

A sharp interface immersed boundary method for solving flow with arbitrarily irregular and changing geometry

Huangrui Mo^{*1}, Fue-Sang Lien¹, Fan Zhang², and Duane S. Cronin¹

¹*Department of Mechanical Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada*

²*Defence R&D Canada – Suffield, P.O. Box 4000, Medicine Hat, AB T1A 8K6, Canada*

February 23, 2016

Abstract

In this paper, a sharp interface immersed boundary method is developed for efficiently and robustly solving flow with arbitrarily irregular and changing geometries. The proposed method employs a three-step prediction-correction flow reconstruction scheme for boundary treatment and enforces Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner. Numerical experiments concerning flow of two and three space dimensions, stationary and moving objects, convex and concave geometries, no-slip and slip wall boundary conditions are conducted to demonstrate the proposed method.

1 Introduction

Particle jets, a rapid and nonuniform dispersal of granular media under impulsive energy release, are observed in many physical processes such as explosions with explosive charges surrounded by or mixed with solid particles [1, 2], explosive volcanic eruptions [3], and impact of a solid projectile on granular media [4, 5].

In order to understand this jet phenomenon, many studies have been conducted over the past decades [6]. However, the formation mechanism of particle jets is still unidentified [7], which is primarily due to the complexity of shock-particle interactions [7] and random nature of the force-chain networks in granular materials [8, 9].

^{*}Email: huangrui.mo@uwaterloo.ca; Corresponding author

To develop a numerical solver for fully investigating the particle jet phenomenon, a computational fluid dynamics solver for simulating shock-particle interactions under strong shocks and complex particle configurations is a prerequisite, of which the capability of dealing efficiently and robustly with arbitrarily irregular and changing geometries is crucial.

In the immersed boundary method introduced by Peskin [10], numerically solving flow with irregular geometries is conducted on generic Cartesian grids, which can greatly simplify the grid generation process and take advantages of some important features of modern high-performance computing architecture [11–13]. For instance, reduced memory requirements and the capability to use linear arrays with linear indexing techniques as main data structures can benefit from modern processor prefetcher and hierachal-cache architecture to achieve a highly efficient numerical solver.

Extensions of the immersed boundary method have been continuously developed to increase interface resolution and relax stability constraints [14–20]. Mohd-Yusof [21] and Fadlun et al. [14] developed the direct forcing immersed boundary method. In the direct forcing approach, boundary forces are implicitly imposed via flow reconstruction, which simplifies the numerical discretization procedure considerably and is well-suited for problems with rigid boundaries. Balaras [22] later improved the reconstruction procedure of direct forcing and applied to large-eddy simulations. Integrating ideas from the ghost fluid method [23, 24] and the direct forcing immersed boundary treatment [14, 25], Tseng and Ferziger [15] systematically developed a polynomial reconstruction based ghost-cell immersed boundary method to further increase implementation flexibility while maintaining sharp interfaces [13, 15]. Kapahi et al. [18] proposed a least square interpolation approach and applied to solving high velocity impact problems. The development of immersed boundary methods was comprehensively reviewed by Peskin [12], Mittal and Iaccarino [13], and Sotiropoulos and Yang [26].

The robustness of a direct forcing immersed boundary method highly depends on the numerical stability and stencil adaption capability of the employed interpolation method [15, 18, 27]. Polynomial reconstruction based methods frequently involve constructing linear systems on neighbouring stencils, including a nearby boundary point, of the interpolated node. When one of the stencils is very close to the boundary point, the resulting linear systems may suffer from numerical singularities [15, 27]. Additionally, a fixed minimum number of stencils is always required to avoid under-determined linear systems. Therefore, special treatments are required when strongly concave or convex geometries exist [18, 27]. To enhance numerical stability and stencil adaption capability, the idea of using inverse distance weighting interpolation is firstly introduced by Tseng and Ferziger [15]; and a hybrid Taylor series expansion / inverse distance weighting approach was later developed by Gao et al. [27].

In addition to numerical stability and stencil adaption capability, correctly enforcing different types of boundary conditions in a straightforward and consistent manner is another vital factor in obtaining an efficient, accurate, and robust immersed boundary method, since a variety of boundary conditions require to be repeatedly enforced on numerical boundaries with a large number of computational nodes [28]. In solving Navier-Stokes equations, constant temperature at a wall and velocity at a no-slip wall have Dirichlet boundary conditions; pressure at a wall and temperature at an adi-

abatic wall have Neumann boundary conditions; and velocity at a slip wall has a type of Cauchy boundary conditions.

