Boundary Data Immersion Method for Cartesian-Grid Simulations of Fluid-Body Interaction Problems

G.D. Weymouth[a], Dick K.P. Yue[a]

[a]*Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139, USA*

Abstract

This paper describes a new robust and accurate Cartesian-grid treatment for the immersion of solid bodies within a fluid with general boundary conditions. This 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. This results in a new governing equation for the complete domain which preserves the behavior of the original system, but significantly simplifies the numerical implementation and reduces the computational effort. Through a canonical Poiseuille flow problem, it is shown that BDIM is consistent, obtains second-order convergence relative to the kernel width, and is robust with respect to the grid and boundary alignment. The 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.

Keywords: Cartesian-grid, Immersed Boundary Method, Multi-phase flows, Fluid-body interactions

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 exists 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

Email addresses: weymouth@mit.edu (G.D. Weymouth), yue@mit.edu (Dick K.P. Yue)

in a cross flow [3] (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 the forcing were explored such as setting the force to exponentially dampen the error in the velocity field (Dommermuth [5]) or using an explicit Integral/Position feedback controller (Goldstein [6]). However, these methods only enforce their boundary conditions approximately in time. Fadlum [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 [12] and Udaykumar [10], [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. This complexity 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.

We introduce a new immersed boundary formulation of this problem, altering the analytic form of the fluid equations to ensure exact enforcement of the boundary data and maintain physically consistent behavior near the smoothed interface. Reference [16] briefly introduces some of these ideas, and expanding upon that work, § 2.1 of the 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 § 2.2 produce a functional more suitable for use in large-scale simulations. An introductory channel flow example in § 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 (§ 3.1) and free-slip (§ 3.2) bodies are derived. The resulting no-sip 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]. 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 formualtion. A simple to implement second-order solution algorithm for the general form of the meta-equations is discussed in § 3.3.

Finally, this approach is tested using a set of canonical fluids problems: unsteady 2D viscous flow in § 4.1 and unsteady 3D multi-phase flows in § 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 Ω_b and Ω_f and their interface σ_s (see Fig. 1a), find the solution $\Psi_0(\vec{x},t)$ which satisfies

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

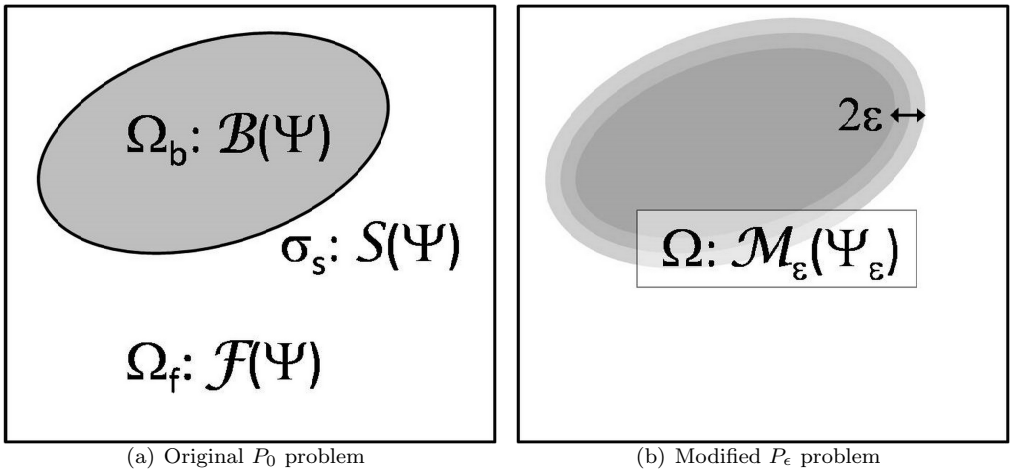
$$\mathcal{F}(\Psi_0,\vec{x},t)=0 \qquad \vec{x}\in\Omega_f \tag{1b}$$

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

where $\mathcal{B}$ and $\mathcal{F}$ are the governing field equations in domains Ω_b and Ω_f respectively, and $\mathcal{S}$ are the interfacial conditions on σ_s. Solid-fluid interaction problems are a special case of Eqs.(1), where $\mathcal{F}$ are the governing equations in the fluid domain Ω_f, $\mathcal{S}$ the boundary condition on the body boundary σ_s, and Ω_b is the body governed by it's own elastic or rigid-body equations $\mathcal{B}$.

In the present work we seek to reformulate Eqs.(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_ϵ problem with a single meta-equation governing the unified domain. Because the boundary data is integrated into the governing field equation, the P_ϵ 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_ϵ is consistent with P_0 as $\epsilon\to 0$; (ii) P_ϵ 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.



