# Immersed Boundary Method with Artificial Density in Pressure Equation for Modeling Flows Confined by Wall Boundaries

Xiaosong Sun and Mikio Sakai

Resilience Engineering Research Center, School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

**Keywords:** Computational Fluid Dynamics, Immersed Boundary, Pressure Poisson Equation, Complex Geometry, Microchannel

The immersed boundary (IB) method is a promising technique for the numerical simulation of complex microchannel flows. However, numerical tests revealed a loss of wall impermeability for certain flow cases leading to unphysical results when using a standard IB method. To solve this problem, we propose a novel artificial density technique incorporated in the pressure solution for IB methods. The pressure equation is modified by scaled density coefficients in IB domains, which reflects the effect the boundary screening. The proposed model is as simple and efficient as the original IB method, and it can eliminate the spurious velocity penetration. Numerical results are shown to justify this method. Its application to a branched microchannel is also presented in the last part of this study.

#### Introduction

The immersed boundary (IB) method (Peskin, 1977) is a versatile numerical technique to model flows interacting with wall geometries and moving boundaries that are difficult to treat in the context of conventional computational fluid dynamics (CFD). A typical IB method utilizes Cartesian grids that are not conforming to the wall geometry, and adopts so-called forcing terms to modify the flow field near the boundary. A widely used model is the direct-forcing IB method (Fadlun et al., 2000). Another variant of this type is that proposed by Kajishima et al. (2001). Both approaches have been applied to complicated fluid boundaries in chemical engineering problems including, e.g., porous membrane (Ishigami et al., 2013), tubes in flow channel (Dhar et al., 2014), resistance correlations of sphere arrays (Kuwagi et al., 2012) and woven meshes (Yoshida et al., 2015), and mobile solid particles (Deen et al., 2012).

Specifically, our interest of research is (but not limited to) using the IB method to simulate complex flows in microchannels. Compared with existing methods, the IB method alleviates the mesh generation work for preprocessing and retains a good computational efficiency at runtime. Moreover, in recent studies, we have successfully extended the scope of the IB method to the simulation of two-phase (Sun and Sakai, 2016) and three-phase (Sun and Sakai, 2015) flows, which is a good place of departure for the practical numerical analysis of complex flows involving droplets, bubbles and particles in microchannels.

The present study is motivated by our recent effort to

Received on April 19, 2016; accepted on September 25, 2016 DOI: 10.1252/jcej.16we115

Correspondence concerning this article should be addressed to X. Sun (E-mail address: song@dem.t.u-tokyo.ac.jp).

simulate microchannels with side branches. We have noticed a critical problem in that sometimes flow velocity may unphysically penetrate the channel walls modelled by the IB. This spurious flow transpires between fluid passages that should be separated by rigid boundaries, and subsequently contaminates the overall results. In other words, it is suspected that the impermeability of fluid boundaries can not be satisfied by the existing IB methodology in certain situations. To our best knowledge, this problem has not been discussed in the literature. This is possibly because most IB studies focus on external flows around rigid bodies rather than confined flows.

Basically, our numerical study revealed that this problem is linked to the solution of pressure. In a standard IB method, the pseudo-fluid inside the IB domain is assumed to have the same density as true fluids. Then, the fluid pressure is solved in the whole computational domain. This procedure neglects the well-known zero-flux condition across walls, which is responsible for the loss of boundary impermeability in peculiar flow cases.

A possible remedy is to use the sharp-interface IB described in Seo and Mittal (2011). However, it mostly depends on the cut-cell technique which is highly complicated and comes with restrictions on time step. It also complicates the task to solve pressure since the cut-cell mesh is no longer an orthogonal Cartesian grid.

Therefore, the present objective is to improve the boundary modelling while keeping the simplicity and efficiency of the IB method. In this study, we propose a new technique to address this problem, in which the pressure equation is modified by artificial density in IB domains. With the new treatment, the flow fields are computed accurately and spurious velocity penetration can be suppressed. Moreover, it retains the Cartesian mesh and a symmetric matrix, for which efficient solvers on structured grids can be used with-

out modifications.

#### 1. Basic Fluid Solver

For incompressible Newtonian fluids, the Navier-Stokes equation and continuity equation are written as follows.

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}$$
 (1)

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2}$$

Here, u, p,  $\rho$  and v are the velocity, pressure, density and kinetic viscosity, respectively. On walls, the boundary conditions (BCs) are assumed to be  $u = u_B$  and  $\nabla_B p = 0$ . Herein, subscript B denotes boundary values.

A projection method is employed to solve the fluid equations. At the n-th step, an intermediate velocity  $u^*$  is calculated with only the convective and viscous terms.

