\Psi) + O(\epsilon) \quad \vec{x} \in \sigma_s^{+\epsilon} \quad (5c)$$

where  $\sigma_s^{+\epsilon}$  is the boundary  $\sigma_s$  plus the domain  $O(\epsilon)$  distance around it, while  $\Omega_{bf}^{-\epsilon} \equiv \Omega_{bf} - \sigma_s^{+\epsilon}$  are the respective domains outside the smooth boundary interface. For example, for the simple 1D domain of Fig. 2 with  $\sigma_s$  at  $x=0$ ,  $\Omega_b^{-\epsilon} = \{x < -O(\epsilon)\}$ ,  $\Omega_f^{-\epsilon} = \{x > O(\epsilon)\}$  and  $\sigma_s^{+\epsilon} = \{|x| < O(\epsilon)\}$ . Thus, in the sense of Eq. (5),  $\mathcal{M}_\epsilon$  is  $O(\epsilon)$  close to the governing equation of each region. Intuitively, one might argue that the solution to 4,  $\Psi_\epsilon$ , will be close to  $\Psi_0$  (of the original  $P_0$  problem) for small  $\epsilon$ . The convergence of  $\Psi_\epsilon$  to  $\Psi_0$  is demonstrated by our numerical results, and indeed we find that  $\Psi_\epsilon = \Psi_0 + O(\epsilon^2)$  for the general classes of problems we considered.

## 2.2. Simplified meta-equations using integral approximations

The form of Eq. (3) are general but the linear systems resulting from their discretization are not sparse due to the kernel dependence on all points within an  $\epsilon$  radius. As the primary contribution in the kernel  $K_\epsilon(\vec{x}, \vec{y})$  is from the point  $\vec{x} = \vec{y}$ , the integrals may be simplified considerably,

$$\mathcal{B}_\epsilon(\Psi_\epsilon, \vec{x}, t) \approx \mathcal{B}(\Psi_\epsilon, \vec{x}, t) \int_{\Omega_b} K_\epsilon(\vec{x}, \vec{x}_b) d\vec{x}_b = \mathcal{B}(\Psi_\epsilon, \vec{x}, t) \delta_\epsilon^B(\vec{x}, t) \quad (6a)$$

$$\mathcal{F}_\epsilon(\Psi_\epsilon, \vec{x}, t) \approx \mathcal{F}(\Psi_\epsilon, \vec{x}, t) \int_{\Omega_f} K_\epsilon(\vec{x}, \vec{x}_f) d\vec{x}_f = \mathcal{F}(\Psi_\epsilon, \vec{x}, t) \delta_\epsilon^F(\vec{x}, t) \quad (6b)$$

$$\mathcal{S}_\epsilon(\Psi_\epsilon, \vec{x}, t) \approx \mathcal{S}(\Psi_\epsilon, \vec{x}, t) \int_{\sigma_s} K_\epsilon(\vec{x}, \vec{x}_s) d\vec{x}_s = \mathcal{S}(\Psi_\epsilon, \vec{x}, t) \delta_\epsilon^S(\vec{x}, t) \quad (6c)$$

with 4 now replaced by

$$\mathcal{M}_\epsilon(\Psi_\epsilon) = \delta_\epsilon^B \mathcal{B}(\Psi_\epsilon) + \delta_\epsilon^F \mathcal{F}(\Psi_\epsilon) + \delta_\epsilon^S \mathcal{S}(\Psi_\epsilon) = 0, \quad (7)$$

such that the kernel moments act as interpolating functions between the domain equations.

To further simplify matters, the general characteristics of the kernel integrals  $\delta_\epsilon^B$ ,  $\delta_\epsilon^F$ , and  $\delta_\epsilon^S$  are independent of the specifics of the domain topologies and the form of  $K_\epsilon$ . As shown in Fig. 2, the value of  $\delta_\epsilon$  is primarily dependant on: (i) the distance to the boundary, and (ii) whether the kernel is integrated over a sub-domain or a boundary. Thus for domain  $\Omega_b$  (say), if we define  $d$  as the signed distance from any point  $\vec{x}$  to nearest point on  $\sigma_s$ , chosen negative within  $\Omega_b$ , an appropriate approximation for  $\delta_\epsilon^B$  is

$$\delta_\epsilon^B(d) = \begin{cases} \frac{1}{2} [1 + \sin(\frac{\pi}{2} \frac{d}{\epsilon})] & \text{for } |d| < \epsilon \\ 1 & \text{for } d < -\epsilon \\ 0 & \text{else} \end{cases} \quad (8)$$

For the complementary domain  $\Omega_f$ , we have simply  $\delta_\epsilon^F(d) = \delta_\epsilon^B(-d) = 1 - \delta_\epsilon^B(d)$ . For the interface  $\sigma_s$ , we approximate  $\delta_\epsilon^S$  as

$$\delta_\epsilon^S(d) \equiv \begin{cases} \frac{1}{2\epsilon} [1 + \cos(\pi \frac{d}{\epsilon})] & \text{for } |d| < \epsilon \\ 0 & \text{else} \end{cases} \quad (9)$$

Thus Eq. 7 may be posed using Eqs. 8 and 9 for the analytic coefficients. This approach maintains the  $\epsilon$ -close properties of  $\mathcal{M}_\epsilon$  in Eq. (5) and allows Eq. 7 to be implemented using simple robust and efficient solution methods for PDEs posed on Cartesian grids, as demonstrated in the next section.

### 2.3. Application to the simple Poiseuille flow problem

Before applying the boundary data immersion method to generate meta-equations for the general Navier–Stokes equations, we first examine a simple Poiseuille flow problem to establish the numerical properties of the approach.

Consider the steady 2D pressure driven flow between two stationary parallel plates. The flow is uniform in the  $x$ -direction resulting in a simple boundary value problem for the horizontal velocity  $u(y)$  for  $0 < y < L$ , where  $L$  is the height of the channel. The governing equation and boundary conditions are

$$\mathcal{F}(u) = f_x + \nu \frac{d^2 u}{dy^2} = 0 \quad 0 < y < L \quad (10a)$$

$$u = 0 \quad y = 0, L \quad (10b)$$

where  $f_x$  is the applied horizontal pressure force, and  $\nu$  the kinematic viscosity. The exact solution is simply

$$u_0(y) = \frac{f_x}{2\nu} y(L - y) \quad (11)$$

within the channel.