(a) Original P_0 problem (b) Modified P_ϵ problem

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

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 $\Omega \equiv \Omega_b \bigcup \Omega_f$. This is done through the use of a nascent delta integration kernel K_ϵ, which has unit zero-th moment and compact support ϵ, i.e., $\int_\Omega K_\epsilon(\vec{x},\vec{y})\ \mathrm{d}\vec{y} \equiv 1$, and $K_\epsilon(\vec{x},\vec{y}) \equiv 0$ for $|\vec{x}-\vec{y}| > \epsilon$. To illustrate the properties of K_ϵ, we define the zero-th moments of the kernel over the sub-domains (which are unsteady for systems with moving interfaces):

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

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

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

Figure 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 ϵ beyond; (b) the values of all the integrals change smoothly; and (c) while δ_ϵ^B and δ_ϵ^F vary from 1 to 0, the integral over the point-boundary σ_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_ϵ to analytically extend the range of each governing equation and smoothly transition between them.

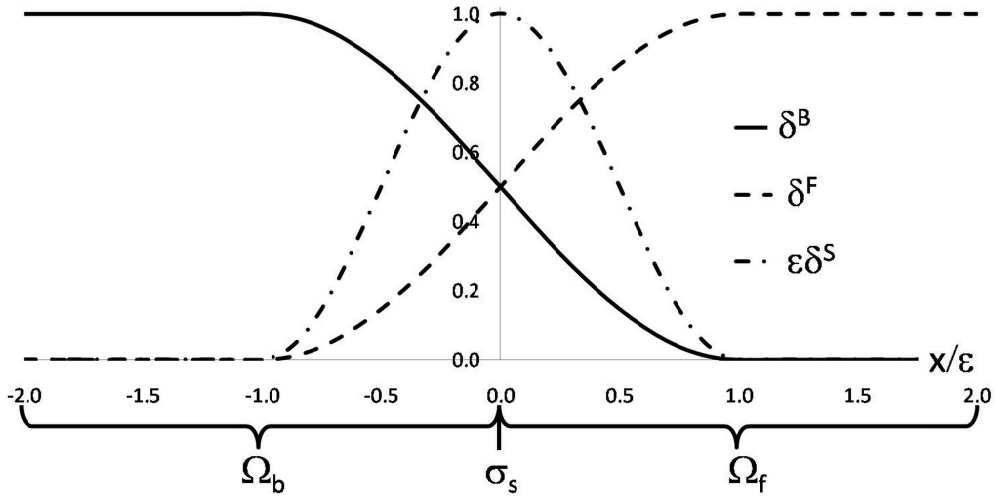


Figure 2: Behavior of the kernel integrals δ_ϵ for a 1D case. For points $x < -\epsilon$ within Ω_b we have δ_ϵ^B=1 and $\delta_\epsilon^S, \delta_\epsilon^F = 0$. Similarly, for $x > \epsilon$, we have δ_ϵ^F=1 and $\delta_\epsilon^S, \delta_\epsilon^B = 0$. Within the smoothing region, points less than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

Convoluting Eqs. (1) with K_ϵ over their respective domains gives

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

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

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

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

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

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

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

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

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

where $\sigma_s^{+\epsilon}$ is the boundary σ_s plus the domain $O(\epsilon)$ distance around it, while $\Omega_{b,f}^{-\epsilon} \equiv \Omega_{b,f} - \sigma_s^{+\epsilon}$ are the respective domains outside the smooth boundary interface. For example, for the simple 1D domain of Fig. 2 with σ_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 Eqs.(5), $\mathcal{M}_\epsilon$ is $O(\epsilon)$ close to the governing equation of each region. Intuitively, one might argue that the solution to (4), Ψ_ϵ, will be close to Ψ_0 (of the original P_0 problem) for small ϵ. The convergence of Ψ_ϵ to Ψ_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 Eqs. (3) are general but the linear systems resulting from their discretization are not sparse due to the kernel dependance on all points within an ϵ 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,

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

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

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

with (4) now replaced by

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

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 δ_ϵ^S are independent of the specifics of the domain topologies and the form of K_ϵ. As shown in Fig. 2, the value of δ_ϵ 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 Ω_b (say), if we define d as the signed distance from any point $\vec{x}$ to nearest point on σ_s, chosen negative within Ω_b, an appropriate approximation for δ_ϵ^B is

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

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

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

Thus Eq. (7) may be posed using Eqs. (8) and (9) for the analytic coefficients. This approach maintains the ϵ-close properties of $\mathcal{M}_\epsilon$ in Eqs. (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

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

$$u = 0 \qquad y = 0, L \tag{10b}$$

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

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

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 Ω_b. The equation for the velocity in Ω_b is simply

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

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 § 3.2, but for systems with continuity conditions the meta-equation formula can be further simplified to

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

giving the meta-equation for the channel flow as

$$\left\{ \delta_\epsilon^B - [1 - \delta_\epsilon^B] \frac{\mathrm{d}^2}{\mathrm{d}y^2} \right\} u_\epsilon = [1 - \delta_\epsilon^B] \frac{f_x}{\nu} \tag{14}$$

where u_ϵ is the solution to this smoothed version of the governing equations. The interpolation function δ_ϵ^B is given by Eq. (8) where the distance function is given by

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

for this simple channel geometry.