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -\nabla \cdot (\boldsymbol{u}\boldsymbol{u})^n + \nu \nabla^2 \boldsymbol{u}^n \tag{3}$$

Next, the pressure *p* is solved from the pressure Poisson equation (PPE).

$$\nabla \cdot \frac{1}{\rho} \nabla p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{u}^* \tag{4}$$

Finally, the intermediate velocity is corrected to satisfy the divergence-free condition.

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\frac{1}{\rho} \nabla p^{n+1} \tag{5}$$

Governing equations are discretized on a staggered grid where velocity components are defined on cell faces and other variables such as the pressure are defined at cell centers. The numerical solver is based on the finite-volume method for which spatial derivatives are calculated by second-order central difference schemes. Details can be found in Sun and Sakai (2015).

# 2. Immersed Boundary Method

The fluid solver described in the preceding section is supplemented with the IB technique to model solid walls immersed in fluids. The specific treatments for fluid velocity and pressure are discussed in this section.

# 2.1 Representation by signed distance function

To facilitate the modelling of complex geometries, their shapes are represented by the signed distance function (SDF) approach (Shigeto and Sakai, 2013). A scalar function  $\psi(x)$  indicates the distance from point x to the nearest point on the boundary. By definition, "fluid" point has  $\psi > 0$ , "solid" point has  $\psi \le 0$ , and the zero-contour of the SDF field  $\psi$  indicates the boundary shape. On a standard staggered grid,  $\psi$  is kept at cell centers as pressure p. Hence, by using  $\psi$ , one can determine the immersed wall cells immediately and obtain the local boundary profile with ease (see (Sun

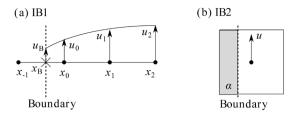


Fig. 1 Two types of IB forcing methods

and Sakai, 2016)).

#### 2.2 Velocity forcing in IB method

The present study relies on the direct-forcing IB methodology which introduces a special forcing term near the solid surface to prescribe desired velocity boundary conditions. Two forcing methods are considered in this study. The first method (IB1) is based on the geometrical interpolation of Fadlun *et al.* (2000), which has been widely used in the IB community. The idea is to determine the fluid velocity nearest to the boundary by interpolation of the local velocity profile. Instead of the original linear interpolation, we use quadratic interpolation in this study. For the point  $x_0$  next to the boundary in **Figure 1(a)**, it can thus be written as Eq. (6).

$$u_0 = c_1 u_1 + c_2 u_2 + c_B u_B \tag{6}$$

Here, u is the velocity component, c is coefficients determined by the Lagrange formula given in Appendix 1, and subscript B indicates the point on the boundary.

The second method (IB2) is the volume-fraction interpolation of Kajishima *et al.* (2001). In this approach, velocity in the vicinity of the boundary is modified by a volume-averaging of local fluid velocity and solid velocity. **Figure 1(b)** shows a control volume intersecting the boundary for which the velocity becomes as expressed by Eq. (7).

$$u = (1 - \alpha)u_f + \alpha u_s \tag{7}$$

Here,  $\alpha$  is the solid volume fraction within the control volume and subscripts f and s denote fluid phase and solid phase, respectively.

#### 2.3 Pressure solution in standard IB

After applying the previous velocity forcing to the intermediate velocity  $u^*$ , the pressure p is obtained by solving the PPE (4). On the staggered grid, the discretized PPE leads to a 5-point stencil in 2D (and 7-point in 3D). For clarity, the discretization stencil is depicted in **Figure 2**. For the center point (i, j), the left-hand side of the PPE (4) is computed by Eq. (8).

$$\nabla \cdot \frac{1}{\rho} \nabla p_{ij} = \frac{1}{\Delta x^2} \left( \frac{p_{i+1,j} - p_{i,j}}{\rho_{i+1/2,j}} + \frac{p_{i-1,j} - p_{i,j}}{\rho_{i-1/2,j}} \right) + \frac{1}{\Delta y^2} \left( \frac{p_{i,j+1} - p_{i,j}}{\rho_{i,j+1/2}} + \frac{p_{i,j-1} - p_{i,j}}{\rho_{i,j-1/2}} \right)$$
(8)

Alternatively, it is written in algebraic equations of equiva-

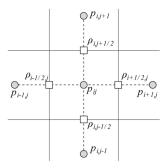


Fig. 2 The 5-point pressure stencil in 2D

lent form as follows.