To test the properties of the boundary data immersion method we first pose this system as a two-domain problem by defining the two semi-infinite ‘wall’ domains outside the fluid ( $y \leq 0, y \geq L$ ) as  $\Omega_b$ . The equation for the velocity in  $\Omega_b$  is simply

$$\mathcal{B}(u) = u = 0, \quad y \leq 0, \quad y \geq L \quad (12)$$

which, combined with a condition of velocity continuity across the interface, gives the ‘no-slip’ boundary conditions of Eq. 10b. Note that continuity conditions on the interface, unlike derivative conditions or interfacial forcing, are implicit if a continuous basis is chosen. Derivative conditions are incorporated in Section 3.2, but for systems with continuity conditions the meta-equation formula can be further simplified to

$$\mathcal{M}_\epsilon(\Psi_\epsilon) = [1 - \delta_\epsilon^B] \mathcal{F}(\Psi_\epsilon) + \delta_\epsilon^B \mathcal{B}(\Psi_\epsilon) = 0 \quad (13)$$

giving the meta-equation for the channel flow as

$$\left\{ \delta_\epsilon^B - [1 - \delta_\epsilon^B] \frac{d^2}{dy^2} \right\} u_\epsilon = [1 - \delta_\epsilon^B] \frac{f_x}{\nu} \quad (14)$$

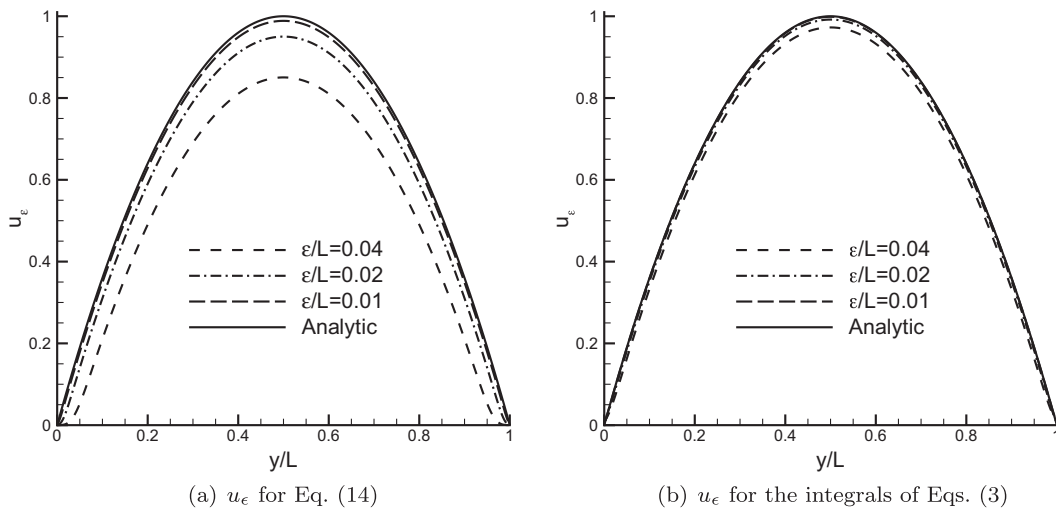
where  $u_\epsilon$  is the solution to this smoothed version of the governing equations. The interpolation function  $\delta_\epsilon^B$  is given by Eq. 8 where the distance function is given by

$$d(y) = \frac{L}{2} - \left| \frac{L}{2} - y \right| \quad (15)$$

for this simple channel geometry.

We demonstrate four important properties of the boundary data immersion method as expressed by the meta-Eq. 14: (1) the system  $P_\epsilon$  is consistent with  $P_0$  when  $\epsilon = 0$ ; (2) when  $\epsilon$  is nonzero and resolved by the grid, the smooth solution is convergent with  $u_\epsilon = u_0 + O(\epsilon^2)$ ; (3) the solution is insensitive to the relative locations of the numerical grid and the interface; and (4) the simplified meta equations of Section 2.2 are of higher accuracy for a given computational cost than the full meta equations of Section 2.1.

First, we ensure the boundary data immersion method solves the correct problem by demonstrating that the  $P_0$  and  $P_\epsilon$  systems are equivalent when  $\epsilon = 0$ . To do so, we first define a numerical grid with  $N$  points  $\mathbf{y}^T = \{y^1, y^2, \dots, y^N\}$ , spaced uniformly  $\Delta y$  apart, with  $y^1 = 0$  and  $y^N = L$ . The solution vector  $\mathbf{u}_\epsilon^T = \{u_\epsilon^1, u_\epsilon^2, \dots, u_\epsilon^N\}$  corresponds to the values of the velocity field at  $u_\epsilon(y^1), \dots, u_\epsilon(y^N)$ . We estimate the second derivative in Eq. 14 using second-order central differences. Setting  $\epsilon = 0$ , we have  $\delta_\epsilon^B = 1$  in the body and  $\delta_\epsilon^B = 0$  in the fluid and the resulting linear algebra system for  $\mathbf{u}_\epsilon$  is



**Fig. 3.** Numerical solution to the Poiseuille flow for  $u_{\max} = fL^2/(8\nu) = 1$  and  $N = L/\Delta y = 400$  using the meta-equation approach.

$$\begin{pmatrix} 1 & & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} u_{\epsilon}^1 \\ u_{\epsilon}^2 \\ \vdots \\ u_{\epsilon}^{N-1} \\ u_{\epsilon}^N \end{pmatrix} = \frac{\Delta y^2 f_x}{\nu} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \quad (16)$$

which is identical to the linear system resulting from the original boundary value problem and the solution  $u_{\epsilon} = u_0$  for all points in the domain. This demonstrates that if the meta-equation is posed on a boundary-fitted grid and the kernel width is set to zero, the standard governing equation and boundary condition system is recovered.

Next, we study the accuracy of the meta-equation for non-zero  $\epsilon$  to determine the convergence of  $u_{\epsilon}$  to  $u_0$ . Using the same uniform grid,  $\mathbf{y}$ , the solution  $u_{\epsilon}$  of Eq. 14 is determined numerically for  $u_{\max} = fL^2/(8\nu) = 1$ . The solution and the error  $E \equiv |u_{\epsilon} - u_0|$  are shown in Fig. 3(a) and Table 1(a) for decreasing values of  $\epsilon$ . The figure and table show that the numerical solution converges to the analytic solution at an approximately second-order rate with decreasing  $\epsilon$  (as evidenced by the column of  $|E|_2$  normalized by  $(\epsilon/L)^2$ ). The second order convergence of the method is further illustrated in the applications section and demonstrates that meta-equation solutions obtain high-order accuracy without using boundary fitted grids or changing the numerical operators near the interface.

To quantify the effect of grid alignment, we shift the computational grid relative to the channel boundary by  $\alpha\Delta y$ ,  $0 < \alpha < 1$ , and tabulate the norms of the maximum error  $E_{\alpha} \equiv \max_{\alpha}(E)$  in Table 1(a). The error is clearly not affected by grid misalignment (the maximum change in  $E$  for any  $\alpha$  is  $O(10^{-6})$  for all the  $\epsilon$  values). This insensitivity to the alignment of the boundary and the underlying grid is a desired property of general robust Cartesian-grid methods.

Finally, we examine the difference between using the simplified integrals of Eq. (6) compared to the full integrals of Eq. (3). As discussed in Section 2.2, using the former can result in substantial computational efficiency gain. For the present 1D problem, for example, the equation matrix resulting from Eq. 16 is tridiagonal while the corresponding equation system using Eq. (3) is  $m$ -diagonal, where  $m = O(\epsilon/\Delta y)$ . For a 3D problem (referencing  $O((\epsilon/\Delta y)^3)$  local points), a conservative esti-

**Table 1**

Error metrics of the solution to the Poiseuille flow problem for  $u_{\max} = fL^2/(8\nu) = 1$  and  $N = L/\Delta y = 400$ . (a) Convergence and sensitivity tests for Eq. 14 (b) Convergence and sensitivity tests for the full integral method of Eq. (3).