Excluding the Dirichlet boundary conditions in which boundary values are determined and known, the enforcement of other types of boundary conditions, particularly Cauchy boundary conditions, for immersed boundaries demands considerable efforts [20, 29, 30]. Recently, Kempe et al. [30] firstly devised a numerical implementation of slip wall boundary conditions in the context of immersed boundary methods. However, the realization is not straightforward due to its complexity [30]. Therefore, to enforce a variety of boundary conditions in a straightforward and consistent manner is a considerably challenging task.

To achieve an efficient and robust boundary treatment method for solving flow with arbitrarily irregular and changing geometries on Cartesian grids, this work develops a sharp interface immersed boundary method. By the development of an inverse distance weighting based three-step prediction-correction flow reconstruction scheme for boundary treatment, the proposed method enforces Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner. The developed method serves the objective of solving flow interacting with multiple objects involving collision, agglomeration, penetration, and fragmentation processes.

This paper is structured as below. Section 2 presents the sharp interface immersed boundary method under a generalized framework of ghost-cell immersed boundary treatment. Section 3 describes the employed three-dimensional Navier-Stokes solver of this paper. Section 4 validates the proposed immersed boundary method. Section 5 draws conclusions.

2 A sharp interface immersed boundary method

2.1 Generalized framework

Fig. 1 shows 2D and 3D schematic diagrams of a computational domain with an immersed boundary. G denotes a ghost node, a computational node that locates at the numerical boundaries but outside the physical domain. O denotes a boundary point with \mathbf{GO} as the outward normal vector. I is the image point of ghost node G reflected by the physical boundary.

In the ghost-cell immersed boundary method [15], the numerical boundary treatment is a reconstruction process of variable values at numerical boundaries via physical boundary conditions and variable values at interior physical domain. To construct the flow at numerical boundaries while admitting the existence of physical boundaries, the method of images [31] is an effective way [15, 32]. Therefore, the reconstruction of a generic flow variable ψ at a ghost node G is a two-step approach:

$$\psi_G = 2\psi_O - \psi_I \quad (1a)$$

$$\psi_I = f(x_I, y_I, z_I) \quad (1b)$$

where $f(x, y, z)$ is a local reconstruction function of ψ at spatial point I .

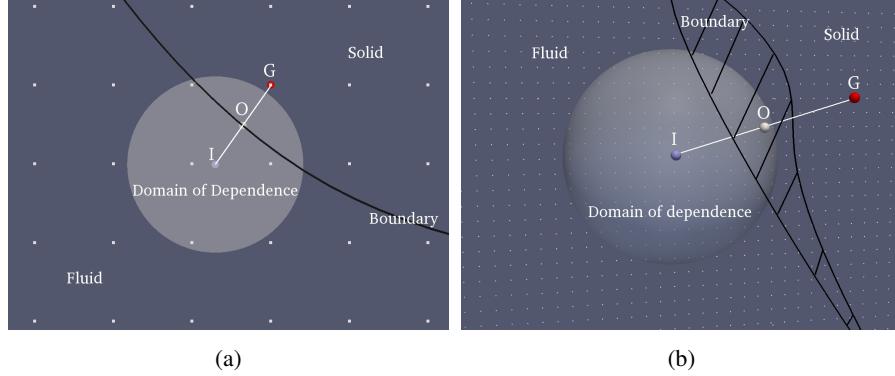


Figure 1: Schematic diagrams of a computational domain with an immersed boundary. G , ghost node; O , boundary point; I , image point. (a) 2D space. (b) 3D Space.

Generally, the reconstruction function needs to be determined by physical boundary conditions and known values of ψ at nearby fluid nodes. That is,

$$\psi_I = f(\{\psi_N\}, \psi_O) \quad (2)$$

where, ψ_O is the value of ψ at the boundary point O where physical boundary conditions are enforced; and $\{\psi_N\}$ represent values of ψ at fluid nodes $\{N\}$ that satisfy:

$$d_N = \|\mathbf{r}_I - \mathbf{r}_N\| \leq R_I \quad (3)$$

where, \mathbf{r}_I and \mathbf{r}_N are the position vectors of point I and N , respectively; R_I , referred as the domain of dependence of point I and illustrated in Fig. 1, is the maximum distance from the point I to nearby fluid nodes that are employed for flow reconstruction.

