Oral presentation | Multi-phase flow

Multi-phase flow-I

Mon. Jul 15, 2024 10:45 AM - 12:45 PM Room D

[1-D-03] Exact interface velocity in immersed boundary method

*Kun Zhou[1] （1. Wuhan University of Science and Technology）

Keywords: immersed boundary method, fluid-structure interaction, numerical method

Twelfth International Conference on
Computational Fluid Dynamics (ICCFD12),
Kobe, Japan, July 14-19, 2024

Exact interface velocity in immersed boundary method

Kun Zhou* and Zhao-Qin Yin**
Corresponding author: zhou.kun@wust.edu.cn
* Department of Energy & Power Engineering, Wuhan University of Science and Technology, Wuhan, China.
** Institute of Fluid Measurement and Simulation, China Jiliang University, Hangzhou, China.

Abstract: Immersed boundary method (IBM) is one of the most popular computational methods in dealing with fluid-structure interaction problem. In IBM, both fluid and solid domains are discretized in a fixed Eulerian mesh, and the fluid-solid interface is tracked by discretized Lagrangian markers, which are independent of the Eulerian mesh and are free to move conforming to the solid dynamics. Within the IBM framework, a popular method to model the interaction force between the fluid and solid is to introduce a feedback model as $\mathbf{f}_{\mathrm{IBM}} = \alpha(\mathbf{U} - \mathbf{u})$, where α is a proportional coefficient, $\mathbf{U}$ and $\mathbf{u}$ denote the solid and fluid velocity at the interface. Despite of its popularity and various improving efforts, the present direct forcing IBM is not able to completely alleviate the discrepancy between the fluid and solid velocities at the interface, because it is the discrepancy per se to provide the interaction force. Through analyzing the typical pressure correction scheme generally used in IBM, we identify two additional obstacles for achieving the exact interface velocity, one from the diffusion step, and the other from the divergence free correction step. By taking the diffusion into account, we develop a new method that achieves exact interface velocity for pure shear flow. The new method performs better in precision than the prevailing method in all test cases.

Keywords: Immersed Boundary Method, Direct Forcing, Interfacial Velocity Error.

1 Introduction

Immersed boundary method (IBM) is in vogue for the simulation of fluid-structure interaction [1, 2, 3, 4]. IBM is a fictitious domain method which uses a universal control equation to model both the fluid and structure dynamics along with additional model to take into account the interface condition. The whole domain is discretized by the fixed Eulerian mesh, while the interface is tracked in a Lagrangian way. IBM was first introduced by Peskin [5] to simulate flow interaction with soft tissue. The interaction force is determined by the configuration of interface through elastic constitutive relation. The original IBM encounters problem when dealing with rigid bodies because the interaction force cannot be deduced from the interface deformation (there is no deformation for a rigid body). Instead of relying on the interface deformation, Goldstein et al. [6] suggested to use a proportional-integral (PI) feedback model to calculate the interaction force, i.e., $\mathbf{f}_{\mathrm{IBM}} = \alpha \int \mathbf{U}_\triangle \mathrm{d}t + \beta \mathbf{U}_\triangle$, where α and β are constants, $\mathbf{U}_\triangle = (\mathbf{U} - \mathbf{u})$ denotes the interface velocity error between the solid and fluid velocities at the interface. The requirement of tuning the parameters α and β to obtain stable and precise control hindered the broad application of the PI model. Mohd-Yusof [7] came across a very simple model $\mathbf{f}_{\mathrm{IBM}} = \mathbf{U}_\triangle/\Delta t$, where Δt is the time step. The force was expected to drive the velocity discrepancy $\mathbf{U}_\triangle$ to zero in the next time step so as to achieve exact interface velocity, as claimed by the developer [7]. The idea of direct-forcing (as coined by the developer) is very appealing, and has attracted a lot of relevant work. Uhlmann [8] adopted the idea of direct forcing, but calculated the force on the solid interface (Lagrangian framework) first and then interpolated the force back into fluid domain (Eulerian framework). Such a back-and-forth process turns out to be very efficient, which was later proved to be a least-squares interpolation between the Eulerian and Lagrangian frameworks [9].

The direct forcing IBM (DF-IBM) has a few drawbacks. First of all, it is not possible to achieve the idea condition $\mathbf{U}_\triangle = 0$, otherwise no feedback force is introduced at all. This problem is not so critical, because it is shown that $\mathbf{U}_\triangle$ approaches zero as reducing the time step Δt [9]. A more critical issue is that the direct forcing brought a wrong “intuition”, in other words, it seduced one to believe that $\mathbf{f}_{\mathrm{IBM}} = \mathbf{U}_\triangle/\Delta t$ represents the “true” interaction force. Misguided by such a wrong intuition, a lot of time and efforts have been wasted on trying to satisfy unnecessary balance conditions [9]. The direct forcing model is actually a simple proportional feedback with $\alpha = 0$ and $\beta = 1/\Delta t$ in term of the PI model of Goldstein et al. [6]. A simple proportional feedback is known to have steady state error, in other words $\mathbf{U}_\Delta$ is never zero.

Here we analyze underlying causes that prevent exact interface velocity in IBM, and develop a new method that does achieve exact interface velocity for pure shear flow. We first introduce the usual IBM method before derive the new method. Then we compare the two methods through a series simulations. This paper also comes with an appendix aiming to analyze the velocity discrepancy in direct forcing IBM.