$\epsilon/L$	$ E _1$	$ E _2$	$ E _2/(\epsilon/L)^2$	$ E_{\alpha} _1 -  E _1$
(a) Convergence and sensitivity tests for Eq. (14)				
0.04	1.4260E-1	1.4457E-1	90.35	2.820E-6
0.02	4.8827E-2	4.9091E-2	122.7	2.225E-6
0.01	1.1191E-2	1.1218E-2	112.2	6.376E-6
(b) Convergence and sensitivity tests for the full integral method Eqs. (3)				
0.04	2.6529E-2	2.6746E-2	16.72	1.655E-5
0.02	7.9179E-3	7.9489E-3	19.87	1.473E-5
0.01	1.6465E-3	1.6783E-3	16.78	1.190E-5

mate based on our experience is that obtaining the solution using the simplified Eq. (6) is a factor of  $O(10^3)$  more efficient relative to that using the full integrals Eq. (3).

To compare the relative accuracy of the two methods the full integral equations are solved numerically using the same grid and values of  $\epsilon$  as above. The results, shown in Fig. 3(b) and Table 1(b), demonstrate that both methods obtain second-order convergence with  $\epsilon$  and that while the solution using the full integrals has a somewhat smaller error the improvement factor is less than  $O(10)$  and independent of  $\epsilon/L$ . This moderate increase in accuracy is insufficient to overcome the relatively massive costs of solving the full integral equations, establishing the efficacy of using Eq. (6).

We have demonstrated that the meta-equation derived using the boundary data immersion method is consistent with the original system, produces high-accuracy (second-order convergent) solutions insensitive to the location of the background grid, and is cost effective. Having shown these for the simple 1D problem, the remainder of the work addresses the usefulness of the method for general 2D and 3D systems with immersed solid bodies.

### 3. Derivation of meta-equations for general fluid-body systems

Following the framework developed in Section 2, generating meta-equations is straightforward once the governing equation for each sub-domain is defined. In this section we apply this boundary data immersion method to quickly derive governing equations for ocean engineering systems with immersed no-slip or free-slip solid bodies. The resulting equations are similar in form to those used in direct forcing methods, but with significant differences in the pressure forcing terms and the ability to model general interface conditions.

In this work we consider the case where the displacements of the immersed object are prescribed, given by the trivial governing equation

$$\mathcal{B}(\vec{u}) = \vec{u} - \vec{U} = 0 \quad (17)$$

where  $\vec{u}$  is velocity field and  $\vec{U}$  is the prescribed displacement rate. We write the general form of the governing equation for the fluid as

$$\frac{\partial \vec{u}}{\partial t} = \vec{r}(\vec{u}) - \frac{1}{\rho} \vec{\nabla} p \quad (18)$$

where  $\rho$  is the density,  $p$  is the pressure field and  $\vec{r}$  is the aggregate of all the non-pressure terms. In general this may include convective, gravitational, viscous, and other modeling terms without altering the meta-equations below. In order to generate tractable meta-equations, we must be able to combine the fluid and solid velocity field terms directly. One approach is simply to integrate Eq. 18 over the time step  $\Delta t$ , giving the velocity impulse equation

$$\mathcal{F}(\vec{u}, p) = \vec{u} - \vec{u}^0 - \vec{R}_{\Delta t}(\vec{u}) + \partial \vec{P}_{\epsilon \Delta t} = 0 \quad (19)$$

where  $\vec{u}^0 = \vec{u}(t_0)$  and

$$\vec{R}_{\Delta t}(\vec{u}) = \int_{t_0}^{t_0 + \Delta t} \vec{r}(\vec{u}) dt \quad (20a)$$

$$\partial \vec{P}_{\Delta t} = \int_{t_0}^{t_0 + \Delta t} \frac{1}{\rho} \vec{\nabla} p dt. \quad (20b)$$

These integral can be evaluated with general implicit or explicit quadratures, and a second-order explicit method appropriate for time dependent simulations is given in the following section.

#### 3.1. Immersion of a no-slip body into a general flow

If we assume the velocity is continuous across the interface (the ‘no-slip’ condition) then the interfacial equations  $S$  are not required and we simply substitute Eqs. 17 and 19 into Eq. 13 giving

$$\mathcal{M}_{\epsilon}(\vec{u}_{\epsilon}, p_{\epsilon}) = \vec{u}_{\epsilon} - \delta_{\epsilon}^B \vec{R} - [1 - \delta_{\epsilon}^B] \left( \vec{u}_{\epsilon}^0 + \vec{R}_{\Delta t}(\vec{u}) - \partial \vec{P}_{\epsilon \Delta t} \right) = 0. \quad (21)$$

A divergence-free constraint on the velocity field

$$\vec{\nabla} \cdot \vec{u}_{\epsilon} = 0 \quad (22)$$

applied to this meta-equation gives

$$\vec{\nabla} \cdot \left( [1 - \delta_{\epsilon}^B] \partial \vec{P}_{\epsilon \Delta t} \right) = \vec{\nabla} \cdot \left( \delta_{\epsilon}^B \vec{U} + [1 - \delta_{\epsilon}^B] \left( \vec{u}_{\epsilon}^0 + \vec{R}_{\Delta t}(\vec{u}) \right) \right) \quad (23)$$

which is inverted for the pressure  $p_{\epsilon}$ .



To aid in comparisons to previous methods we apply explicit Euler integration to Eq. (20) and reorder the resulting terms to give

$$\vec{\nabla} \cdot \left( \frac{1 - \delta_\epsilon^B}{\rho^0} \vec{\nabla} p_\epsilon^0 \right) = \vec{\nabla} \cdot \left[ \vec{r}_\epsilon^0 + \frac{\vec{u}_\epsilon^0}{\Delta t} \right] \quad (24a)$$

$$\frac{\vec{u}_\epsilon' - \vec{u}_\epsilon^0}{\Delta t} = \vec{r}_\epsilon^0 - \frac{1 - \delta_\epsilon^B}{\rho^0} \vec{\nabla} p_\epsilon^0 \quad (24b)$$

where  $\vec{u}_\epsilon'$  is the first-order estimate of the velocity field and  $\vec{r}_\epsilon$  is the adjusted RHS given by

$$\vec{r}_\epsilon = \delta_\epsilon^B \frac{\vec{U} - \vec{u}_\epsilon^0}{\Delta t} + [1 - \delta_\epsilon^B] \vec{r}. \quad (25)$$

To achieve second-order accuracy in time, we adopt Huen's method which follows the predictor step of Eq. (24) with an explicit trapezoidal corrector step given by

$$\vec{\nabla} \cdot \left( \frac{1 - \delta_\epsilon^B}{\rho'} \vec{\nabla} p_\epsilon' \right) = \vec{\nabla} \cdot \left[ \vec{r}_\epsilon' + \frac{\vec{u}_\epsilon' + \vec{u}_\epsilon^0}{\Delta t} \right] \quad (26a)$$

$$\frac{\vec{u}_\epsilon - \vec{u}_\epsilon^0}{\Delta t} = \frac{1}{2} \left[ \frac{\vec{u}_\epsilon' - \vec{u}_\epsilon^0}{\Delta t} + \vec{r}_\epsilon' - \frac{1 - \delta_\epsilon^B}{\rho'} \vec{\nabla} p_\epsilon' \right] \quad (26b)$$

which is solved for the velocity  $\vec{u}_\epsilon$  at the new time  $t = t_0 + \Delta t$ .