The benefit of the incorporation of physical boundary conditions in the reconstruction function of ψ_I can be demonstrated by the following equations:

$$\|\mathbf{r}_G - \mathbf{r}_O\| = \|\mathbf{r}_I - \mathbf{r}_O\| \quad (4a)$$

$$\lim_{\|\mathbf{r}_I - \mathbf{r}_O\| \rightarrow 0} \psi_I = \lim_{\|\mathbf{r}_I - \mathbf{r}_O\| \rightarrow 0} f(\{\psi_N\}, \psi_O) = \psi_O \quad (4b)$$

$$\lim_{\|\mathbf{r}_G - \mathbf{r}_O\| \rightarrow 0} \psi_G = 2\psi_O - \lim_{\|\mathbf{r}_I - \mathbf{r}_O\| \rightarrow 0} \psi_I = \psi_O \quad (4c)$$

Hence, the constructed ψ_G converges to the exact physical boundary conditions when G converges to O . This convergence property is helpful to alleviate unphysical flux over the immersed boundary, an issue resulting from using non-body conformal Cartesian grids [13, 33, 34] and examined in the numerical results of this paper.

Several approaches are available to construct flow at image points [15, 18, 27, 35, 36]. In this study, we develop an inverse distance weighting based flow reconstruction function to achieve efficient and robust boundary treatment for flow with arbitrarily irregular and changing geometries and to enforce Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner.

2.2 Inverse distance weighting interpolation

As a convex combination of candidate stencils, inverse distance weighting is a popular interpolation method for the approximation of scattered data sets [37, 38]. The inverse distance weighting [37] for interpolating the value of a variable ψ at a spatial point c is expressed as the following:

$$\psi_c = \frac{\sum w(d_n) \psi_n}{\sum w(d_n)}, \quad d_n \neq 0 \text{ and } d_n \leq R_c \quad (5)$$

where ψ_c is the interpolated value; $\{\psi_n\}$ are the known values of ψ at stencil points $\{n\}$; $\{d_n\}$ are the distance from $\{n\}$ to the interpolated point c ; $\{w(d_n)\}$ are weighting functions; R_c is the size of the domain of dependence for interpolating ψ at point c .

As discussed in [37], the desired $\lim_{d_n \rightarrow 0} \psi_c = \psi_n$ is mathematically satisfied. However, an overflow problem may arise from calculating an inverse distance. Shepard [37] suggested using a conditional statement to avoid this issue:

$$d = \max(d, d_{tiny}) \quad (6)$$

where d_{tiny} is a predefined positive constant to avoid float arithmetic overflow of inverting distance. In this paper, this value is set as a function of mesh sizes:

$$d_{tiny} = \epsilon_0 * \min(\Delta_x, \Delta_y, \Delta_z) \quad (7)$$

where, $\Delta_x, \Delta_y, \Delta_z$ are mesh sizes in x, y, z directions respectively; ϵ_0 is a constant representing the proportion of d_{tiny} to the smallest mesh size, for instance, $\epsilon_0 = 1.0 \times 10^{-6}$.

Generally, the weighting function $w(d)$ employs an inverse-power law $1/d^q$, and the typical value of q is 2 [37, 39]. Our numerical experiments on $q = 1, 2$ with $R_c/\Delta_{max} = 2, 4, 6$ ($\Delta_{max} = \max(\Delta_x, \Delta_y, \Delta_z)$) indicate that computational results are not sensitive to the choice of q and R_c . Hence, $q = 2$ with $R_c/\Delta_{max} = 2$ are used for the numerical results of this paper.

2.3 A three-step prediction-correction flow reconstruction scheme

The proposed three-step prediction-correction scheme of this paper for constructing $\psi_I = f(\{\psi_N\}, \psi_O)$ with ψ representing a generic field variable is presented as below:

1. Prediction step: pre-estimate the value of ψ_I by applying inverse distance weighting on the fluid nodes that locate in the domain of dependence of the image point I . Denote the predicted value as ψ_I^* .

$$\psi_I^* = \frac{\sum w(d_N) \psi_N}{\sum w(d_N)} \quad (8)$$

2. Physical boundary condition enforcement step: determine the value of ψ_O via the physical boundary conditions that ψ needs to satisfy at the boundary point O and the values of ψ at interior physical domain.