2 Immersed boundary method

2.1 Model equation

The momentum equation for incompressible fluid in the whole domain (including the space occupied by particles) is

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \nu(\nabla^2 \mathbf{u}) + \mathbf{f}_{\mathrm{IBM}} \tag{1}$$

where $\mathbf{u}$, p, ν are the fluid velocity, pressure, density, kinematic viscosity, respectively, $\mathbf{f}_{\mathrm{IBM}}$ is the force term to take into account the fluid-particle interaction, and the compact notations $\mathbf{f}$ and $\mathbf{f}_L$ are clear from the equations. The fluid density ρ_f is not explicitly included, which is assimilated into relevant terms, such as the pressure and the interaction force $\mathbf{f}_{\mathrm{IBM}}$.

2.2 Numerical scheme

The projection method [11] with explicit convection and implicit diffusion is broadly used in DF-IBM. For such a method, a typical kth Runge-Kutta time marching step is outlined as follows [8]

$$\tilde{\mathbf{u}} = \mathbf{u}^{k-1} + \Delta t(2\alpha_k \nu \nabla^2 \mathbf{u}^{k-1} - 2\alpha_k \nabla p^{k-1} - \gamma_k[(\mathbf{u} \cdot \nabla)\mathbf{u}]^{k-1} - \xi_k[(\mathbf{u} \cdot \nabla)\mathbf{u}]^{k-2}), \tag{2a}$$

$$\mathbf{f}_{\mathrm{IBM}} = \mathbf{I}_{\mathrm{mask}} \frac{a(\mathbf{U} - \tilde{\mathbf{u}})}{\Delta t}, \tag{2b}$$

$$\nabla^2 \mathbf{u}^* - \frac{\mathbf{u}^*}{\alpha_k \nu \Delta t} = -\frac{1}{\alpha_k \nu}\left(\frac{\tilde{\mathbf{u}}}{\Delta t} + \mathbf{f}_{\mathrm{IBM}}\right) + \nabla^2 \mathbf{u}^{k-1}, \tag{2c}$$

$$\nabla^2 \phi^k = \frac{\nabla \cdot \mathbf{u}^*}{2\alpha_k \Delta t}, \tag{2d}$$

$$\mathbf{u}^k = \mathbf{u}^* - 2\alpha\Delta\nabla\phi^k, \tag{2e}$$

$$p^k = p^{k-1} + \phi^k - \alpha_k \Delta t \nu \nabla^2 \phi^k, \tag{2f}$$

where α_k, γ_k and ξ_k are coefficients for a specific Runge-Kutta method (such as the third order method [12]). Equation (2b) calculates the IBM force through the velocity difference between the specified velocity $\mathbf{U}$ and the intermediate velocity $\tilde{\mathbf{u}}$. Here, an additional constant a is added, which is absent in the original DF-IBM [7]. Generally, $\mathbf{U}$ is only known in the Lagrangian framework, and interpolation between the Lagrangian and the Eulerian framework is necessary. Various interpolations have been proposed, which actually determine one of the major differences among the various methods. Here, it is assumed that $\mathbf{U}$ is also defined in the Eulerian grid, so all the variations relevant to the interpolation are dormant. In other words, the above model (2b) completely eliminates the errors of spatial interpolation. The term $\mathbf{I}_{\mathrm{mask}}$ denotes a mask function that sets all IBM force to zero except those at the interface locations. The introduction of the mask function is for notational convenience only.

Instead of analyzing the above typical numerical scheme, we investigate the following simplified model

$$\tilde{\mathbf{u}} = \mathbf{u}^{k-1} + \Delta t(\mathbf{u} \cdot \nabla)\mathbf{u} \tag{3a}$$

$$\mathbf{f}_{\text{IBM}} = \mathbf{I}_{\text{mask}} \frac{a(\mathbf{U} - \tilde{\mathbf{u}})}{\Delta t}, \tag{3b}$$

$$\nabla^2 \mathbf{u}^* - \frac{\mathbf{u}^*}{\nu \Delta t} = -\frac{1}{\nu \Delta t}(\tilde{\mathbf{u}} + \Delta t \mathbf{f}_{\text{IBM}}), \tag{3c}$$

$$\nabla^2 p^k = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \tag{3d}$$

$$\mathbf{u}^k = \mathbf{u}^* - \Delta t \nabla p^k. \tag{3e}$$

This is a first order time marching scheme, which preserves the main characteristics of the pressure correction method, such as explicit convection term, implicit diffusion term. Throughout this paper, we assume the Lagrangian and the Eulerian mesh are aligned with each other, which eliminates complexity of the interpolation between them. Although such an assumption is very restrictive in practice, the analysis here focuses on the dynamical interplay of fluid and solid, spatial inaccuracy is believed not to change the temporal coupling qualitatively.