Eq. (24), Eq. (26) are in the family of direct forcing methods except for the unique treatment of the pressure term. The  $1 - \delta_\epsilon^B$  coefficient mollifying the pressure force in the transition region is directly analogous to the discrete operators appearing in sharp interface treatments of the Poisson equation, and acts to implicitly enforce the Neumann boundary condition on the pressure field,

$$\frac{\partial p}{\partial n} = \rho \left( r_n - \frac{\partial U_n}{\partial t} \right) \quad (27)$$

where  $\hat{n}$  is the interface normal. This modified pressure treatment is a direct outcome of convolving the solid and fluid governing equations of motion as opposed to adding a forcing term to the fluid as in Immersed Boundary methods. As shown in Section 4 this new formulation enables stable and accurate pressure force predictions even in water entry problems which have rapid changes in forcing.

### 3.2. Extension to the immersion of a free-slip body

The 'no-slip' boundary condition of Section 3.1 is sometimes not the best choice for large-scale simulations. For example, there is a length scale disparity of up to seven orders of magnitude between a near-wall viscous flow and the gravity waves generated by a (full-scale) ship [18]. In many such applications, modeling the body with a slip-condition of some kind is physically realistic and much more computationally practical. The general method in which the boundary data immersion method was formulated allows us to easily incorporate such modeling adjustments.

The normal component of velocity should still be continuous, giving the no-penetration condition

$$S_n(\vec{u}) = u_n(\vec{x}_b) - U(\vec{x}_b) = 0 \quad (28)$$

which, as above, is enforced implicitly through the use of a continuous basis. However, the tangential velocity may be governed by any user-defined slip model, typically in CFD, a zero Neumann condition on the tangential velocity components. In the context of immersed surfaces it is important to emphasize that the zero Neumann condition does not mean the normal derivatives are zero across the interface, rather that they approach zero *from either side*, allowing the field to be discontinuous across  $\sigma_s$ . With that technicality understood, the tangential surface equations are written as

$$S_\sigma(\vec{u}) = \frac{\partial u_\sigma}{\partial n} = 0 \quad (29a)$$

$$S_\tau(\vec{u}) = \frac{\partial u_\tau}{\partial n} = 0 \quad (29b)$$

where  $\hat{\sigma}$  and  $\hat{\tau}$  are the local tangential vectors. Eqs. 17, 19 and Eq. (29) are substituted into Eq. 7 to give

$$\vec{u}_\epsilon + \vec{\mathcal{L}}(\vec{u}_\epsilon) = \delta_\epsilon^B \vec{U} + [1 - \delta_\epsilon^B] \left( \vec{u}_\epsilon^0 + \vec{R}_{\Delta t}(\vec{u}) - \partial \vec{P}_{\epsilon \Delta t} \right) \quad (30)$$



where  $\mathcal{L}$  is the left hand side operator, defined as

$$\vec{\mathcal{L}}(\vec{u}) \equiv \delta_\epsilon^S \frac{\partial \vec{u}_\tau}{\partial n} \hat{\tau} + \delta_\epsilon^S \frac{\partial \vec{u}_\sigma}{\partial n} \hat{\sigma}, \quad (31)$$

for the Neumann condition. This LHS term is the only functional difference between Eqs. 21 and 30. A solver with a flag to turn this term on or off, such as discussed in Section 3.3, can easily model slip and/or no-slip immersed bodies. Another feature of interest in this equation is that that even though the transition between the solid and fluid governing equations takes place gradually over the width  $\epsilon$ , Eq. 30 allows the solution  $\vec{u}_\epsilon$  to be discontinuous across the interface. This enables a smoothed immersed boundary method to model discontinuous slip conditions accurately, demonstrated in Section 4.

### 3.3. Solution method

A basic algorithm for solving the meta-equations in Section 3.1, Section 3.2 is given in Algorithm 1. These equations are posed on a staggered mesh and central differences are used for all spacial derivatives except in the convective term in  $\vec{r}$  which uses a flux limited QUICK scheme for stability.

---

#### Algorithm 1. Psuedocode solution algorithm

---

```

1: Initialize:  $\vec{u}_\epsilon^0, \rho^0$ 
2: for number-time-steps do
3:   compute:  $d, \vec{U}, \delta_\epsilon$  [Eqs. 8,9]
4:   if free-slip then
5:      $\vec{U} = U_n \hat{n}$ 
6:   end if
7:   for  $i = 1, 2$  do
8:     compute:  $\vec{r}$ 
9:     compute:  $\vec{r}_\epsilon$  [Eq. 25],  $\frac{1-\delta_\epsilon}{\rho}$ 
10:    if  $i == 1$ 
11:      compute:  $p_\epsilon, \vec{u}_\epsilon$  [Eq. (24)]
12:    else
13:      compute:  $p_\epsilon, \vec{u}_\epsilon$  [Eq. (26)]
14:    end if
15:    if free-slip then
16:      compute:  $\vec{u}_\epsilon + \mathcal{L}(\vec{u}_\epsilon) \leftarrow \vec{u}_\epsilon$ 
17:    end if
18:    compute:  $\rho$ 
19:  end for
20:   $\vec{u}_\epsilon^0 \leftarrow \vec{u}_\epsilon, \rho^0 \leftarrow \rho$ 
21: end for

```

---

Each time step of the algorithm above requires up to four inversions; two for the pressure equation (step 11,13) and two for the  $\mathcal{L}$  velocity operator (step 16). Inversion of  $\mathcal{L}$  is straightforward because the  $\mathcal{L}$  operator is banded lower-diagonal when the points are ordered by their distance from the interface. Therefore the velocity “inversion” is accomplished by back substitution and the computational cost scales linearly with the number of points in the smoothing region. A multi-grid method with diagonally preconditioned conjugate-gradient “roughing” is used to invert the pressure equation. This inversion scheme is chosen for its excellent convergence properties and scalability on HPC platforms.

Algorithm 1 shows that altering an existing flow solver to compute solid–fluid interactions with BDIM is extremely straightforward.<sup>1</sup> The only novel computations in the predictor/corrector loop are step (9) to adjust the RHS and scale the fluid density, and step (16) to back-solve the  $\mathcal{L}$  operator for the free-slip conditions. In timing these routines we have found they make up less than 2% of the simulation cost, and speeds of 10 iterations per second for 100 M point simulations have been achieved on modern HPC facilities.