We demonstrate four important properties of the boundary data immersion method as expressed by the meta-equation (14): (1) the system P_ϵ is consistent with P_0 when $\epsilon = 0$; (2) when ϵ 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 § 2.2 are of higher accuracy for a given computational cost than the full meta equations of § 2.1.

First, we ensure the boundary data immersion method solves the correct problem by demonstrating that the P_0 and P_ϵ systems are equivalent when $\epsilon = 0$. To do so, we first define a numerical grid with N points $\mathbf{y}^T = \{y^1, y^2, \ldots, y^N\}$, spaced uniformly Δy apart, with $y^1 = 0$ and $y^N = L$. The solution vector $\mathbf{u}_\epsilon^T = \{u_\epsilon^1, u_\epsilon^2, \ldots, u_\epsilon^N\}$ corresponds to the values of the velocity field at $u_\epsilon(y^1), \ldots, u_\epsilon(y^N)$. 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

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

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 ϵ to determine the convergence of u_ϵ to u_0. Using the same uniform grid, $\mathbf{y}$, the solution u_ϵ 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 ϵ. The figure and table show that the numerical solution converges to the analytic solution at an approximately second-order rate with decreasing ϵ (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 α is $O(10^{-6})$ for all the ϵ 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 Eqs. (6) compared to the full integrals of Eqs. (3). As discussed in § 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 Eqs. (3)is m-diagonal, where $m = O(\epsilon/\Delta y)$. For a 3D problem (referencing $O((\epsilon/\Delta y)^3)$ local points), a conservative estimate based on our experience is that obtaining the solution using the simplified Eqs. (6) is a factor of $O(10^3)$ more efficient relative to that using the full integrals Eqs. (3).

To compare the relative accuracy of the two methods the full integral equations are solved numerically using the same grid and values of ϵ as above. The results, shown in Fig. 3(b) and Tab. 1(b), demonstrate that both methods obtain second-order convergence

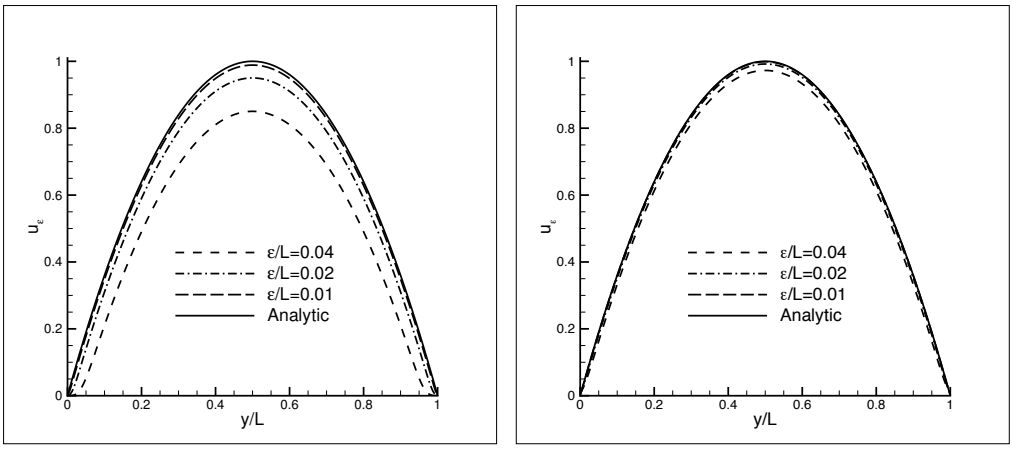
Table 1: Error metrics of the solution to the Poiseuille flow problem for $u_{\max} = fL^2/(8\nu) = 1$ and $N = L/\Delta y = 400$.

ϵ/L	$\lvert E\rvert_1$	$\lvert E\rvert_2$	$\lvert E\rvert_2/(\epsilon/L)^2$	$\lvert E_\alpha\rvert_1 - \lvert E\rvert_1$
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

(a) Convergence and sensitivity tests for Eq. (14)

ϵ/L	$\lvert E\rvert_1$	$\lvert E\rvert_2$	$\lvert E\rvert_2/(\epsilon/L)^2$	$\lvert E_\alpha\rvert_1 - \lvert E\rvert_1$
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

(b) Convergence and sensitivity tests for the full integral method of Eqs. (3)



(a) u_ϵ for Eq. (14)

(b) u_ϵ for the integrals of Eqs. (3)

Figure 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.

with ϵ 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 ϵ/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 Eqs. (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 §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

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

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

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

where ρ is the density, p is the pressure field and and $\vec{r}$ is the aggregate of all the non-pressure terms. In general this may include convective, gravitational, viscous, and modeling terms such as a turbulent sub-grid-scale (SGS) models 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 Δt, giving the velocity impulse equation

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

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

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

$$\partial\vec{P}_{\Delta t} = \int_{t_0}^{t_0+\Delta t} \frac{1}{\rho}\vec{\nabla} p\ \mathrm{d}t\ . \tag{20b}$$

In practice these integral are evaluated with a numerical quadrature but we retain the general form here to highlight that any appropriate implicit or explicit quadrature can be used.