$$\nabla \cdot \frac{1}{\rho} \nabla p_{ij} = A_{W} p_{i-1,j} + A_{E} p_{i+1,j} + A_{S} p_{i,j-1} + A_{N} p_{i,j+1} + A_{P} p_{ij}$$

$$A_{W} = (\Delta x^{2} \rho_{i-1/2,j})^{-1}, \quad A_{E} = (\Delta x^{2} \rho_{i+1/2,j})^{-1}$$

$$A_{S} = (\Delta y^{2} \rho_{i,j-1/2})^{-1}, \quad A_{N} = (\Delta y^{2} \rho_{i,j+1/2})^{-1}$$

$$A_{P} = -(A_{W} + A_{E} + A_{S} + A_{N})$$

It involves pressure in adjacent neighbor cells and density defined at staggered face positions. In the standard IB method, those density components are set to  $\rho$  everywhere, and hence the differential operator is reduced to a Laplace operator. Then, the linear equations are solved in *the entire computational domain* including both fluid and solid parts. This simple approach is efficient and reasonably accurate in many flow cases (see Guy and Hartenstine (2010)).

## 2.4 Improved pressure treatment

Despite the success in many past studies, recently we have noticed that spurious velocity penetration may transpire through the IB separating individual fluid regions. In that case, the IB method fails to model impermeable fluid boundaries. See Section 4 for an example.

This problem is ascribed to the pressure solution in the standard IB method. Although the Dirichlet boundary condition for velocity is satisfied via the forcing term, the Neumann boundary condition for pressure is not treated properly. In fact, the aforementioned numerical procedure does not require any knowledge about the boundary.

We propose an improved pressure treatment by replacing the physical density  $\rho$  with an artificial density  $\hat{\rho}$  in Eqs. (4) and (5). Hence, the modified equations become Eqs. (9) and (10).

$$\nabla \cdot \frac{1}{\hat{\rho}} \nabla p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{u}^* \tag{9}$$

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^*}{\Delta t} = -\frac{1}{\hat{\rho}} \nabla p^{n+1} \tag{10}$$

The discrete form thereof is that given by Eq. (8). Herein, the artificial density  $\hat{\rho}$  takes a scaled value of  $\rho$  if any adjacent cell belongs to the solid (recall that fluid and solid cells are distinguished by the SDF  $\psi$  defined at cell centers). For example, the density at the *x*-staggered position (i+1/2, j) is

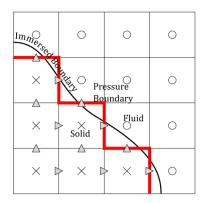


Fig. 3 IB method with artificial density. Fluid and solid cells are shown by circles and crosses. Face points affected by the IB (curved line) are tagged by filled triangles, that is, they lie on solid-solid or solid-fluid faces. They forms an approximate pressure boundary as the red line

given by Eq. (11).

$$\hat{\rho}_{i+1/2,j} = \begin{cases} \gamma \rho & \text{if } \psi_{ij} \le 0 \text{ or } \psi_{i+1,j} \le 0, \\ \rho & \text{otherwise} \end{cases}$$
 (11)

Here,  $\gamma \ge 1$  is a scaling factor. The same judgment also applies to any y-staggered position (i, j+1/2). The current strategy is in fact adapted to the usage of a staggered mesh on which the pressure is defined at cell centers. Figure 3 illustrates schematically one possible configuration when this algorithm is applied to the IB on a staggered grid. The x- and y-face points identified by the procedure of Eq. (11) are tagged with solid triangles to be associated with the artificial density. They lie at either solid-solid faces or solidfluid faces, while the fluid-fluid faces are not affected at all. Therefore, it is natural to determine boundary face points by using the SDF from adjacent cells. Similar techniques have been used by Sussman et al. (2007) to locate a sharp gas-liquid interface. An alternative method may consist in checking the SDF at face points directly, which however involves more interpolation operations than the current method.

To understand the role of  $\hat{\rho}$ , one may consider placing a "heavier" fluid inside the IB so that the local momentum flux is penalized. The extreme case of  $\gamma \to \infty$  is equal to decouple all solid points from the PPE and prescribe homogeneous Neumann pressure BC on fluid-solid faces (i.e., the bold red line\* in Figure 3). Alternatively, the modified PPE (9) can be equivalently rewritten as  $\nabla \cdot (\varepsilon \rho^{-1} \nabla p) = \text{RHS}$  with  $\varepsilon = \rho/\hat{\rho}$ . This form is similar to that found in porous media (Sun and Sakai, 2015) and the ratio  $\varepsilon$  as the porosity that is unity in fluid and  $\varepsilon = \gamma^{-1} \le 1$  in solid. We use  $\gamma = 10^3$  in this study if not specified. The effect of this parameter will be explored in Section 4.1.