An interesting but generally unnoticed fact is that the introduction of the IBM force is only an indirect way to specify the intermediate velocity $\tilde{\mathbf{u}}$ in the interface region. This can be readily seen by inserting Eq. (2b) into Eq. (2c). The parentheses on the right hand side in Eq. (2c) is then

$$(\tilde{\mathbf{u}} + \Delta t \mathbf{f}_{\text{IBM}}) = \begin{cases} \tilde{\mathbf{u}}, & \text{region away from interface} \\ a\mathbf{U} + (a-1)\tilde{\mathbf{u}}, & \text{interface region.} \end{cases} \tag{4}$$

It is clear that the intermediate velocity $\tilde{\mathbf{u}}$ is revised in the interface region, where a weighted average of $\mathbf{U}$ and $\tilde{\mathbf{u}}$ is actually used. Specifically, if $a = 1$, then $\tilde{\mathbf{u}} = \mathbf{U}$, which means only the known interface velocity U is retained in $\tilde{\mathbf{u}}$, all previous value of $\tilde{\mathbf{u}}$ is discarded. When $a \to \infty$, U and previous $\tilde{\mathbf{u}}$ take equation weight in determining the final $\tilde{\mathbf{u}}$, which is used in the next implicit step for diffusion.

The objective of obtaining $u = U$ at the interface is the key for accurate simulation of the fluid-solid coupling. In the prevailing direct forcing IBM as outlined above, there are obstacles preventing the satisfaction of $u = U$. First, the implicit diffusion process is likely to change the interface velocity. Second, the divergence free correction (Eq. (3e)) will change the interface velocity further. Next, we will introduce a new method to address the first obstacle, which is able to render exact interface velocity when the second obstacle is absent.

2.3 New method

Generally, the interface problem has codimension one, such as a line interface in 2D, a surface interface in 3D. In the normal direction to the interface, the diffusion can be approximated as 1D

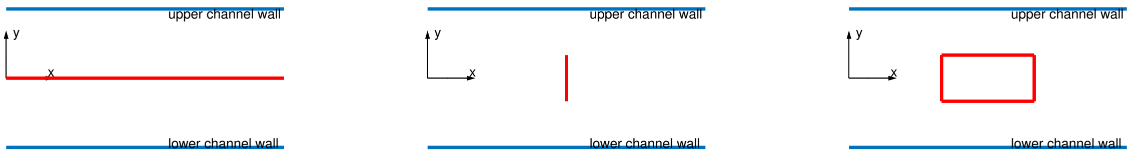
$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}. \tag{5}$$

Using implicit Euler scheme for time, implicit center difference for the diffusion, we obtain the following discretized evolution equation for the velocity at the interface location

$$\frac{u_0^{(n+1)} - u_0^{(n)}}{\Delta t} = \frac{\nu}{h^2}(u_1^{(n+1)} + u_{-1}^{(n+1)} - 2u_0^{(n+1)}), \tag{6}$$

where h is grid size. We use $u_0^{(n)}$, $u_1^{(n)}$, and $u_{-1}^{(n)}$ to denote the velocity at the interface and at the adjacent grid on both sides in the previous time step (denoted by the superscript $^(n)$). Correspondingly, the superscript $^{(n+1)}$ denotes the values at the next time step. Suppose the objective is to choose $u_0^{(n)}$ at previous time step so that at the next time step $u_0^{(n+1)} = U$. Therefor $u_0^{(n)}$ is required to be

$$\begin{aligned} u_0^{(n)} &= -\frac{\nu \Delta t}{h^2}(u_1^{(n+1)} + u_{-1}^{(n+1)}) + \left(1 + \frac{2\nu \Delta t}{h^2}\right) U \\ &\approx -\frac{\nu \Delta t}{h^2}(u_1^{(n)} + u_{-1}^{(n)}) + \left(1 + \frac{2\nu \Delta t}{h^2}\right) U. \end{aligned} \tag{7}$$



(a) Horizontal plate

(b) Horizontal plate

(c) Horizontal plate

Figure 1: Flow configurations

Since $u_1^{(n+1)}$ and $u_1^{(n+1)}$ are unknown values at the next time step, we approximate them by their previous time values. After merging the above equation into the pressure projection scheme, we have the new IBM

$$(\tilde{\mathbf{u}} + \Delta t \mathbf{f}_{\text{IBM}}) = \begin{cases} \tilde{\mathbf{u}}, & \text{region away from interface} \\ \left(1 + \frac{2\nu\Delta t}{h^2}\right) \mathbf{U} - \frac{\nu\Delta t}{h^2}(\tilde{\mathbf{u}}^+ + \tilde{\mathbf{u}}^-), & \text{interface region .} \end{cases} \tag{8}$$

This new model shares one similarity with the prevailing model (4), both of which use a weighted average of objective velocity U and some available velocity. The most striking difference is that the prevailing method calculates the "available velocity" from values at the interface position, while the new method calculates it from values on adjacent locations. Loosely speaking, the prevailing method approximates the interface velocity through the "temporal direction", while the new method approximates the interface velocity through the "spatial direction". It might be possible to approximate along the characteristic direction by combining both temporal and spatial directions, which is not yet tried in this paper. Another difference is that the new model does not introduce additional weight parameter a (which is proved to be a relaxation factor [9]). The coefficient in front of U one plus a correction term which is the squared ratio between the diffusion scale and the grid size. In the next section we will compare the precision of the two methods.

3 Numerical simulation results

In order to investigate the performance of the new method, three flow configurations have been considered (Figure 1). To reduce the simulation time, all three configurations are two dimensional, and the conclusions obtained are believed to apply to three dimensions also.