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 $\mathcal{S}$ are not required and we simply substitute Eqs. (17) and (19) into Eq. (13) giving

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

A divergence-free constraint on the velocity field

$$\vec{\nabla} \cdot \vec{u}_\epsilon = 0 \tag{22}$$

applied to this meta-equation gives

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

which is inverted for the pressure p_ϵ.

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

$$\frac{\vec{u}_\epsilon - \vec{u}_\epsilon^0}{\Delta t} = \vec{r}^0 - \frac{1 - \delta_\epsilon^B}{\rho^0} \vec{\nabla} p_\epsilon + \vec{f} \tag{24a}$$

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

where

$$\vec{f} = \delta_\epsilon^B \left[\frac{\vec{U} - \vec{u}_\epsilon^0}{\Delta t} - \vec{r}^0 \right] \, . \tag{25}$$

Equations (24) appear similar to the direct forcing method formulation except for the 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,

$$\frac{\partial p}{\partial n} = \rho \left(r_n - \frac{\partial U_n}{\partial t} \right) \tag{26}$$

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 §4 this new formulation enables stable and accurate pressure force predictions even during nonlinear events such as arise in water entry problem.

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

The 'no-slip' boundary condition of §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 [17]. 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

$$\mathcal{S}_n(\vec{u}) = u_n(\vec{x}_b) - U(\vec{x}_b) = 0 \tag{27}$$

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 σ_s. With that technicality understood, the tangential surface equations are written as

$$\mathcal{S}_\sigma(\vec{u}) = \frac{\partial u_\sigma}{\partial n} = 0 \tag{28a}$$

$$\mathcal{S}_\tau(\vec{u}) = \frac{\partial u_\tau}{\partial n} = 0 \tag{28b}$$

where $\hat{\sigma}$ and $\hat{\tau}$ are the local tangential vectors. Equations (17), (19) and (28) are substituted into Eq. (7) to give

$$\vec{u}_\epsilon + \vec{\mathcal{L}}(\vec{u}_\epsilon) = \delta_\epsilon^B \vec{U} + [1 - \delta_\epsilon^B] \left(\vec{u}_\epsilon^0 + \vec{R}_{\Delta t}(\vec{u}) - \partial \vec{P}_{\epsilon\,\Delta t} \right) \tag{29}$$

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

$$\vec{\mathcal{L}}(\vec{u}) \equiv \delta_\epsilon^S \frac{\partial \vec{u}_\tau}{\partial n} \hat{\tau} + \delta_\epsilon^S \frac{\partial \vec{u}_\sigma}{\partial n} \hat{\sigma}, \tag{30}$$

for the Neumann condition. This LHS term is the only functional difference between Eqs. (21) and (29). A solver with a flag to turn this term on or off, such as discussed in §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 ϵ, Eq. (29) 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 §4.

3.3. Solution method: Time quadrature and operator inversion

To solve the meta-equations presented above, a time quadrature is required. We adopt Huen's second-order method which uses an explicit Euler predictor step followed by an explicit trapezoidal corrector step, an approach commonly used for unsteady flows.

Each time step then requires four inversions; two for the pressure equation and two for the $\mathcal{L}$ velocity operator. A multi-grid method with diagonally preconditioned conjugate-gradient smoothing is used for the pressure equation for computational efficiency. The $\mathcal{L}$ operator is banded lower-diagonal when the points are ordered by their distance from the interface. This is equivalent to using points in the fluid domain to set the values of the 'ghost nodes' in a fitted grid method. Therefore the velocity 'inversion' is accomplished

by back substitution and the computational cost is insignificant compared to the pressure inversion.

Altering an existing orthogonal-grid single-domain solver to run this solution method is extremely straightforward[1]. The sole difficulty in computing flows with immersed solids becomes the need to establish the interface distance function d for use in Eqs. (8) and (9). 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.