3. Correction step: solve the value of ψ_I by adding the boundary point O as a stencil node for the inverse distance weighting of ψ_I .

$$\psi_I = \frac{\sum w(d_N)\psi_N + w(d_O)\psi_O}{\sum w(d_N) + w(d_O)} = \frac{\psi_I^* + \frac{w(d_O)}{\sum w(d_N)}\psi_O}{1 + \frac{w(d_O)}{\sum w(d_N)}} \quad (9)$$

It is beneficial to note that there is no need to re-do calculations on fluid nodes when the sum of weights and sum of weighted values in Eq. (8) are preserved.

The physical boundary condition enforcement step is described below by the implementation of practical boundary conditions.

Dirichlet boundary condition If ψ satisfies Dirichlet boundary condition, the value of ψ_O is purely determined by the specified boundary condition:

$$\psi_O = g \quad (10)$$

where g is a given value or function.

Neumann boundary condition ψ satisfies the following equation:

$$\left. \frac{\partial \psi}{\partial n} \right|_O = \frac{\partial \psi_O}{\partial n} \quad (11)$$

where $\partial \psi_O / \partial n$ is a given value or function. Rewritten Eq. (11) as the following:

$$\lim_{l \rightarrow 0} \frac{\psi(\mathbf{r}_O + l\mathbf{n}) - \psi(\mathbf{r}_O)}{l} = \frac{\partial \psi_O}{\partial n} \quad (12)$$

where \mathbf{r}_O is the position vector and \mathbf{n} is the unit normal vector at boundary point O .

Since point I is on the normal direction of point O , we have:

$$\mathbf{n} = \frac{\mathbf{r}_I - \mathbf{r}_O}{\|\mathbf{r}_I - \mathbf{r}_O\|} \quad (13)$$

Therefore,

$$\frac{\psi_I - \psi_O}{\|\mathbf{r}_I - \mathbf{r}_O\|} - \left. \frac{\partial^2 \psi}{\partial n^2} \right|_O \|\mathbf{r}_I - \mathbf{r}_O\| + O(\|\mathbf{r}_I - \mathbf{r}_O\|^2) = \frac{\partial \psi_O}{\partial n} \quad (14)$$

Due to Eq. (1a), the second order derivative term is negligible:

$$\left. \frac{\partial^2 \psi}{\partial n^2} \right|_O = \frac{\psi_I + \psi_G - 2\psi_O}{2\|\mathbf{r}_I - \mathbf{r}_O\|^2} + O(\|\mathbf{r}_I - \mathbf{r}_O\|^2) \quad (15)$$

Hence, ψ_O is determined as:

$$\psi_O = \psi_I - \|\mathbf{r}_I - \mathbf{r}_O\| \frac{\partial \psi_O}{\partial n} \quad (16)$$

Robin boundary condition A linear combination of the values of ψ and its normal derivative on the boundary point O is specified:

$$\alpha\psi_O + \beta \frac{\partial\psi}{\partial n} \Big|_O = g \quad (17)$$

where α and β are the linear combination coefficients, g is a given value or function.

After approximating the normal derivative, we have:

$$\alpha\psi_O + \beta \frac{\psi_I - \psi_O}{\|\mathbf{r}_I - \mathbf{r}_O\|} = g \quad (18)$$

Then,

$$\psi_O = \frac{\beta\psi_I - \|\mathbf{r}_I - \mathbf{r}_O\|g}{\beta - \|\mathbf{r}_I - \mathbf{r}_O\|\alpha} \quad (19)$$

Cauchy boundary condition For illustration purpose, ψ is replaced by the velocity $\mathbf{V} = (u, v, w)$ that satisfies the slip wall boundary condition:

$$(\mathbf{V} \cdot \mathbf{n})|_{\mathbf{r}=\mathbf{r}_O} = \mathbf{V}_S \cdot \mathbf{n} \quad (20a)$$

$$\frac{\partial(\mathbf{V} \cdot \hat{\mathbf{t}})}{\partial n} \Big|_{\mathbf{r}=\mathbf{r}_O} = 0 \quad (20b)$$

$$\frac{\partial(\mathbf{V} \cdot \tilde{\mathbf{t}})}{\partial n} \Big|_{\mathbf{r}=\mathbf{r}_O} = 0 \quad (20c)$$

where \mathbf{n} , $\hat{\mathbf{t}}$, and $\tilde{\mathbf{t}}$ are the unit normal vector, unit tangent vector, and unit bitangent vector at boundary point O , respectively. \mathbf{V}_S is the velocity of the boundary surface.