It is evident that the coefficient matrix resulting from

<sup>\*</sup>It is important to note that, the approximated pressure boundary in Figure 3 does not necessarily mean that geometries are modelled in a stair-shaped fashion. Owing to the IB forcing, smooth boundary shapes are readily taken into account for velocity calculation.

the modified PPE (9) is still preferably symmetric. On the other hand, unlike the original matrix, it has coefficients varying across the IB. Such a variable-coefficient matrix has typically larger condition numbers compared with its constant-coefficient counterpart. Thus, it generally requires more computational effort to solve the modified PPE. Plain iterative solvers (e.g., the conjugate gradient (CG) method) may suffer from degraded performance in this case. Based on our experiences, the multigrid preconditioned conjugate gradient (MGPCG) method is usually the optimal choice so it is employed in this study. In the later Section 4.2, we show that the modified PPE can be treated by using an existing MGPCG solver as the original equation, provided that the parameter  $\gamma$  is chosen carefully.

#### 3. Numerical Tests of Poiseuille Flows

The Poiseuille viscous flow developed in channels is fundamental to microfluidic engineering. In this section, we use the IB to model channel boundaries and examine the accuracy and convergence of the current method. This serves as validation tests for our model.

#### 3.1 Poiseuille flows in different channels

We compute first two cases of (a) 2D flow between plates and (b) 3D flow in a circular tube where the flow direction is parallel with coordinate axis, and the third case (c) 2D flow in an inclined channel taking an angle  $\Theta = 30^{\circ}$  relative to the axis. The steady solution of stream velocity u is analytically given by a parabolic profile as Eq. (12).

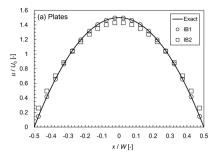
$$u = CU(1 - 4x^2/W^2) \tag{12}$$

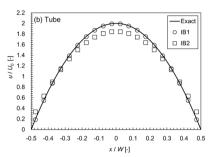
This is measured in spanwise position x from the center line of the channel, in which U is the mean velocity and W is the channel width. C is a problem-dependent constant: C = 1.5 for 2D plates and C = 2 for 3D tube.

In the simulations, we set a uniform velocity  $U_0$  at the inlet side and freestream condition at the outlet side. The Reynolds number based on channel width W, flow speed  $U_0$  and fluid kinematic viscosity v is approximately  $Re = U_0W/v \sim 1$ , which is sufficiently low to keep the channel flow laminar. The channel is also long enough for the flow to fully develop (typically we have the channel length L>8W). The cell size is given in terms of the channel width as  $\Delta x = W/20$ . Specifically, in the planar case (a), the parallel walls are exactly aligned with the cell faces separating fluid and solid cells (c.f. Appendix 2).

We compare the computed velocity profiles with the analytical solution. Figures 4(a) and (b) plot the results of 2D plates and 3D tube, respectively. Solutions obtained by the two different IB forcing methods (IB1 and IB2) are juxtaposed for comparison. In both cases (a) and (b), the IB1 results can recover the parabolic profile, but IB2 tends to slip near the channel wall and lag in the center.

Their difference stems from the treatment of boundary condition for fluid velocity near the wall. In channel flows, local boundary condition is actually required by the first





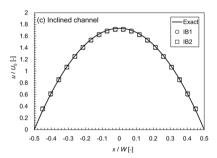


Fig. 4 Velocity profile of Poiseuille flow

layer of fluid points next to the IB for the velocity (mainly dominated by the viscosity) to be accurately calculated. In this respect, the IB1 approach makes use of information about both the exact boundary position and velocity, and it determines the near-wall velocity by interpolation formula rather than solving the governing equations. In IB2, however, the near-wall velocity is first computed from the fluid solver ignoring the IB, and then corrected by a volume-of-solid manner in a later stage. Especially, since in cases (a) and (b) the cell boundaries are aligned with the wall, in IB2 there are some velocity points that lie next to the wall but have vanishing solid fractions  $\alpha = 0$ , for which the IB2 forcing term thereby becomes zero. As a result, it turns out that those points cannot benefit from the velocity correction of IB2, which consequently causes the finite slip of near-wall velocity

**Figure 4(c)** plots the velocity profile of the inclined channel. It is important that the IB method can give accurate predictions for such non-trivial geometries. The results agree well with the analytical solution (note that the mean velocity in this case is  $U = U_0/\cos\Theta \approx 1.15 U_0$ ). Incidentally, the difference between IB1 and IB2 is minor in this case. It is possibly

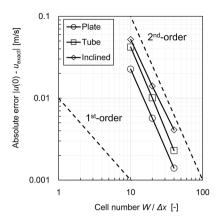


Fig. 5 Error convergence of Poiseuille flow

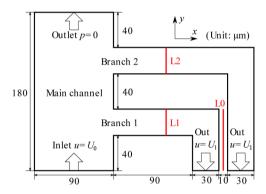


Fig. 6 Schematic diagram of the microchannel

because, unlike (a) and (b), in case (c) the wall is not parallel to the cell faces, so there is almost no vanishing  $\alpha$  for any velocity point next to the wall. Thus, the IB2 forcing can work properly again. It is noted that IB2 is not based on geometrical interpolation, so its accuracy might be more prone to the influence of grid configuration. In Appendix 2, we provide extra supporting data for this problem.