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 [18]. 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 $\vec{F}_p$ is to integrate over the Cartesian-grid using the distance function d and the pressure BC Eq. (26) to implicitly determine p on σ_s and perform the surface integral. Reference [18] derives an appropriate one-sided Derivative Informed Kernel (DIK) integration for $\vec{F}_p$ of the form

$$\vec{F}_p = \int_\Omega \left[p - d\hat{n} \cdot \left(\vec{r} - \frac{\vec{U} - \vec{u}_0}{\Delta t} \right) \right] \hat{n} \delta_\epsilon^+ \, \mathrm{d}\vec{x} \tag{31}$$

where δ_ϵ^+ 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. (31) 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

[1] In our experience with multiple 2D and 3D solvers, the alteration typically takes less than a day.

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 $Re = 150$. The present BDIM results for different values ϵ/D are compared to Liu [19] (boundary-fitted grid simulations), Silva [20] (immersed boundary simulations), and Williamson [21] (experimental measurements).

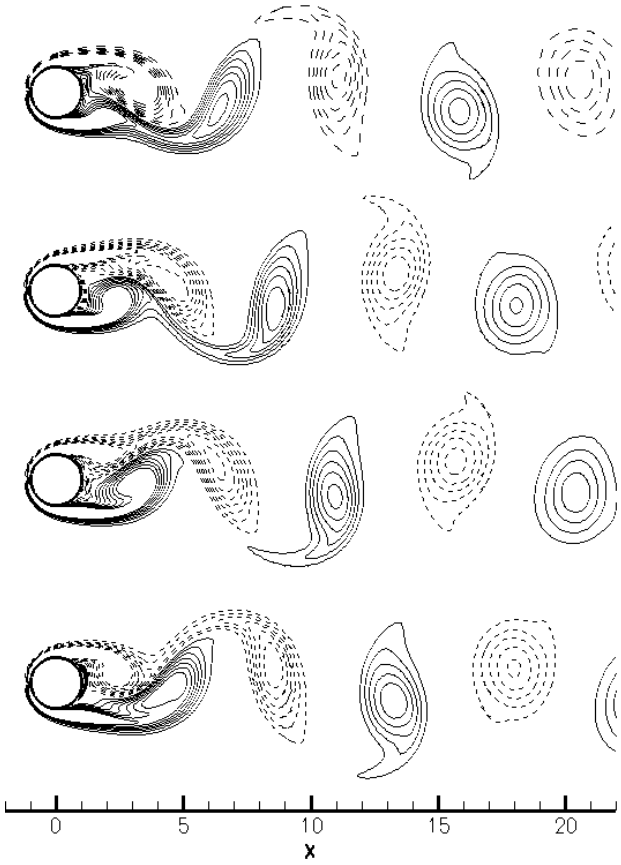
	BDIM					
ϵ/D	0.125	0.0883	0.0625	Liu [19]	Silva [20]	Williamson [21]
St	0.181	0.183	0.183	0.182	0.180	0.183
C_d	0.140 ± 0.04	1.37 ± 0.03	1.35 ± 0.03	1.33 ± 0.03	1.37	-
C_l	0.0 ± 0.59	0.0 ± 0.56	0.0 ± 0.54	0.0 ± 0.53	-	-

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 ϵ/D=0.125, 0.0883, 0.0625 = 2/16, $\sqrt{2}$/16, 1/16, to further establish the convergence properties of the method.

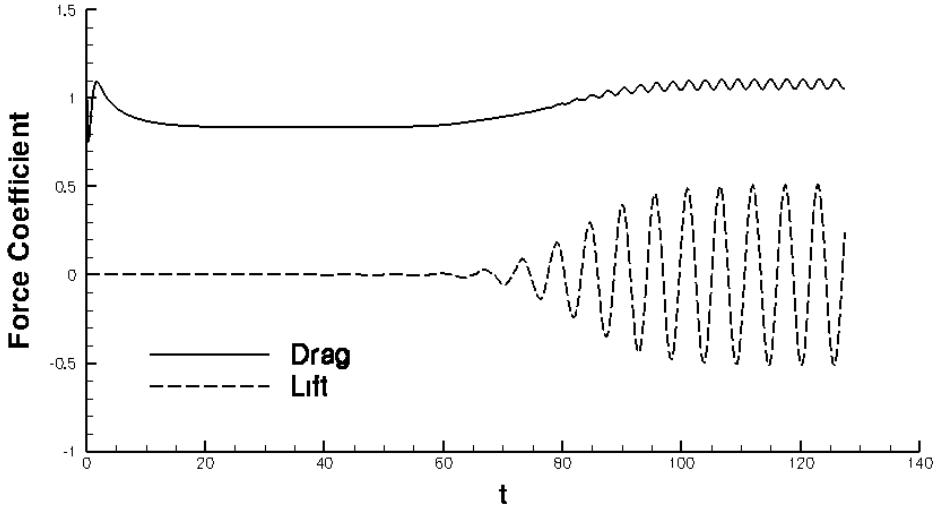
The simulation is run at a Reynolds number of Re=150 which is characterized by an unsteady but laminar and two-dimensional wake. This is done to avoid confusing the analysis of the boundary data immersion method with the additional modeling required to simulate higher Reynolds number turbulent flows. Figure 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. Figure 4(b) shows the time history of the integrated pressure force coefficients on the cylinder (computed using the DIK method and Eq. (31) with the finest value of ϵ/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 [22]. The Strohal number $St = D/UT$ and total lift and drag coefficients obtained using BDIM with different values of ϵ/D, are compared to the boundary-fitted grid simulations of Liu [19], the immersed boundary simulations of Silva [20], and the experimental measurements of Williamson [21] 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 [22] to compare to the total drag values in this table. The current results demonstrate second order convergence and compare 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. [23],[24]), and involve special challenges for simulation