After approximating normal derivatives, we have:

$$u_O n_x + v_O n_y + w_O n_z = u_S \hat{n}_x + v_S \hat{n}_y + w_S \hat{n}_z \quad (21a)$$

$$u_O \hat{t}_x + v_O \hat{t}_y + w_O \hat{t}_z = u_I \hat{t}_x + v_I \hat{t}_y + w_I \hat{t}_z \quad (21b)$$

$$u_O \tilde{t}_x + v_O \tilde{t}_y + w_O \tilde{t}_z = u_I \tilde{t}_x + v_I \tilde{t}_y + w_I \tilde{t}_z \quad (21c)$$

Since the coefficient matrix is orthogonal, \mathbf{V}_O is determined as below:

$$\begin{pmatrix} u_O \\ v_O \\ w_O \end{pmatrix} = \begin{bmatrix} n_x & n_y & n_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}^T \begin{pmatrix} u_S \hat{n}_x + v_S \hat{n}_y + w_S \hat{n}_z \\ u_I \hat{t}_x + v_I \hat{t}_y + w_I \hat{t}_z \\ u_I \tilde{t}_x + v_I \tilde{t}_y + w_I \tilde{t}_z \end{pmatrix} \quad (22)$$

All the solution equations of ψ_O now can be written in a unified form:

$$\psi_O = C\psi_I + R.R.H.S. \quad (23)$$

where, the value of the coefficient C and the rest right hand side $R.R.H.S.$ are in Table 1.

Table 1: Value map of C and $R.R.H.S.$ for different boundary conditions.

Type	Example Form	C	$R.R.H.S.$
Dirichlet	$\psi_O = g$	0	g
Neumann	$\frac{\partial \psi}{\partial n} \Big _{\mathbf{r}_O} = \frac{\partial \psi_O}{\partial n}$	1	$- \mathbf{r}_I - \mathbf{r}_O \frac{\partial \psi_O}{\partial n}$
Robin	$\alpha \psi_O + \beta \frac{\partial \psi}{\partial n} \Big _{\mathbf{r}_O} = g$ $(\mathbf{V} \cdot \mathbf{n}) \Big _{\mathbf{r}=\mathbf{r}_O} = \mathbf{V}_S \cdot \mathbf{n}$	$\frac{\beta}{\beta - \mathbf{r}_I - \mathbf{r}_O \alpha}$	$\frac{- \mathbf{r}_I - \mathbf{r}_O g}{\beta - \mathbf{r}_I - \mathbf{r}_O \alpha}$
Cauchy	$\frac{\partial (\mathbf{V} \cdot \hat{\mathbf{t}})}{\partial n} \Big _{\mathbf{r}=\mathbf{r}_O} = 0$ $\frac{\partial (\mathbf{V} \cdot \tilde{\mathbf{t}})}{\partial n} \Big _{\mathbf{r}=\mathbf{r}_O} = 0$	$\begin{bmatrix} n_x & n_y & n_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}$	$\begin{bmatrix} n_x & n_y & n_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}^T \begin{bmatrix} n_x & n_y & n_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{V}_S$

Due to the unknown ψ_I in Eq. (23), the solution equation of ψ_O is coupled with the solution equation of ψ_I in the correction step. To solve this problem, one method is a synchronous solving approach to solve ψ_O and ψ_I simultaneously:

$$\begin{cases} \psi_O = C\psi_I + R.R.H.S. \\ \psi_I = \frac{\psi_I^* + \frac{w(d_O)}{\sum w(d_N)} \psi_O}{1 + \frac{w(d_O)}{\sum w(d_N)}} \end{cases} \quad (24)$$

The other is an asynchronous solving approach: first, solve ψ_O via approximating the unknown ψ_I with the pre-estimated ψ_I^* , then, solve ψ_I in the correction step.

$$\begin{cases} \psi_O = C\psi_I^* + R.R.H.S. \\ \psi_I = \frac{\psi_I^* + \frac{w(d_O)}{\sum w(d_N)} \psi_O}{1 + \frac{w(d_O)}{\sum w(d_N)}} \end{cases} \quad (25)$$

The enforcement of Dirichlet and trivial Neumann boundary conditions is equivalent in these two approaches. When the asynchronous solving approach is adopted, the physical boundary condition enforcement step and the correction step can be iteratively implemented to improve the accuracy of enforcing other boundary conditions.