The sole difficulty in computing flows with immersed solids becomes the need to establish the interface distance function  $d$  in step (3) of the algorithm above. There is an extensive literature on obtaining distance functions, but those details are not fundamental to the boundary data immersion method. For demonstration of the solid/fluid solver, we choose canonical circular geometries which generate rich physical systems with a trivial distance function.

<sup>1</sup> In our experience with multiple 2D and 3D solvers, the alteration typically takes less than a day.

## 4. 2D and 3D applications

This section presents applications of the boundary data immersion method (BDIM) to canonical 2D and 3D flows with immersed circular solid bodies. First, the viscous flow about a two-dimensional circular cylinder is modeled and the wake structures and forces are compared to experimental and numerical results. Second, the boundary data immersion method is used with the Volume-of-Fluid method to simulate the 3D water entry of both a sphere and a disk and the predicted forces and free surface profiles are validated against experimental observations and previous simulation methods.

### 4.1. Viscous single-phase 2D flow

We apply the boundary data immersion method to the two-dimensional flow around a stationary circular cylinder. This example demonstrates the properties of BDIM for the viscous Navier–Stokes equation for a single-phase flow using the standard no-slip boundary conditions.

Forces are calculated using the kernel-based method detailed in Weymouth [19]. Briefly, evaluating the pressure force  $\vec{F}_p$  on the body by numerical quadrature of a surface integral, as done in fitted-grid methods, would first require creation of an otherwise unnecessary surface mesh and establishing the pressure on those new grid points. A more direct and computationally efficient method of determining  $F_p$  is to integrate over the Cartesian-grid using the distance function  $d$  and the pressure BC Eq. 27 to implicitly determine  $p$  on  $\sigma_s$  and perform the surface integral. Weymouth [19] derives an appropriate one-sided Derivative Informed Kernel (DIK) integration for  $\vec{F}_p$  of the form

$$\vec{F}_p = \int_{\Omega} \left[ p - d\hat{n} \cdot \left( \vec{r} - \frac{\vec{U} - \vec{u}_0}{\Delta t} \right) \right] \hat{n} \delta_{\epsilon}^+ d\vec{x} \quad (32)$$

where  $\delta_{\epsilon}^+$  is a kernel designed to sample in the fluid near the surface. For this work we used  $\delta_{\epsilon}^+(d) \equiv \delta_{\epsilon}^S(d - \epsilon)/[\epsilon(d/R + 1)]$  where  $R$  is the radius of curvature. Again, the advantage is that evaluation of Eq. 32 gives the normal pressure force on the body in one step without a surface grid.

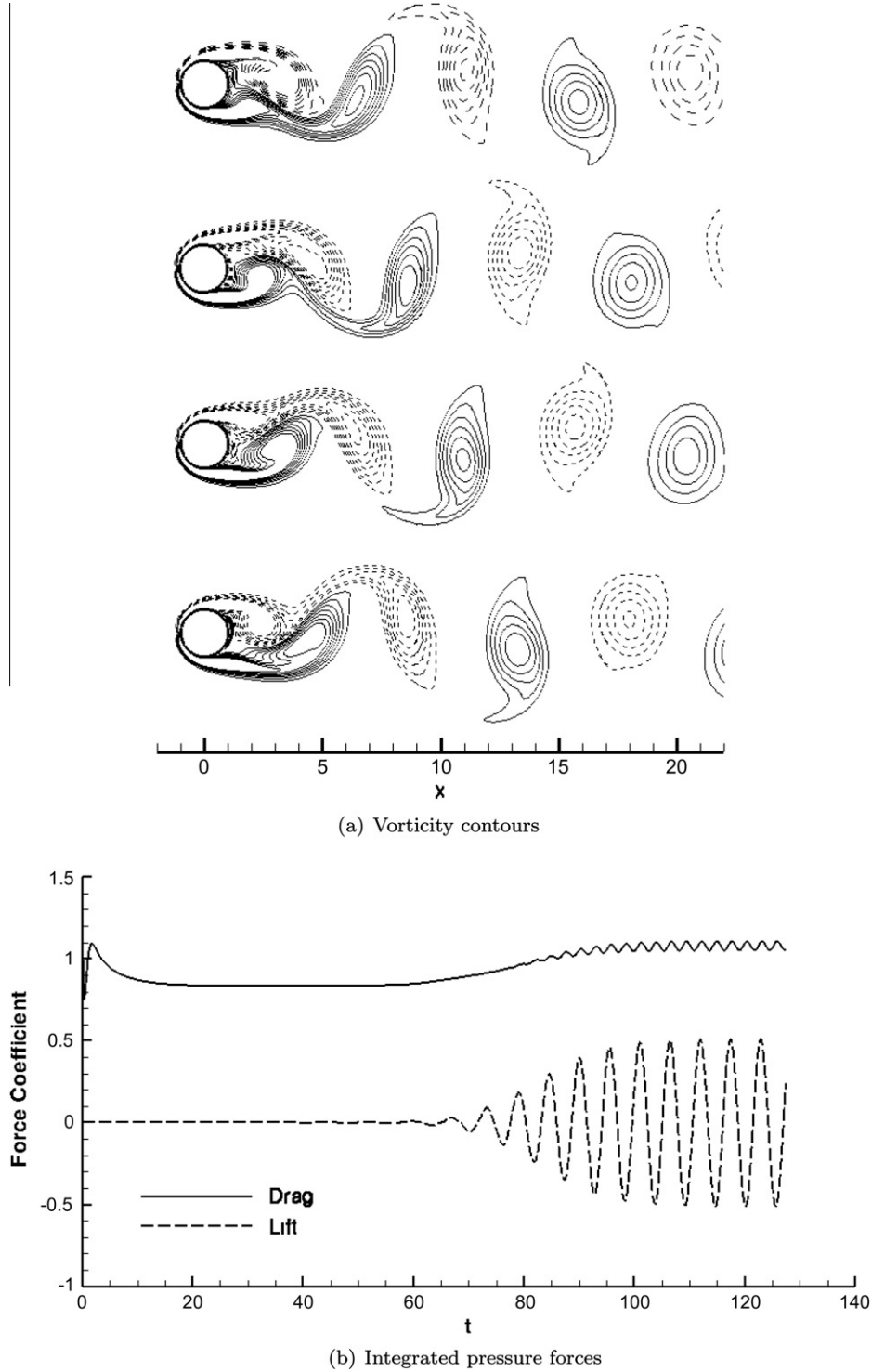
Using BDIM, a two-dimensional flow around a stationary circular cylinder is simulated in a 16X12 diameter domain, constant inflow  $\vec{u} = \vec{U}$  on the inlet, free-slip conditions on the upper and lower boundaries, and a zero gradient exit condition. All of these boundaries, as well as the stationary cylinder boundary, are immersed using the formulation above. The grid spacing is set to 40 points per diameter near the cylinder with a 1% geometric expansion ratio for the grid spacing in the far-field. Simulations are run for a set of smoothing widths  $\epsilon/D = 0.125, 0.0883, 0.0625 = 2/16, \sqrt{2}/16, 1/16$ , to further establish the convergence properties of the method.