(a) Vorticity contours



(b) Integrated pressure forces

Figure 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 $\rho DU^2/2$.

modeling: 3D moving bodies, nonlinear jumps in fluid conditions, 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

$$\rho(\vec{x},t) = c(\vec{x},t)\rho_{water} + [1 - c(\vec{x},t)]\rho_{air} \tag{32a}$$

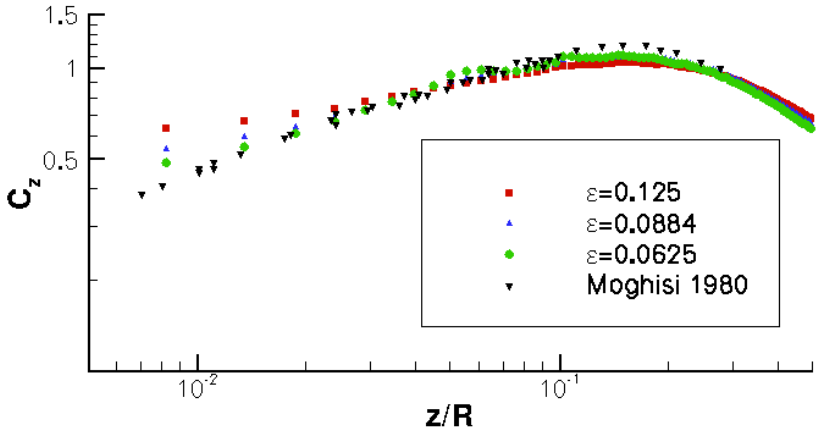
$$\mu(\vec{x},t) = c(\vec{x},t)\mu_{water} + [1 - c(\vec{x},t)]\mu_{air} \tag{32b}$$

for the viscosity. The conservative second-order VOF method of [25] is used to transport the color-function.

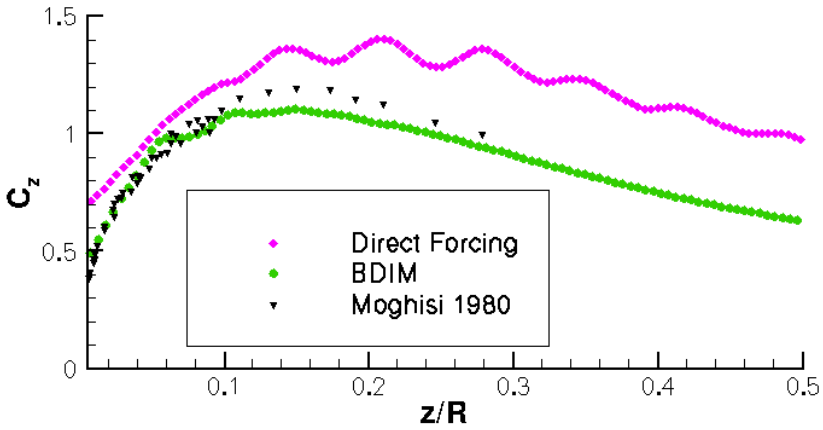
First, BDIM is used to predict the pressure forces experienced by a sphere moving with constant downward velocity U as it impacts the air/water interface. A 3X3X6 diameter computational domain is used with free-slip conditions on the far-field boundaries, no-slip conditions on the immersed sphere, and 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. Figure 5(a) shows BDIM results for progressively refined ϵ/D values, compared to the experimental measurements of [26]. 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 ϵ 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 §3.1, the meta equation for an immersed body with prescribed no-slip boundary conditions diverges from the direct-forcing method of Fadlum only in the treatment of the pressure. To directly evaluate the impact of this modification on the simulation, the test case was run again without the pressure adjustment in the Poisson equation to replicate the direct-forcing methodology. Figure 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 nonlinear 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 [27]. 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 3X3X12 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



(a) ϵ convergence study



(b) Pressure adjustment study

Figure 5: Pressure force coefficients $C_z = F_z(\frac{1}{2}\rho A U^2)$ 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 ϵ compared to the experimental measurements of [26]. 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∼3. The simulations show monotonic convergence with ϵ 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.