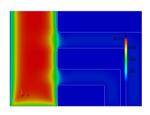
Based on numerical tests in this section, it seems that the IB1 approach yields better accuracy than IB2 for channel flows in various geometries and configurations. Therefore, numerical studies will be restricted to those using IB1 hereinafter.

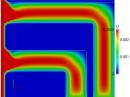
#### 3.2 Grid-convergence study

We have also performed mesh-convergence studies for channel flow tests to verify the numerical implementation. We evaluate the absolute error in velocity magnitude at the channel center position between the numerical result and the analytical solution. This measurement is similar to the error analysis for IB by Peller *et al.* (2006). The numerical errors from all three channel geometries are summarized in **Figure 5** against the grid resolution. Slopes corresponding to first-order and second-order convergence rates are also drawn as a guide to the eye. When refining the grid, we confirmed a second-order convergence behavior, which is expected for the present IB method in optimal conditions.

Table 1 Computational conditions for the microchannel

Parameter	Value	Unit
Domain size	250×180×26	[µm]
Grid number	$125\times90\times13$	[—]
Cell size $\Delta x$	$2 \times 10^{-6}$	[m]
Time step $\Delta t$	$2 \times 10^{-8}$	[s]
Density	1.0	$[kg/m^3]$
Kinetic viscosity	$1.8 \times 10^{-5}$	$[m^2/s]$





- (a) Overall velocity contour.
- (b) Colorbar rescaled for two side channels.

**Fig.** 7 Representative snapshot of the steady state ( $\gamma = 10^3$ )

**Table 2** Simulation cases with different  $\gamma$  values

	Case 1	Case 2	Case 3	Case 4	Case 5
γ	$10^{0}$	$10^{1}$	$10^{2}$	$10^{3}$	$10^{4}$

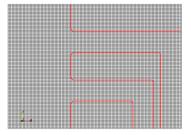
## 4. Application to Microchannel Simulation

In this section, the comparison between the artificial density approach and standard IB will be shown within a truncated microchannel. The problem setup is schematically illustrated in Figure 6. It has a main channel with velocity inlet  $u = U_0$  and pressure outlet p = 0 at the two ends, respectively. Two branches are connected with the bulk flow, and constant-velocity drainage  $u = U_1$  is applied at their end. We set  $U_0 = 0.0427 \,\text{m/s}$  and  $U_1 = -0.00128 \,\text{m/s}$ . The fluid is assumed to have density  $\rho = 1 \text{ kg/m}^3$  and kinematic viscosity  $v = 1.8 \times 10^{-5} \,\mathrm{m}^2/\mathrm{s}$ . The Reynolds number is  $Re \approx 0.21$ based on  $U_0$ ,  $\nu$  and main channel width. The simulations are conducted in a  $250 \times 180 \times 26 \,\mu m$  box bounding the channel. Grid spacing is  $\Delta x = 2 \,\mu\text{m}$ . The flow state is evolved with a time step of  $\Delta t = 2 \times 10^{-8}$  s until a steady state is obtained. These computational conditions are summarized in Table 1. A representative snapshot of the steady flow is given in Fig**ure 7**, for which we use  $\gamma = 10^3$ .

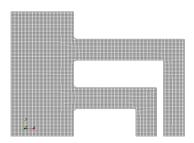
#### 4.1 Effect of the artificial density

The factor y of artificial density is varied from y = 1 to  $10^4$  in order to investigate its effect on the numerical results. The five cases being compared are listed in **Table 2**. We note that the computation falls back to the standard IB approach for Case 1 (y = 1).

A reference solution is also calculated by the open-source CFD software OpenFOAM®. By comparing the IB result and the OpenFOAM solution (referred to as "OF" hereinafter), it is intended to critically examine the consis-



(a) IB method



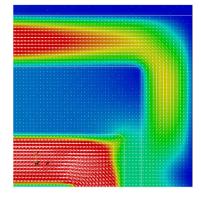
(b) OpenFOAM

**Fig. 8** Computational meshes used in the microchannel simulation; rendered by grid lines on the *x*–*y* slice; (a) Cartesian grid of Immersed boundary method, where the red line states for the channel wall; (b) Hexahedral grid of OpenFOAM

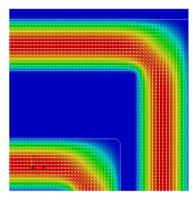
tency of the IB-based wall modelling with conventional approaches where the wall boundary condition is directly prescribed. Specifically, we use the simpleFoam, which is a finite-volume solver on unstructured grids. For its usage, body-fitted hexahedral meshes are generated with an average mesh size of  $2.0\,\mu m$ . The same boundary conditions are used as those of the IB simulation. We setup the solver using second-order spatial discretization schemes, and obtain the steady-state solution by monitoring the convergence of velocity and pressure fields. The computational grids used by the IB simulation and the OpenFOAM are shown in **Figure 8**.