2.4 Method discussion

The proposed three-step prediction-correction flow reconstruction scheme enables the developed immersed boundary method to enforce a wide class of boundary conditions in a straightforward and consistent manner. In polynomial reconstruction based methods, different linear systems require to be constructed and solved for flow variables satisfying different boundary conditions. In the current method, the enforcement of boundary conditions at boundary points is a separate step of flow reconstruction. This feature provides efficient and uniform boundary treatment for flow with an arbitrary number of field variables that satisfy different types of boundary conditions.

Moreover, the proposed immersed boundary method here is scalable to the number of stencils used in the flow reconstruction. In contrast, for polynomial reconstruction based methods, a fixed minimum number of stencils is always required to avoid under-determined linear systems, and special treatments are required for strongly concave or

convex geometries [18, 27]. Therefore, the scalable property is where the robustness of the method herein lies: it leads to the automatic adaption to a varying number of stencil nodes; and it guarantees uniform validity when at least one fluid node exists in the domain of dependence of the image point, a condition that is ensured by the definition of a ghost node. In addition, the proposed immersed boundary method can be applied to multiple layers of ghost nodes without extra constraints.

The asynchronous solving approach without iterative implementation is currently adopted and examined in this paper, since the validity of the synchronous solving approach is established when the validity of the asynchronous solving approach is proved. A code that features the implementation of the three-step prediction-correction flow reconstruction scheme is provided in the A to illustrate the simplicity and efficiency of our proposed immersed boundary method.

3 Numerical implementation

The governing equations employed in the numerical solver of this paper is the nondimensionalized conservative form of the three-dimensional Navier-Stokes equations in Cartesian coordinates [40]. The temporal derivatives in the governing equations are discretized by using the third-order TVD Runge-Kutta method [41, 42]. The second order upwind TVD scheme [43] and the fifth order WENO scheme [44] are both implemented for the discretization of convective fluxes. Meanwhile, central differencing scheme is used for the discretization of diffusive fluxes [45]. Strang splitting [46, 47] is employed for dimensional splitting to relax stability constraints and decrease numerical complexities [48]. The fluid-solid coupling pattern used in this paper is described in B.

4 Numerical experiments

Due to the using of the Navier-Stokes equations, the no-slip wall condition with an adiabatic assumption is enforced in the test cases unless otherwise stated. Numerical results are computed by the fifth order WENO scheme unless otherwise stated.

4.1 Shock diffraction over a cylinder

A Mach 2.81 planar incident shock interacting with a stationary circular cylinder is considered. Comprehensive descriptions of this problem are available in [49–53]. In the current study, a circular cylinder with diameter $D = 1$ is positioned at the center of a $6D \times 6D$ square domain while an initial shock is positioned $0.5D$ upstream of the cylinder. This configuration is similar to [52] except that a full domain size without symmetrical boundary assumption is used in this paper.

4.1.1 Grid sensitivity test

A consistent numerical method should provide numerical solutions that become less sensitive to the grid size as the mesh is refined. However, the level of numerical error

depends on the features of the flow resolved by the grid [54]. The presence of discontinuities, such as shocks, slip surfaces, and interfaces, develops numerical errors on a grid in a complex way [54] and affects the error estimation process, especially for the evaluation of local grid convergence behavior [55]. While admitting these difficulties, this part provides an examination of the grid convergence behavior of the presented immersed boundary method for solving flow with strong discontinuities.