The first configuration is a thickless plate lying along the center of a channel, where both still and oscillating plate have been considered separately. Both channel walls are moving at the same constant velocity. Periodic boundary conditions are used in the two open boundaries. No pressure gradient is applied along the flow direction. The flow is homogeneous along the flow direction. The exact velocity profile is like a rotated V. The effect of the plate on the flow is modeled by both the prevailing IBM method and the new method. This setting is an ideal case to test the new method to see if the exact velocity interface is achieved at the plate position, since the second obstacle related to the pressure correction is absent.

The second configuration is a thickless plate lying across in the channel. No-slip boundary condition is used for both channel walls. A velocity inlet with the parabolic profile is adopted on the left, and natural outflow condition is applied on the right. The vertical plate is modeled by two methods.

The third configuration is a box lying in the channel. This is a combination of the previous two configurations. The channel boundary conditions are the same as those in the second configuration.

3.1 Horizontal plate

For the horizontal plate configuration, the flow is homogeneous along the channel. Figure 2 gives an example how the streamwise velocity contour looks like. Figure 3 shows the velocity profiles across the channel, obtained by the new method and prevailing method with two different time steps. The new method is independent of the time step. At the plate position $y = 0$, the new method gives zero velocity

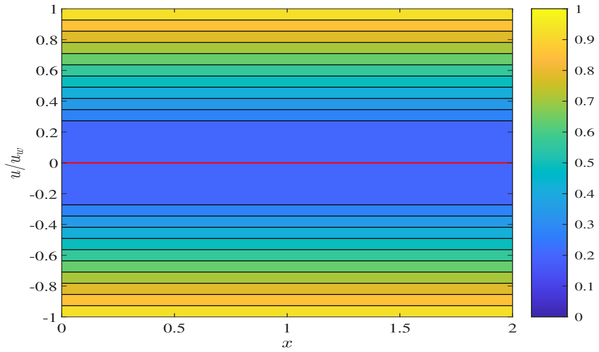


Figure 2: Streamwise velocity contour for a still horizontal plate located along the channel centerline.

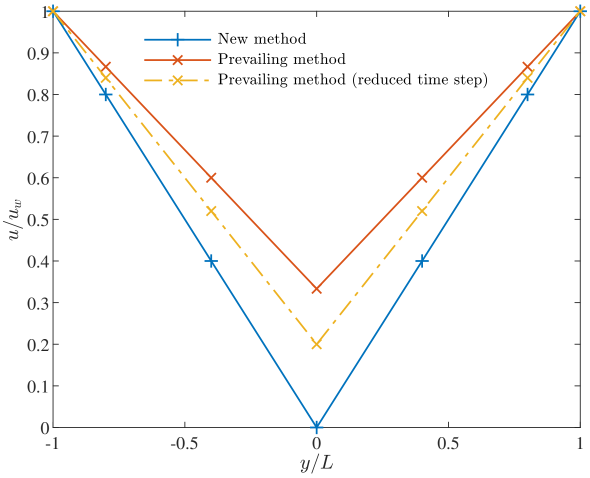


Figure 3: Velocity profile for a still horizontal plate located along the channel centerline.

as wanted, but the prevailing method shows apparent error which decreases when reducing the time step. More careful analysis shows that the error in the prevailing method is actually inversely proportional to the time step [9]. While the prevailing method cannot achieve the exact interface velocity at any time step, the new method does achieve the exact interface velocity at any time step.

We also considered the second Stokes problem when the plate is oscillating at a give frequency ω on the plate plane. For this case, there is no need to move the Lagrangian discretization points on the plate, which alleviates the extra layer complexity of interpolation between the Eulerian and Lagrangian mesh. Figure 4 shows the velocity discrepancy at the plate location for $\omega = 10$. Despite that neither method can render the exact interface velocity, the new method produces substantial smaller error than the prevailing method. The error from both method decreases when reducing the time step or refining the grid. If increasing the oscillation frequency without revising the time step or mesh, the error also increases. Figure 5 shows the velocity profile across the oscillating plate. Due to symmetry, only result on one side above the plate is given. Besides the simulation results, the analytical result is also provided. The profile from the the new method agrees with the analytical result very well. Despite the large error around the plate location ($y = 0$), the prevailing method also captures the general trend well away from the plate. This seems to suggest that the velocity discrepancy at the interface does not propagate deeply into fluid domain, and the error due to IBM tends to dies away from the interface. The new method developed here takes advantage of this property, which uses known velocities from interface neighbors instead of those at the interface to improve the precision, because the neighbor values are believed to have less “contaminated” error those at the interface.

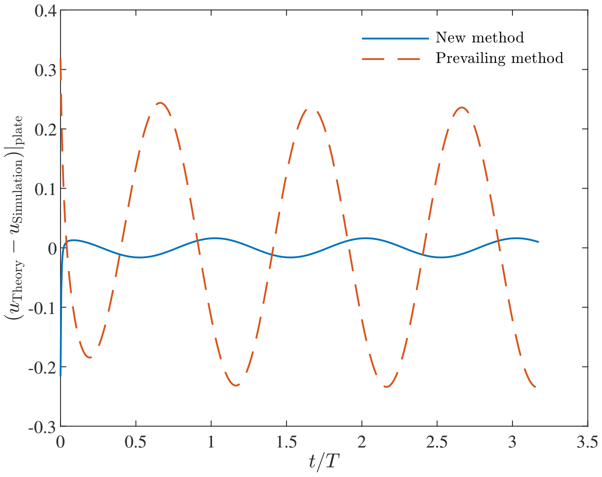


Figure 4: Fluid solid velocity discrepancy at the plate location for the second Stokes problem with oscillation frequency $\omega = 10$.