The differences between Case 1 and other cases become significant if we focus on the flow in the two side channels. Images in **Figure 9** show closed-up details for the velocity solutions near the corner. In Case 1 ( $\gamma$ =1), there is spurious velocity build-up in the IB region between two side channels as depicted by Figure 9(a). This velocity unphysically penetrates the wall and connects the two distinct fluid regions which results in abnormal flow patterns. On the other hand, the computed results in Figure 9(b) by Case 4 ( $\gamma$ =10<sup>3</sup>) seem to be more reasonable in that the two channel flows do not interfere with each other. A proof of its validity is also supported by the qualitative agreement in flow velocity with the OpenFOAM solutions in Figure 9(c).

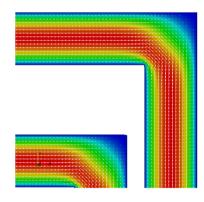
The velocity magnitude along the gauge line  $L_0$  (see Figure 6) is plotted for different  $\gamma$  values in **Figure 10(a)**. Since  $L_0$  lies in the IB region, zero velocity is expected therein.



(a) Case 1 ( $\gamma = 1$ )



(b) Case 4 ( $\gamma = 10^3$ )

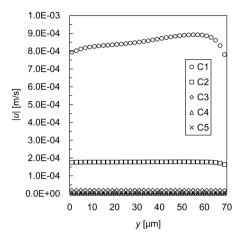


(c) OpenFOAM

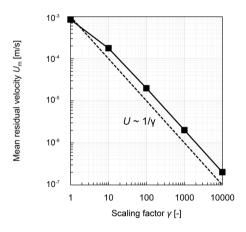
Fig. 9 Close-up for flow in side channels; the arrows are velocity vectors and background color is velocity contour

Apparently, this spurious velocity vanishes as the value of  $\gamma$  increases. Finite spurious modes remain for Cases 1–3, but they become very close to zero for Cases 4 and 5. **Figure 10(b)** shows the relationship between the mean velocity and  $\gamma$  value used. The reduction of spurious velocity is found to be inversely proportional with  $\gamma$ , as implied by the dashed line of  $O(\gamma^{-1})$ . This observation is consistent with one's physical intuition because the mobility in the IB domain is now controlled by the reciprocal of artificial density.

For purpose of further validation, we compare the results



(a) Velocity profile along  $L_0$ 



(b) Convergence of mean residual velocity

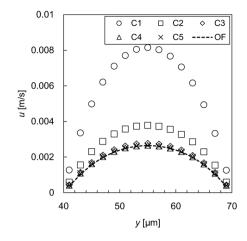
Fig. 10 Reduction of spurious velocity on  $L_0$ 

of horizontal velocity component on gauge lines  $L_1$  and  $L_2$  (x=120, see Figure 6) from different cases together with the OpenFOAM solution. At low  $\gamma$  values, the velocity profiles in **Figure 11** significantly deviate from the reference solution. This problem is especially severe for Case 1 (namely the standard IB method) in that the flow may even move in the reverse direction (Figure 11(b)). On the other hand, the results from Cases 4 and 5 are generally in good agreement with the reference solution. Therefore, the artificial density approach is shown to be effective provided that the factor  $\gamma$  is sufficiently large (typically  $\gamma \ge 10^3$  in our tests).

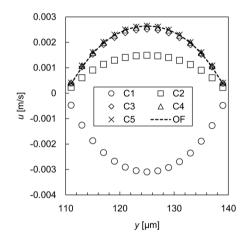
### 4.2 Impact on computational cost

In this section, we investigate the extra computation cost that comes with the artificial density approach. All simulations run on a Windows PC equipped with Intel Core i7 CPU and 16 GB of RAM. The program is parallelized by the OpenMP technique using twelve threads. The convergence criterion for the MGPCG solver is set to  $\varepsilon = 10^{-8}$ .