The simulation is run at a Reynolds number of  $Re = 150$  which is characterized by an unsteady but laminar and two-dimensional wake. This is done to avoid confusing the analysis of the boundary data immersion method with the additional modeling required to simulate higher Reynolds number turbulent flows. Fig. 4(a) shows the computed vorticity contours in the wake of the cylinder at this Reynolds number. The wake displays the characteristic vortex shedding pattern and four snapshots are displayed within the shedding cycle. Fig. 4(b) shows the time history of the integrated pressure force coefficients on the cylinder (computed using the DIK method and Eq. 32 with the finest value of  $\epsilon/D$ ). The mean pressure drag coefficient is found to be  $C_{pd} = 1.07$ , which compares well to the value of  $C_{pd} = 1.05$  from the spectral element simulations of Henderson [25]. The Strouhal number  $St = D/UT$  and total lift and drag coefficients obtained using BDIM with different values of  $\epsilon/D$ , are compared to the boundary-fitted grid simulations of Liu et al. [20], the immersed boundary simulations of Silva et al. [21], and the experimental measurements of Williamson [22] in Table 2. The shear velocity near the wall is unlikely to remain physical due to the imposition of the no-slip condition in the smoothing region, therefore we use the friction drag coefficient of  $C_{fd} = 0.28$  from Henderson [25] to compare to the total drag values in this table. As in Section 2.3 the current results demonstrate second order convergence in  $\epsilon$ . To further validate the method,  $Re = 100, 200$  were simulated at  $\epsilon/D = 1/16$  and the results are compared to other Cartesian-grid methods in Table 3. The present study compares well with the documented results; validating the BDIM's ability to simulate bluff body viscous flows accurately, as well as validating the DIK method for implicit pressure integration.

### 4.2. Three-dimensional water-entry applications

To further illustrate the usefulness of the boundary data immersion method, we apply BDIM to a set of 3D water-entry problems; the impact force on a sphere in water entry, the cavity formed by a circular disk entering water, and finally an investigation on the effect of surface boundary conditions on the cavity formed by a sphere. These problems offer realistic tests of the robustness and accuracy of the BDIM meta-equation approach, are of practical importance (e.g. Truscott and Techet [26], Yan et al. [27]), and involve special challenges for simulation modeling: 3D moving bodies, jumps in density and viscosity across the free-interface, impact forces, and separation dynamics.

The density and viscosity are functions of position and time in these two-phase flows. They are determined from the color function  $c(\vec{x}, t)$  using



**Fig. 4.** Computed vorticity contours and pressure forces for a stationary circular cylinder at  $Re = 150$  with  $\epsilon/D = 0.0625$ . Figure (a) shows contour of vorticity in the wake at the quarter-periods of the shedding cycle. Figure (b) shows initial time history of the integrated pressure lift and drag force on the cylinder normalized by  $\rho D U^2/2$ .

$$\rho(\vec{x}, t) = c(\vec{x}, t)\rho_{\text{water}} + [1 - c(\vec{x}, t)]\rho_{\text{air}} \quad (33a)$$

$$\mu(\vec{x}, t) = c(\vec{x}, t)\mu_{\text{water}} + [1 - c(\vec{x}, t)]\mu_{\text{air}} \quad (33b)$$

**Table 2**  
Simulated and experimental measurements of the shedding frequency and the mean and fluctuating values of the drag and lift force coefficient on a circular cylinder at  $Re = 150$  compared to BDIM of a range of  $\epsilon/D$  values.

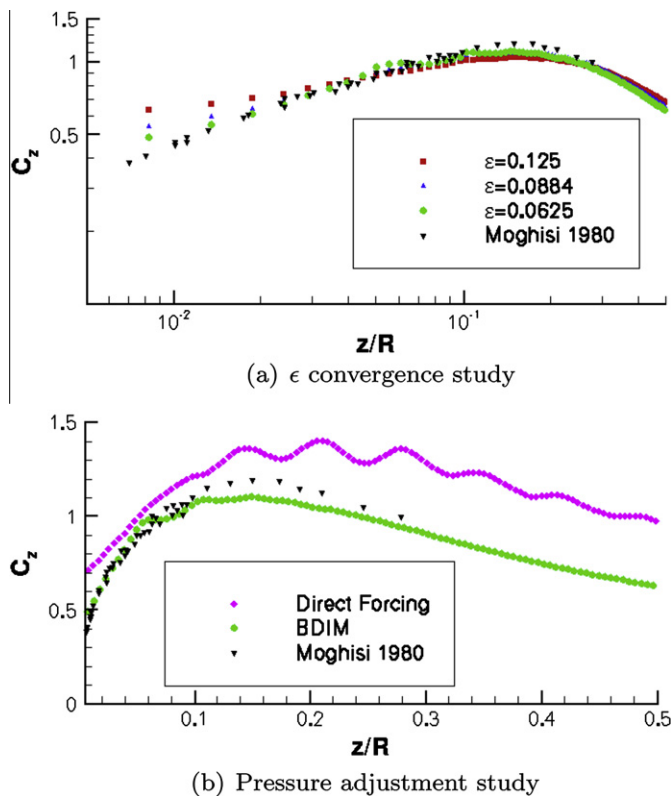
$\epsilon/D$	BDIM			Liu et al. [20]	Silva et al. [21]	Williamson [22]
	0.125	0.0883	0.0625			
$C_D$	$0.140 \pm 0.04$	$1.37 \pm 0.03$	$1.35 \pm 0.03$	$1.33 \pm 0.03$	1.37	–
$C_L$	$0.0 \pm 0.59$	$0.0 \pm 0.56$	$0.0 \pm 0.54$	$0.0 \pm 0.53$	–	–
$St$	0.181	0.183	0.183	0.182	0.180	0.183

**Table 3**  
Simulated measurements of the shedding frequency and the mean drag on a circular cylinder at  $Re = 100, 200$  compared to BDIM with  $\epsilon/D = 1/16$ .

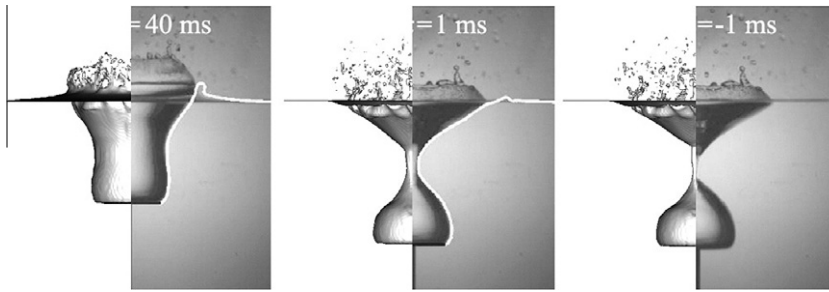
Authors	Re = 100			Re = 200		
	$C_D$	$C_L$	$St$	$C_D$	$C_L$	$St$
Liu et al. [20]	$1.35 \pm 0.012$	$\pm 0.339$	0.164	$1.31 \pm 0.049$	$\pm 0.69$	0.192
Choi et al. [23]	$1.34 \pm 0.011$	$\pm 0.315$	0.164	$1.36 \pm 0.048$	$\pm 0.64$	0.191
Chiu et al. [24]	$1.35 \pm 0.012$	$\pm 0.303$	0.167	$1.37 \pm 0.051$	$\pm 0.71$	0.198
Present	$1.35 \pm 0.013$	$\pm 0.330$	0.167	$1.34 \pm 0.050$	$\pm 0.69$	0.195

for the viscosity. The conservative second-order VOF method of Weymouth and Yue [28] is used to transport the color-function.