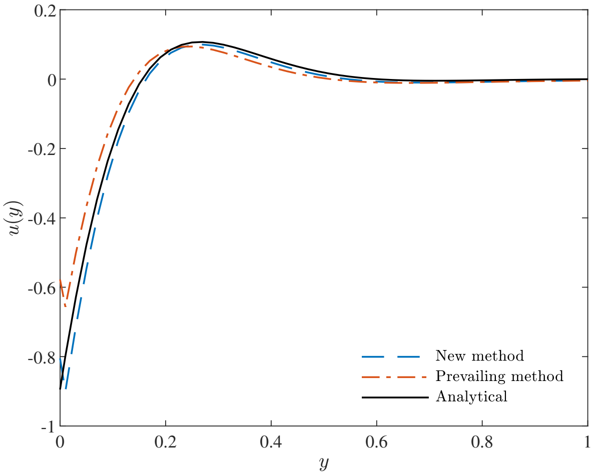
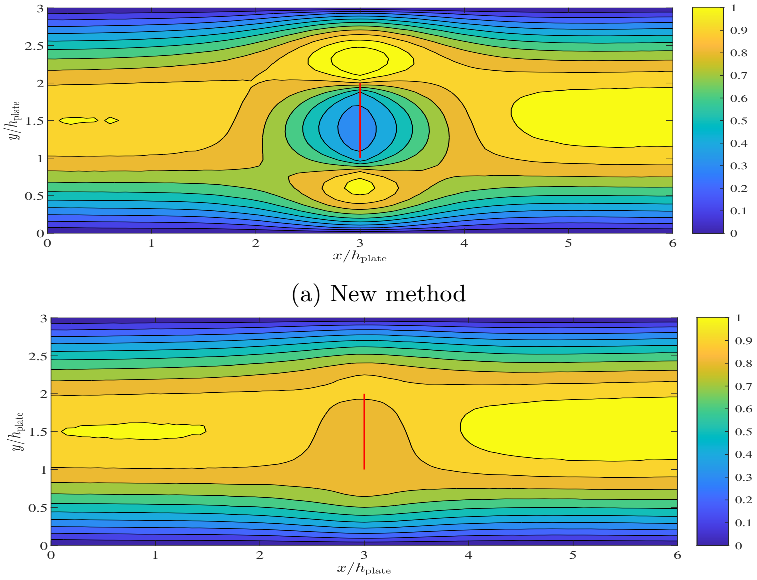


Figure 5: Velocity profile for the second Stokes problem at $\omega = 10$.



(a) New method

(b) Prevailing method

Figure 6: Streamwise velocity contour for a vertical plate located across the channel at Reynolds number Re $= 0.5$.

3.2 Vertical plate

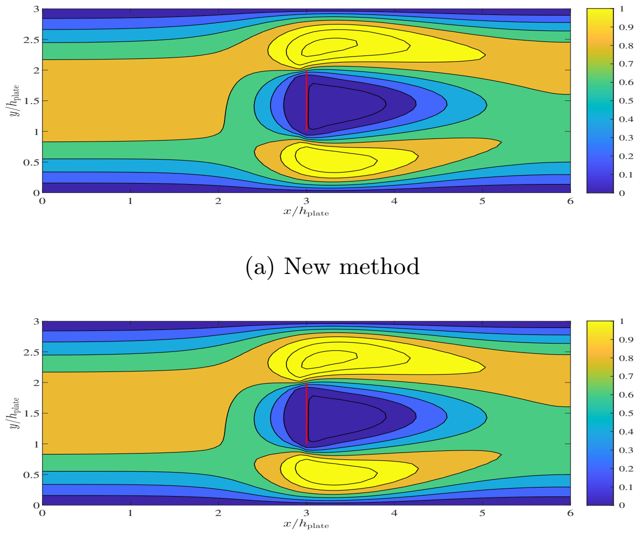
Figure 6 shows the streamwise velocity contour for the vertical plate configuration for Reynolds number Re $= 0.5$ (which is defined by the plate height and the channel inlet velocity). In the prevailing method, the fluid flow barely “sees” the plate, it just flows through the plate. The method fails completely. While in the new method, the flow slows down around the plate, but there is still noticeable flow penetration through the plate. Neither method provides satisfactory result. Figure 7 shows the contour for Re $= 50$. Both methods give very similar results, where around the plate the flow has been blocked as wanted. The precision of IBM is known [9] to deteriorate for Re < 1. Here the same has been observed.

3.3 Box

Figure 8 shows the streamwise velocity contours for the box configuration for Reynolds number Re $= 5$. Here IBM only apples to the box boundary. At the top and bottom boundaries of the box, the new method satisfies the zero velocity much better than the prevailing method. However, unrealistic fluid circulation is observed in both velocity contours. Although the new method is able to satisfy the interface velocity condition very well, it cannot effectively prevent the fluid penetrating the vertical plate, which results in fluid circulation inside the box. Hence, we also tried to implement IBM also inside the box mesh points. The corresponding results are show in Figure 9. Now the fluid circulation inside the box has been suppressed substantially in both methods. In order to compare the two methods in a more quantitative way, velocity profiles across the box are overlaid in the same plot as in Figure 10. The plateau corresponds to the region inside the box. Both methods produce very similar magnitude of non-zero velocity inside the box, which means the two methods have similar precision in dealing with the fluid flow inside the box. However, the new method gives much sharper transition across the box boundary, which is a much wanted property so as to achieve more accurate shear stress at the interface.