We collect performance data among 2,000 steps after



(a) Velocity on  $L_1$ 



(b) Velocity on  $L_2$ 

Fig. 11 Comparison of flow velocity in side channels

the flow has reached a steady state, then evaluate the average computational time for each single simulation step. We also check the iteration number for the pressure solver to converge. They are given in Figure 12. Clearly a large factor  $\gamma$  tends to increase the computational cost. This is not surprising, as mentioned previously, that the discontinuous coefficients in the modified pressure equation will add to the condition number of the system matrix. Nevertheless, the additional cost is acceptable considering the simplicity of the present method, besides the improved modelling of immersed boundary. For example, compared with Case 1  $(\gamma = 1)$ , Case 4  $(\gamma = 10^3)$  requires a moderate increase of only 6 more solver cycles and 1.48 times the CPU time. Based on the tests in the preceding and current sections, an artificial scaling factor of  $y = 10^3$  is considered to be a good compromise between accuracy and efficiency.

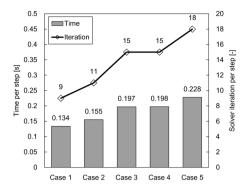


Fig. 12 Computational cost of different cases

#### **Conclusions**

In this study, we analyzed the spurious velocity penetration in the standard IB method and developed a new approach to tackle this problem. The proposed method uses an artificial density in the IB domains so that the boundary resistance can be taken into account. Our methodology is straightforward and requires only moderate modifications to existing IB methods.

This method has been first validated for Poiseuille flows in various channel geometries. We compare results by using different IB forcing terms, and find that a combination of direct-forcing and artificial-density approaches is suitable for resolving flow cases in channels.

The proposed method is then applied to an actual microchannel with multiple branches. In this test, it is demonstrated that standard IB yields erroneous velocity fields; whereas, the new approach alleviates this problem. The computational cost of the present method is also examined. Based on a compromise between accuracy and efficiency, an optimal parameter is suggested for the artificial density model. In the near future, we are planning to simulate fluid-particle interactions (e.g., particle separations) in complex microchannels by using the proposed method.

## Acknowledgement

This study was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 13J09948. KOZO KEIKAKU ENGINEERING Inc. is also gratefully acknowledged for financially supporting this study. We thank two referees for their constructive advice for improving the quality of the manuscript and inspiring us to focus on the grid dependency problems in the current method.

#### **Appendix**

# 1. Quadratic interpolation scheme

Consider the 1D stencil in Figure 1(a), we have the following weights for the quadratic Lagrangian interpolation (6).

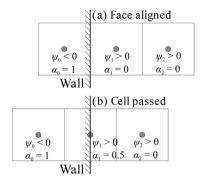


Fig. A1 Grid configurations of Poiseuille flow

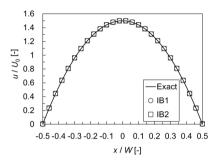


Fig. A2 Velocity profile of Poiseuille flow

$$c_1 = \frac{x_0 - x_B}{x_1 - x_B} \frac{x_0 - x_2}{x_1 - x_2}, \quad c_2 = \frac{x_0 - x_B}{x_2 - x_B} \frac{x_0 - x_1}{x_2 - x_1},$$

$$c_B = \frac{x_0 - x_1}{x_B - x_1} \frac{x_0 - x_2}{x_B - x_2}$$

Here, *x* denotes the point coordinate. In actual 3D case, we compute the 1D interpolations distinctly in each spatial direction if applicable, and take a simple average of those interpolated results.

#### 2. Grid dependency of IB2 scheme

As shown in Section 3.1, the IB2 scheme blending fluid and solid velocities can work well for an inclined channel rather than straight channels. It is argued that the dependency of IB2 forcing on the local grid/IB configuration is responsible for this problem. That is, despite the simplicity of the wall geometry, a finite value of solid volume fraction  $\alpha$  along the boundary is necessary for IB2 to be activated.

Specifically, we consider two configurations for a vertical wall embedded in uniform grids in **Figure A1**: (a) the cell face is aligned with the wall, and (b) the cell center is passed by the wall. Figure A1(a) corresponds to the numerical test of Section 3.1, in which the first fluid cell next to the IB (Cell 1) has  $\alpha_1$ =0 leading to obvious slip near the wall (see Figure 4(a)). On the other hand, the wall in Figure A1(b) runs through the center of Cell 1, giving  $\alpha_1$ =0.5, from which a better result can be expected.

**Figure A2** shows the velocity profile computed by the new grid configuration. Indeed, the results of IB1 (based on

velocity interpolation) and IB2 (based on volume-fraction interpolation) are almost the same, and they both agree with the analytical solution. As a matter of course, the IB2 forcing still introduces some slip near the wall. For Cell 1 on the wall (Figure A1(b)), the IB1 scheme gives an essentially zero velocity ( $<10^{-14}$  m/s), while IB2 has a small but nonzero velocity of  $4.69\times10^{-3}$  m/s at steady state.