First, BDIM is used to predict the pressure forces experienced by a sphere moving with constant downward velocity  $U$  as it impacts the air/water interface. A  $3 \times 3 \times 6$  diameter computational domain is used with free-slip conditions on the far-field



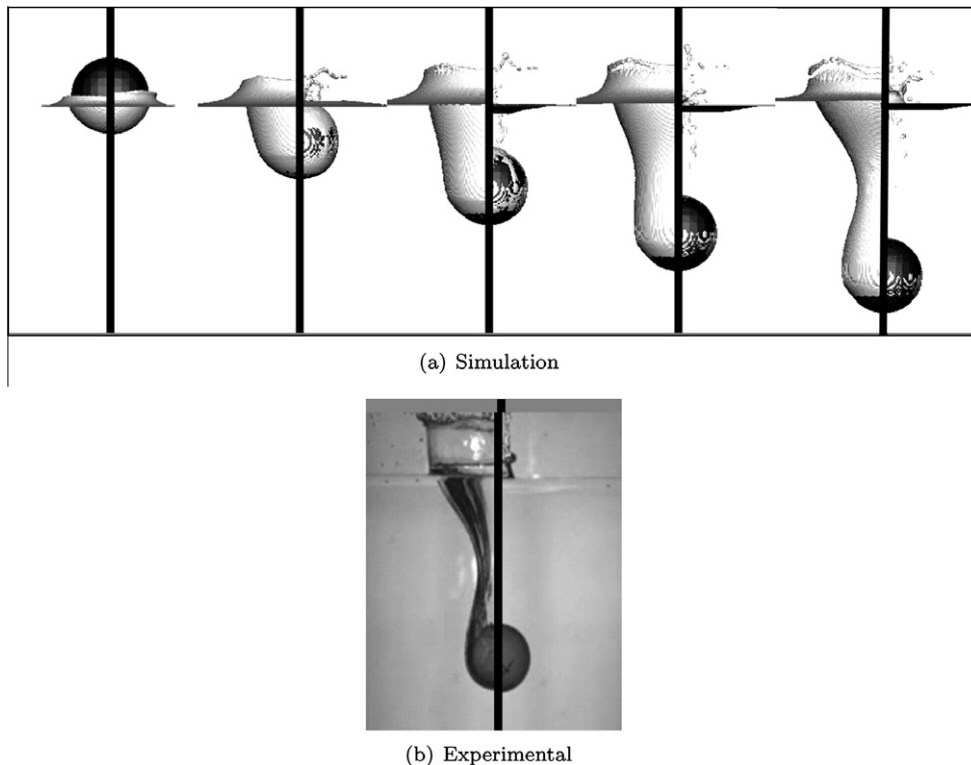
**Fig. 5.** Pressure force coefficients  $C_z = F_z(\frac{1}{2}\rho AU^2)$  for the vertical impact of a sphere with constant speed on an air/water interface. Figure (a) shows the BDIM result using progressively finer values of  $\epsilon$  compared to experimental measurements [29]. As in that work the results are shown on a log scale to examine the early impact history. The Froude number of the simulation is  $Fr = U/\sqrt{gD} = 3$ . The experimental results are for  $Fr = 1 \sim 3$ . The simulations show monotonic convergence with  $\epsilon$  towards quantitative agreement with the experimental measurements. Figure (b) evaluates the effect of removing the BDIM pressure adjustment, as in the direct-forcing treatment. We see direct-forcing treatment results in spurious high-frequency fluctuations in the pressure force and over prediction for the duration of the impact event.



**Fig. 6.** Snapshots of the BDIM simulated impact of a disk on the air/water interface (left) compared to experimental results [30] (right) and the potential flow results found therein (white line in the first two snapshots on the right). Both the current result and the experimental result show splash-up, crowning, a large well-formed cavity, and pinch-off.

boundaries, no-slip conditions on the immersed sphere, and symmetry conditions on the  $x = 0$  and  $y = 0$  planes. A uniform grid is used with 80 points per diameter and the vertical pressure force  $F_z$  is integrated implicitly using the DIK method, as in the previous section. Fig. 5(a) shows BDIM results for progressively refined  $\epsilon/D$  values, compared to the experimental measurements of Moghisi and Squire [29]. As in that work, the results are plotted on a log scale to better examine the impact history near the time of impact where the conditions for the sphere jump from the low density of air to the high density of water. This nonlinear jump in conditions is yet another test of the robustness of BDIM relative to earlier test cases which experience transitions gradually over the smoothing width. All three values of  $\epsilon$  compare well to the experimental results and the figure shows that refinement of the smoothing width results in monotonic convergence to the experimental measurements.

Next the importance of the BDIM treatment of the pressure is investigated in this challenging nonlinear 3D test. As discussed in Section 3.1, the meta equation for an immersed body with prescribed no-slip boundary conditions diverges from the direct-forcing method of Fadlun only in the treatment of the pressure. To directly evaluate the impact of this modifica-



**Fig. 7.** Snapshots of two sphere water-entry simulations illustrating the effect of boundary condition on cavity formation at five penetration depths;  $z/R = 0.7, 1.9, 3.1, 4.3, 5.5$ . In each image the no-slip simulation is shown on the left and the free-slip on the right serving as simple models of hydrophobic and hydrophilic surface coatings, respectively. For comparison, figure (b) shows experimental images of a test case at a comparable Froude number for hydrophobic and hydrophilic coatings [26].

tion on the simulation, the test case was run again without the pressure adjustment in the Poisson equation to replicate the direct-forcing methodology. Fig. 5(b) shows that the result of this omission is a non-physical fluctuation in the predicted force as well as an increase in the predictions of around 20% for the peak response and all values following. Additionally, the method was much more taxing on the pressure solver, requiring twice the number of iterations (and therefore twice the time) to reduce the residual to the same levels as in the BDIM test. The poor predictions and conditioning indicate that these air/water interface impacts can excite numerical instabilities using the standard direct-forcing treatment of the pressure equation.

In the second 3D test case the cavity formed behind a disk at low Froude numbers is studied and compared to the laboratory measurements of Bergmann et al. [30]. In that work a large disk is pulled down through the free-interface with constant speed at low Froude number ( $Fr = 1.8$ ) to investigate the large cavity and pinch off that results. We simulate the same system (without the shaft used to pull the disk) using a  $3 \times 3 \times 12$  domain and uniform grid with 40 points per diameter at the same Froude number. While this background grid is not sufficiently fine to resolve the thinnest section of the splash crown nor the fine details of pinch-off (which are not symmetric) the results, shown in Fig. 6, are still in remarkable agreement with the experimental results. Unlike the potential flow results also included in that work, the current method is able to model the splash-up and continue simulating after pinch-off.