4 Conclusion

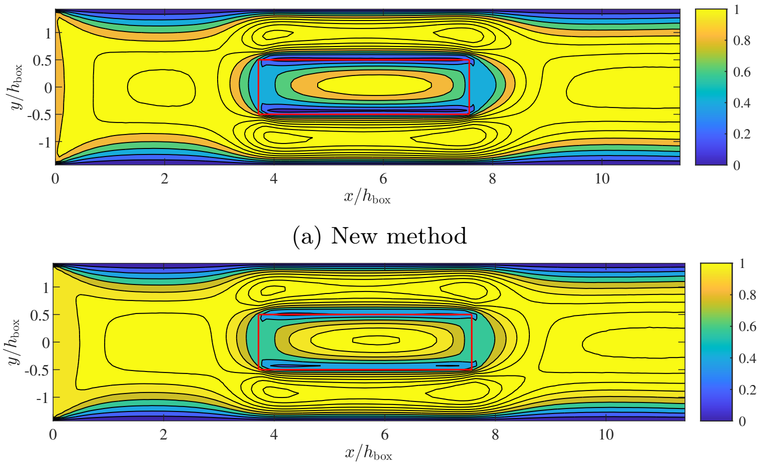
Direct forcing IBM is very popular in dealing with fluid-structure interaction simulation. Generally, a restoring force proportional to the fluid solid velocity discrepancy at the interface is introduced so as to achieve the interface velocity condition, i.e., same fluid and solid velocity. However, there is always velocity discrepancy in the direct forcing method, because generating the restoring force relies on the very existence of the velocity discrepancy. It is worth pointing out that the direct forcing is actually the



(a) New method

(b) Prevailing method

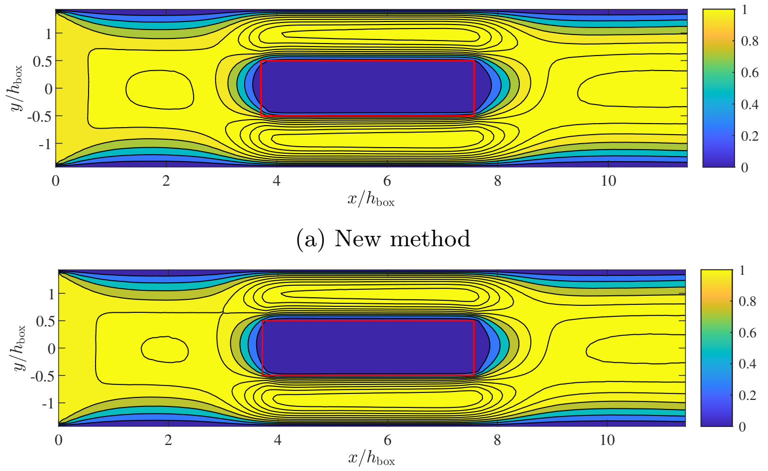
Figure 7: Streamwise velocity contour for a vertical plate located across the channel at Reynolds number Re = 50.



(a) New method

(b) Prevailing method

Figure 8: Streamwise velocity contour for a box located in the channel at Reynolds number Re = 5, where IBM apples only at the box boundary.



(a) New method

(b) Prevailing method

Figure 9: Streamwise velocity contour for a box located in the channel at Reynolds number Re = 5, where IBM applies both at the box boundary and inside.

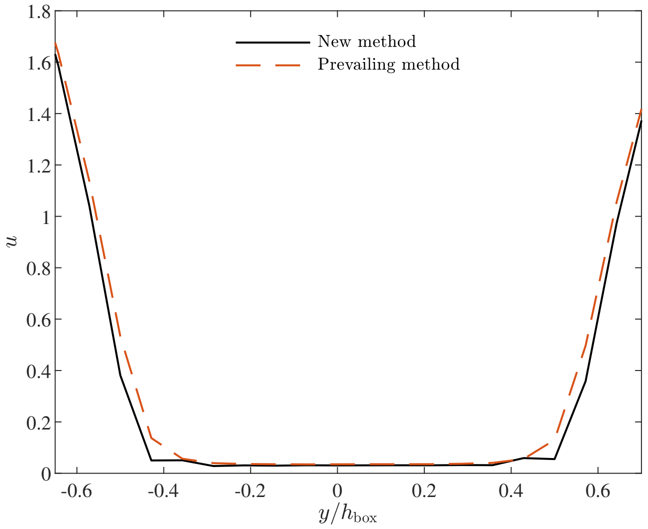


Figure 10: Streamwise velocity profile across the box center.

proportional feedback control, and it is well known that there is always a steady state error related to such control.