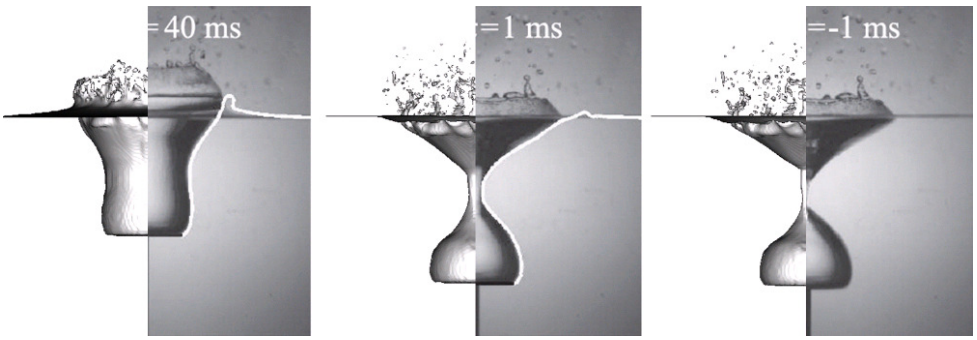


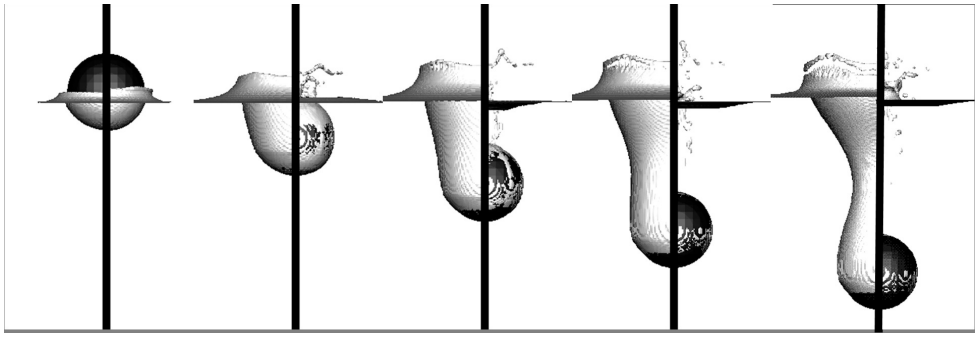
Figure 6: Snapshots of the BDIM simulated impact of a disk on the air/water interface (left) compared to experimental results of [27] (right) and the potential flow results found therein (white line in the first two snapshots on the right). Both the current result and the experimental result show splash-up, crowning, a large well-formed cavity, and pinch-off.

after pinch-off.

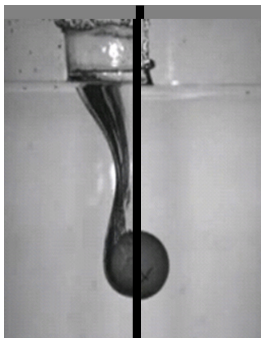
Finally, the ability of BDIM to include general solid-fluid interfacial conditions is used to study the effect of the boundary condition the cavity formed by an impacting sphere. Unlike in the case of a disk, the separation point behind an interface impacting sphere is not fixed and depends on entry 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 [23]. Precise modeling of such surface effects is difficult, but an initial investigation can be made using BDIM and the slip model of Eq. (28). 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 [23] at a comparable Froude number (Fr=1.75). Further research along these lines is in progress.

5. Conclusion

This paper presents an contribution 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 ϵ smoothed interface. By using a finite smoothing width ϵ, BDIM avoids the numerical order of accuracy and stability problems common to sharp-interface methods. This new method is demonstrated to maintain the system dynamics — consistency, second-order convergence with ϵ and negligible modeling error in high-resolution simulations. A number of canonical applications are presented: Poiseuille flow, 2D viscous flow past a circle, and 3D water entry problems



(a) Simulation



(b) Experimental

Figure 7: Simulated snap shots of the sphere and the air/water interface 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 (from [23])

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.

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 [17], [18], [28]. Without any appreciable increase in computational effort, the boundary data immersion method enables high-accuracy and robust simulations of complex problems with significant reduction in the computational complexity. The Boundary Data Immersion Method we describe is general in its approach and should be useful in the broader context involving complex interfaces/boundaries in partial differential equations on numerical grids 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.

[1] C. S. Peskin, Flow patterns around heart valves: A numerical method, Journal of Computational Physics 10 (2) (1972) 252 – 271. doi:DOI: 10.1016/0021-9991(72)90065-4.
URL http://www.sciencedirect.com/science/article/B6WHY-4DDR57R-YK/2/b5a8c317f91802612928b092b1b22b54

[2] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220 – 252. doi:DOI: 10.1016/0021-9991(77)90100-0.
URL http://www.sciencedirect.com/science/article/B6WHY-4DD1NNB-G6/2/cddd393d67643f0a56ebfa8e6c3a148e

[3] L. Zhu, C. S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, J. Comp. Phys. 179,2 (2002) 452–468.