Finally, the ability of BDIM to include general solid–fluid interfacial conditions is used to study the effect of the boundary condition the cavity formed by an impacting sphere. Unlike in the case of a disk, the separation point behind an interface impacting sphere is not fixed and depends on entry speed and surface-coatings among other factors. Of particular interest is the effect of a hydrophobic versus hydrophilic surface/coating of the sphere on the resulting cavity formation [26]. Precise modeling of such surface effects is difficult, but an initial investigation can be made using BDIM and the slip model of Eq. (29). Using the same numerical grid as in the disk cavity investigation above, two sets of low Froude number ( $Fr = 2.0$ ) simulations are run; one with the no-slip boundary condition and one with the Neumann ‘free-slip’ condition. The resulting dynamics for the two cases are shown side-by-side in Fig. 7 with the left side of each snap-shot showing the no-slip case and the right showing the free-slip case at the same instant in time. Despite the lack of dynamic response in the sphere and the primitive (even trivial) surface coating model, the results display good qualitative agreement with the experimental work of Truscott and Tchet [26] at a comparable Froude number ( $Fr = 1.75$ ). Further research along these lines is in progress.

## 5. Conclusion

This paper contributes to the state of the art in Cartesian-grid methods through the development of the Boundary Data Immersion Method (BDIM), a new robust and accurate treatment of general boundary conditions in solid/fluid interaction problems. BDIM establishes analytic meta-equations by posing the original multi-domain problem in a single domain with immersed  $\epsilon$  smoothed interface.

A number of canonical applications are presented: Poiseuille flow, 2D viscous flow past a circle, and 3D water entry problems which confirmed the efficacy of this new boundary treatment for Cartesian-grid methods. Indeed, it is shown that new formulation can generate robust and accurate simulation results using standard second-order numerical methods and simple interpolation kernels.

By using a finite smoothing width  $\epsilon$ , BDIM avoids the implementation complexities of sharp-interface methods, but unlike typical direct forcing methods, the new method is demonstrated to maintain the system dynamics – consistency, second-order convergence with  $\epsilon$  and negligible modeling error in high-resolution simulations. This hold true even in the demanding cases with bodies interacting with multi-phase flows such as slamming.

Further applications of BDIM for solid/fluid systems (including flows around oscillating bodies, non-trivial and deformable solid geometries, as well as DNS/LES turbulent flow simulations) are given in [18,19,31]. Application of the boundary data immersion method in higher order solvers is under active research. Indeed, the method described is general in its approach and should be useful in the broader context involving complex interfaces/boundaries in partial differential equations on numerical grids and discretizations of any type.

This work was supported by the Office of Naval Research. The computational resources for this work were provided through a Challenge Project grant from the DoD High Performance Computing Modernization Office.

## References

- [1] C.S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (2) (1972) 252–271.
- [2] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (3) (1977) 220–252.
- [3] L. Zhu, C.S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, *J. Comput. Phys.* 179 (2) (2002) 452–468.
- [4] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [5] D.G. Dommermuth, G. Innis, T. Luth, E. Novikov, E. Schlageter, J. Talcott, Numerical simulation of bow waves, in: 22nd Symposium on Naval Hydrodynamics, 1998.
- [6] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* 105 (2) (1993) 354–366. ISSN 0021-9991.
- [7] 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 (1) (2000) 35–60. ISSN 0021-9991.

- [8] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2) (2005) 448–476. ISSN 0021-9991.
- [9] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, *J. Comput. Phys.* 228 (18) (2009) 6617–6628. ISSN 0021-9991.
- [10] 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 (1) (2001) 345–380. ISSN 0021-9991.
- [11] R.J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* (1994).
- [12] 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 (1) (2002) 205–227. ISSN 0021-9991.
- [13] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.* 210 (1) (2005) 1–31. ISSN 0021-9991.
- [14] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2) (2003) 593–623. ISSN 0021-9991.
- [15] D.G. Dommermuth, T.T. O'Shea, D.C. Wyatt, M. Sussman, G.D. Weymouth, D.K. Yue, P. Adams, R. Hand, The numerical simulation of ship waves using Cartesian-grid and volume-of-fluid methods, in: 26th Symposium on Naval Hydrodynamics, 2006.
- [16] R.D. Guy, D.A. Hartenstine, On the accuracy of direct forcing immersed boundary methods with projection methods, *J. Comput. Phys.* 229 (7) (2010) 2479–2496. ISSN 0021-9991.
- [17] G.D. Weymouth, D.G. Dommermuth, K. Hendrickson, D.K.-P. Yue, Advancements in Cartesian-grid methods for computational ship hydrodynamics, in: 26th Symposium on Naval Hydrodynamics, 2006.
- [18] G.D. Weymouth, K. Hendrickson, T. O'Shea, D. Dommermuth, D.K.P. Yue, P. Adams, R. Hand, Modeling breaking ship waves for design and analysis of naval vessels, in: HPCMP Users Group Conference, 2006.
- [19] 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, 2008.
- [20] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* 139 (1998) 35–57.
- [21] A.L.E. Silva, A. Silveira-Neto, J. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, *J. Comput. Phys.* 189 (2003) 351–370.
- [22] C.H.K. Williamson, Vortex dynamics in the cylinder wake, *Ann. Rev. Fluid Mech.* 28 (1996) 477–539.
- [23] J.-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, *J. Comput. Phys.* 224 (2) (2007) 757–784.
- [24] 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 (12) (2010) 4476–4500.
- [25] R.D. Henderson, Details of the drag curve near the onset of vortex shedding, *Phys. Fluids* 7 (9) (1995) 2102–2104.
- [26] T.T. Truscott, A.H. Techet, A spin on cavity formation during water entry of hydrophobic and hydrophilic spheres, *Phys. Fluids* (2010).
- [27] H. Yan, Y. Liu, J. Kominiarczuk, D.K.-P. Yue, Cavity dynamics in water entry at low Froude numbers, *J. Fluid. Mech.* 641 (2009).
- [28] G. Weymouth, D.K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids, *J. Comput. Phys.* 229 (8) (2010) 2853–2865. ISSN 0021-9991.
- [29] M. Moghisi, P.T. Squire, An experimental investigation of the initial force of impact on a sphere striking a liquid surface, *J. Fluid Mech.* 108 (1981) 133–146.
- [30] R. Bergmann, D. van der Meer, M. Stijnman, M. Sandtke, A. Prosperetti, D. Lohse, Giant bubble pinch off, *Phys. Rev. Lett.* 96 (2006) 54505.
- [31] K. Hendrickson, D.K.-P. Yue, Models for simulating breaking wave in computational ship hydrodynamics, in: 27th Symposium on Naval Hydrodynamics, 2008.