Through analyzing a typical pressure projection scheme, which is usually used in direct forcing IBM, we realize that the direct forcing actually resets the intermediate velocity (which is obtained after finishing the convection step) at the interface region through a weighted average of two velocities, i.e., the available fluid and solid velocities. The weight is a free parameter. A larger weight for the solid velocity tends to increase the convergence rate towards exact interface velocity, but encounters more restrict stability problem (**??**). In the projection scheme, after the IBM step aiming for exact interface velocity there are still two steps that are going to revise the fluid velocity at the interface, i.e., diffusion and divergence free correction. Through considering simple 1D diffusion, we derived a new IBM method that does not introduce the IBM force but instead directly specify the intermediate velocity at the interface.

The new method has been compared with the prevailing direct forcing method for three flow configurations, i.e., horizontal plate, vertical plate and box. The new method renders exact interface velocity for the horizontal plate. For the vertical plate and box problem, the new method exhibits similar drawbacks as the prevailing method for not being able to achieve exact interface velocity. For all cases, the new method outperforms the prevailing method in precision. This work solves the first obstacle for achieving the exact interface velocity, i.e., the diffusion step, and further work needs to be done address the second obstacle relating to the divergence free correction.

5 Acknowledgment

Support from NSFC grant 11972335 is gratefully acknowledged.

A Analytical solution of 1D diffusion equation for direct forcing IBM

In direct forcing IBM, an unknown force is introduce to model the fluid-solid interaction. The magnitude of the force is determined implicitly by requiring the final fluid and solid velocity to match at the interface. Here we solve the 1D diffusion equation with a given delta source so as to build up a direct relation between the velocity field and the applied delta force. The analytical solution can be used to analyze the error in direct forcing IBM. To be more specific, we can insert an IBM force model, such as $\mathbf{f}_{\text{IBM}} = \frac{a(\mathbf{U}-\tilde{\mathbf{u}})}{\Delta t}$, into the analytical solution to obtain an algebraic equation for the fluid velocity $\tilde{\mathbf{u}}$. It is then straightforward to see how the velocity discrepancy (or error) $\mathbf{U} - \tilde{\mathbf{u}}$ depends on various parameters, such as time step, grid size, and viscosity. Below the solutions for a still and oscillating thickless plate have been derived separately. This appendix can be seen as an improvement of the error analysis presented in Zhou and Balachandar [9].

A.1 Steady 1D diffusion equation

Consider the 1D diffusion equation for velocity u with a concentrated force f at the middle of the domain

$$\nu \frac{\mathrm{d}^2 u}{\mathrm{d}y^2} = -\frac{f}{h_{\text{unit}}}\delta(y), \quad -L < y < L \tag{9a}$$

$$u(L) = u_{w^+}, u(-L) = u_{w^-}. \tag{9b}$$

On the right hand side, h_{unit} is introduced to render the equation dimensionally correct. For 1D, $h_{\text{unit}} = 1$. For 2D, $h_{\text{unit}} = h$, where h is the x-direction grid size. For 3D, $h_{\text{unit}} = h^2$, where h^2 denotes the grid element area in the x-z plane.

If the force f is known, this equation can be easily solved analytically as shown below. If the force f is unknown, but insteady the condition $u(0) = u_p$ is imposed, then this equation models the viscous channel flow along a thickless parallel plate locating in the middle, and the plate velocity is u_p.

A.2 Analytical solution

The differential equation with known f can be solved analytically by separating the domain into two parts $y < 0$ and $y > 0$. In each part the solution is a linear function with two unknown constants. At

$y = 0$, the two solutions match; and their derivatives u'_L and u'_R (which are constant in each parts) satisfy the jump condition

$$u'_R - u'_L = -\frac{f}{\nu h_{\text{unit}}}. \tag{10}$$

Through grouping the two boundary conditions and the two matching conditions, the four unknown constants can be solved to render the solutions

$$u = \begin{cases} u_0 + (u_{w^+} - u_0)\frac{y}{L}, & y \geq 0, \\ u_0 - (u_{w^-} - u_0)\frac{y}{L}, & y \leq 0, \end{cases} \tag{11}$$

where u_0 is the velocity at $y = 0$

$$u_0 = \frac{1}{2}\left[(u_{w^+} + u_{w^-}) + \frac{fLh_{\text{unit}}}{\nu}\right]. \tag{12}$$

These solutions $u(y)$ are ramp functions with a corner at the delta force location.

Finite difference to solve the model

$$\frac{u_0^* - u_0}{\Delta t} = \frac{\nu}{h^2}(u_1^* + u_{-1}^* - 2u_0^*) \tag{13}$$

where u_0, u_1, and u_{-1} denote the velocity at $y = 0$, $y = h$, and $y = -h$, respectively, and the superscript * denotes the solution at the next time step.

$$\frac{u_0^{(n+1)} - u_0^{(n)}}{\Delta t} = \frac{\nu}{h^2}(u_1^{(n+1)} + u_{-1}^{(n+1)} - 2u_0^{(n+1)}) \tag{14}$$

The objective is to choose $u_0^{(n)}$ at previous time step so that at the next time step $u_0^{(n+1)} = u_0$. Therefor $u_0^{(n)}$ is required to be

$$u_0^{(n)} = -\frac{\nu\Delta t}{h^2}(u_1^{(n+1)} + u_{-1}^{(n+1)}) + \left(1 + \frac{2\nu\Delta t}{h^2}\right)u_0 \tag{15}$$