Therefore, it is demonstrated that there might exist some room to improve the IB2 results by carefully adjusting the grid configuration, which is, however, not likely to be achieved for general problems. Some authors (e.g., Lin *et al.* 2011) using similar approaches may benefit from substantially collocating numerical cells on the immersed boundaries. To our knowledge, a systematic investigation of the grid dependency for IB2-type methods has not been carried out in past studies.

#### Nomenclature

= velocity

p	=	pressure	[Pa]
α	=	solid volume-fraction	[—]
γ	=	factor of artificial density	[—]
ν	=	kinetic viscosity	$[m^2/s]$
ρ	=	density	[kg/m³]
$\psi$	=	signed distance function	[m]

#### (Subscripts)

B = boundary value f = fluid phase s = solid phase

## **Literature Cited**

- Deen, N. G., S. H. Kriebitzsch, M. A. van der Hoef and J. Kuipers; "Direct Numerical Simulation of Flow and Heat Transfer in Dense Fluid-Particle Systems," *Chem. Eng. Sci.*, **81**, 329–344 (2012)
- Dhar, A., N. Shimada, K. Hayashi and A. Tomiyama; "An Interface-Capturing Method for Free-Surface Flows in a Flow Channel Consisting of Solid Obstacles," *J. Chem. Eng. Japan*, 47, 230–240 (2014)
- Fadlun, E., R. Verzicco, P. Orlandi and J. Mohd-Yusof; "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations," J. Comput. Phys., 161, 35–60

(2000)

[m/s]

- Guy, R. D. and D. A. Hartenstine; "On the Accuracy of Direct Forcing Immersed Boundary Methods with Projection Methods," J. Comput. Phys., 229, 2479–2496 (2010)
- Ishigami, T., H. Fuse, S. Asao, D. Saeki, Y. Ohmukai, E. Kamio and H. Matsuyama; "Permeation of Dispersed Particles through a Pore and Transmembrane Pressure Behavior in Dead-End Constant-Flux Microfiltration by Two-Dimensional Direct Numerical Simulation," *Ind. Eng. Chem. Res.*, 52, 4650–4659 (2013)
- Kajishima, T., S. Takiguchi, H. Hamasaki and Y. Miyake; "Turbulence Structure of Particle-Laden Flow in a Vertical Plane Channel Due to Vortex Shedding," JSME Int. J. Ser. B Fluids Therm. Eng., 44, 526–535 (2001)
- Kuwagi, K., Y. Shimoyama, A. Bin Alias, H. Hirano and T. Takami; "Examination of Various Estimation Equations for Drag Force by Using Immersed Boundary Method," J. Chem. Eng. Japan, 45, 107–113 (2012)
- Lin, S.-Y., Y.-H. Chin, J.-J. Hu and Y.-C. Chen; "A Pressure Correction Method for Fluid-Particle Interaction Flow: Direct-Forcing Method and Sedimentation Flow," *Int. J. Numer. Methods Fluids*, 67, 1771–1798 (2011)
- Peller, N., A. L. Duc, F. Tremblay and M. Manhart; "High-Order Stable Interpolations for Immersed Boundary Methods," Int. J. Numer. Methods Fluids, 52, 1175–1193 (2006)
- Peskin, C. S.; "Numerical Analysis of Blood Flow in the Heart," *J. Comput. Phys.*, **25**, 220–252 (1977)
- Seo, J. H. and R. Mittal; "A Sharp-Interface Immersed Boundary Method with Improved Mass Conservation and Reduced Spurious Pressure Oscillations," J. Comput. Phys., 230, 7347–7363 (2011)
- Shigeto, Y. and M. Sakai; "Arbitrary-Shaped Wall Boundary Modeling Based on Signed Distance Functions for Granular Flow Simulations," *Chem. Eng. J.*, **231**, 464–476 (2013)
- Sun, X. and M. Sakai; "Three-Dimensional Simulation of Gas-Solid-Liquid Flows Using the DEM-VOF Method," Chem. Eng. Sci., 134, 531–548 (2015)
- Sun, X. and M. Sakai; "Numerical Simulation of Two-Phase Flows in Complex Geometries by Using the Volume-of-Fluid/Immersed-Boundary Method," *Chem. Eng. Sci.*, **139**, 221–240 (2016)
- Sussman, M., K. Smith, M. Hussaini, M. Ohta and R. Zhi-Wei; "A Sharp Interface Method for Incompressible Two-Phase Flows," J. Comput. Phys., 221, 469–505 (2007)
- Yoshida, Y., Y. Inoue, A. Shimosaka, Y. Shirakawa and J. Hidaka; "Numerical Simulation of Flow Resistivity of Metal Woven Mesh," *J. Chem. Eng. Japan*, **48**, 545–555 (2015)