[4] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517. doi:10.1017/S0962492902000077.

[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, Journal of Computational Physics 105 (2) (1993) 354 – 366. doi:DOI: 10.1006/jcph.1993.1081.
URL http://www.sciencedirect.com/science/article/B6WHY-45P128J-5F/2/8bde2bb4eef39c36fda3ec7c0a54f141

[7] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (1) (2000) 35 – 60. doi:DOI: 10.1006/jcph.2000.6484.
URL http://www.sciencedirect.com/science/article/B6WHY-45FC8JY-4M/2/6b24ae68ad18e0ba9af4096f2e8a3117

[8] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, Journal of Computational Physics 209 (2) (2005) 448 – 476. doi:DOI: 10.1016/j.jcp.2005.03.017.
URL http://www.sciencedirect.com/science/article/B6WHY-4G82Y8V-1/2/2898d763ae29c8cd06738d9da0ff619c

[9] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, Journal of Computational Physics 228 (18) (2009) 6617 – 6628. doi:DOI: 10.1016/j.jcp.2009.06.003.
URL http://www.sciencedirect.com/science/article/B6WHY-4WH8CFM-2/2/909309b710031f07e9e0a9d9e02efe39

[10] H. S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface cartesian grid method for simulating flows with complex moving boundaries, Journal of Computational Physics 174 (1) (2001) 345 – 380. doi:DOI: 10.1006/jcph.2001.6916.
URL http://www.sciencedirect.com/science/article/B6WHY-45BC23K-16/2/1fba2b86e11821c27fc96c41a57cd9e7

[11] R. J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal.

[12] F. Gibou, R. P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the poisson equation on irregular domains, Journal of Computational Physics 176 (1) (2002) 205 – 227. doi:DOI: 10.1006/jcph.2001.6977.
URL http://www.sciencedirect.com/science/article/B6WHY-456JP19-B/2/de98099cb9a7deb6e3e470a9fc5290c6

[13] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface cartesian grid method i: An easily implemented technique for 3d moving boundary computations, Journal of Computational Physics

210 (1) (2005) 1 – 31. doi:DOI: 10.1016/j.jcp.2005.03.031.
URL http://www.sciencedirect.com/science/article/B6WHY-4G7GG23-F/2/583ddb5c91bc31e82aedc346c1f1557c

[14] Y.-H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, Journal of Computational Physics 192 (2) (2003) 593 – 623. doi:DOI: 10.1016/j.jcp.2003.07.024.
URL http://www.sciencedirect.com/science/article/B6WHY-49JHHVM-1/2/f4c4ed2c938d6204dad3f1dcf1eb6e2b

[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] 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.

[17] G. D. Weymouth, K. Hendrickson, T. OShea, 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.

[18] 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).

[19] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, Journal of Computational Physics 139 (1998) 35–57.

[20] A. L. E. Silva, A. Silveira-Neto, J. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, Journal of Computational Physics 189 (2003) 351–370.

[21] C. H. K. Williamson, Vortex dynamics in the cylinder wake, Annual Review of Fluid Mechanics 28 (1996) 477–539. doi:10.1146/annurev.fl.28.010196.002401.

[22] R. D. Henderson, Details of the drag curve near the onset of vortex shedding, Physics of Fluids 7 no. 9 (1995) 2102–2104.

[23] T. T. Truscott, A. H. Techet, A spin on cavity formation during water entry of hydrophobic and hydrophilic spheres, Physics of Fluids.

[24] H. Yan, Y. Liu, J. Kominiarczuk, D. K.-P. Yue, Cavity dynamics in water entry at low froude numbers, Journal of Fluid Mechanics 641.

[25] G. Weymouth, D. K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on cartesian-grids, Journal of Computational Physics 229 (8) (2010) 2853 – 2865. doi:DOI: 10.1016/j.jcp.2009.12.018.

[26] M. Moghisi, P. T. Squire, An experimental investigation of the initial force of impact on a sphere striking a liquid surface, Journal of Fluid Mechanics 108 (1981) 133–146.

[27] R. Bergmann, D. van der Meer, M. Stijnman, M. Sandtke, A. Prosperetti, D. Lohse, Giant bubble pinch off, Physical Review Letters 96 (2006) 154505. doi:10.1103/PhysRevLett.96.154505.

[28] K. Hendrickson, D. K.-P. Yue, Models for simulating breaking wave in computational ship hydrodynamics, in: 27th Symposium on Naval Hydrodynamics, 2008.