B Oscillatory 1D diffusion equation

Consider the 1D diffusion equation for velocity u with an oscillatory source term concentrated at the middle of the domain

$$\frac{\partial u}{\partial t} = \nu\frac{\mathrm{d}^2 u}{\mathrm{d}y^2} + \frac{C}{h_{\text{unit}}}e^{i\omega t}\delta(y), \quad -\infty < y < \infty \tag{16}$$

where C is a complex constant, i is the imaginary unit, and ω is the frequency. Make the ansatz

$$u = f(y)e^{i\omega t}. \tag{17}$$

Substitute the ansatz into the model equation (16) to get

$$i\omega f(y)e^{i\omega t} = \nu f''(y)e^{i\omega t} + \frac{C}{h_{\text{unit}}}e^{i\omega t}\delta(y) \tag{18}$$

For either $y > 0$ or $y < 0$, the above equation simplifies to

$$i\omega f(y)e^{i\omega t} = \nu f''(y)e^{i\omega t}. \tag{19}$$

Hence $f(y)$ satisfies the equation

$$f''(y) - \frac{i\omega}{\nu}f(y) = 0, \tag{20}$$

and its general solution is

$$f^+(y) = A^+e^{yk(1+i)} + B^+e^{-yk(1+i)}, \quad (y > 0), \tag{21a}$$

$$f^-(y) = A^-e^{yk(1+i)} + B^-e^{-yk(1+i)}, \quad (y < 0). \tag{21b}$$

where

$$k = \sqrt{\frac{\omega}{2\nu}}, \tag{22}$$

A^+, B^+, A^-, and B^- are constants to be determined. From the condition $u(\infty) = 0$, it is known $A^+ = 0$; from the condition $u(-\infty) = 0$, it is known $B^- = 0$. From the continuous condition $\lim_{y\to 0} f^+(y) = \lim_{y\to 0} f^-(y)$, we have $B^+ = A^-$. In order to determine B^+ and A^-, we integrate both sides of equation (18) with respect to y over the region $[-\epsilon, \epsilon]$, and let $\epsilon \to 0$, then

$$0 = \nu[\lim_{y\to 0} {f^+}'(y) - \lim_{y\to 0} {f^-}'(y)] + \frac{C}{h_{\text{unit}}}. \tag{23}$$

From equation (21), we have $\lim_{y\to 0} {f^+}'(y) = -k(1+i)B^+$, and $\lim_{y\to 0} {f^-}'(y) = k(1+i)A^-$. Therefore

$$0 = \nu[-k(1+i)B^+ - k(1+i)A^-] + \frac{C}{h_{\text{unit}}}. \tag{24}$$

Along with the condition $B^+ = A^-$, we have

$$B^+ = A^- = \frac{1}{h_{\text{unit}}} \frac{C}{2k(1+i)}. \tag{25}$$

Therefore, the velocity profile is known from equations (17), (21), and (25) as

$$u(y) = \begin{cases} \frac{1}{h_{\text{unit}}} \frac{C}{2k(1+i)} e^{-yk(1+i)+i\omega t}, & (y \geq 0), \\ \frac{1}{h_{\text{unit}}} \frac{C}{2k(1+i)} e^{yk(1+i)+i\omega t}, & (y \leq 0). \end{cases} \tag{26}$$

Specifically, if $C = 2k(1+i)u_0$ (with $h_{\text{unit}} = 1$), then

$$u(y) = \Re(u_0 e^{-yt+i(\omega t - yk)}) = u_0 e^{-yt} \cos(\omega t - yk), \quad (y \geq 0), \tag{27}$$

which is the solution for the second Stokes problem when a plate oscillates with the velocity $u_0 \cos(\omega t)$. It is worthy pointing out shear stress source term on the right hand side of equation (5) is twice of that on the oscillatory plate in the second Stokes problem, since equation (5) actually models two oscillatory plates aligned in a mirror configuration.

If the force f is known, this equation can be easily solved analytically as shown below. If the force f is unknown, but insteady the condition $u(0) = u_p$ is imposed, then this equation models the viscous channel flow along a thickless parallel plate locating in the middle, and the plate velocity is u_p.

References

[1] C. S. Peskin. The immersed boundary method. Acta Numerica **11** (2002) 479–517.

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

[3] B. E. Griffith and N. A. Patankar. Immersed methods for fluid-structure interaction. Annu. Rev. Fluid Mech. **52** (2020) 421–448.

[4] R. Verzicco. Immersed Boundary Methods: Historical Perspective and Future Outlook. Annu. Rev. Fluid Mech. **55** (2023-01) 129–155.

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

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

[7] J. Mohd-Yusof. Combined immersed boundary/B-spline methods for simulations of flow in complex geometry. techreport, CTR Annual Research Briefs, NASA Ames/Stanford University (1997).

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

[9] K. Zhou and S. Balachandar. An analysis of the spatio-temporal resolution of the immersed boundary method with direct forcing. Journal of Computational Physics **424** (2021) 109862.

[10] K. Zhou and T. L. Chan. Simulation of homogeneous particle nucleation in a free turbulent jet. Aerosol Sci. Technol. **45** (2011) 973–987.

[11] A. J. Chorin. Numerical solution of the Navier-Stokes equations. Math. Comput. **22** (1968) 745–762.

[12] M. M. Rai and P. Moin. Direct simulations of turbulent flow using finite-difference schemes. J. Comput. Phys. **96** (1991) 15–53.