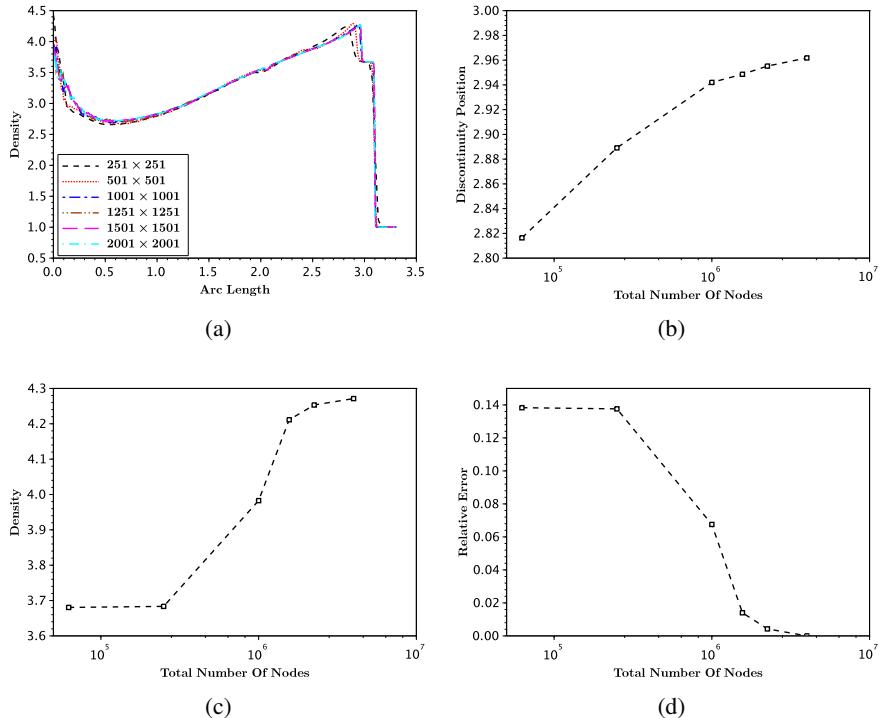


Figure 2: Grid sensitivity test. (a) Global grid convergence at a 33° line. (b) Global grid convergence on shock position. (c) Local grid convergence at a point with $\text{Arc Length} = 2.96$. (d) Local relative error at the sample point.

A series of successively refined grids are employed to study the grid sensitivity of the developed immersed boundary method, and the numerical results are obtained at $t = 1.0$. Due to the presence of complex discontinuity patterns, the global grid convergence behavior is represented by and examined on a line segment from point $(-0.27232, 0.41934)$ to point $(2.5, 2.21970)$, which is located on a 33° tangent line of the cylinder and is plotted in Fig. 6.

As shown in Fig. 2a, when grid resolution changes significantly from the coarsest 251×251 grid to the finest 2001×2001 grid, excellent overall agreements are achieved on global grid convergence with main discrepancies occurring near the region of boundary interfaces and flow discontinuities. However, these discrepancies are

effectively reduced when the grid resolution is sufficiently increased. By observing that the discrepancy of the predicted discontinuity is mainly about its position rather than magnitude, a global grid convergence behavior on discontinuity position is examined and shown in Fig. 2b, in which a well performed grid convergence is presented.

Since local errors are transported throughout the computational region and are strongly affected by flow discontinuities, an examination of the local grid convergence behavior has limited implications [55]. For instance, the predicted position of discontinuities, which highly depends on grid resolution, has major influence on the values of flow quantities near discontinuities. Nonetheless, the local grid convergence at a sample point, which corresponds to the peak value of the 2001×2001 grid solution, is examined in this paper. Fig. 2c shows the predicted density at this sample point, and Fig. 2d shows the numerical error relative to the predicted value of the finest grid. As shown in Fig. 2d, the local relative error of the predicted density at this sample point is effectively reduced as the grid is refined.

According to the discussed results, the developed immersed boundary method of this paper has well-behaved global and local grid convergence properties over a wide range of grid resolution.

4.1.2 Numerical results

Numerical results of three grids with 501×501 , 1001×1001 , and 2001×2001 nodes, denoted as grid *C*, grid *M*, and grid *F* respectively, are discussed below.

Numerical results of grid *M* are illustrated in Fig. 3 in the form of the time evolution of density contour lines. These snapshots clearly show the diffraction of the curved Mach stems over the cylinder and the formation of wake by the collision of the two opposite diffracting shocks, which physical processes are reported in experimental studies [49, 50]. Comparing with the interferometric measurements of [51] and numerical results in [52, 53, 56], the slip line, the reflected and diffracted shocks over the immersed boundary in the results of this paper are all resolved remarkably well, which demonstrates the high validity of the developed method.

To investigate the capability of the developed immersed boundary method on enforcing different physical boundary conditions, two types of wall boundary conditions are studied on grid *M* and results at $t = 1.0$ are shown in Fig. 4. After the formation of collision wakes as in Fig. 4a and Fig. 4c, a more wall-adhesive low-speed wake is formed in the case of no-slip wall. Fig. 4b and Fig. 4d show the velocity gradient distribution at the wall region. A thin but gradually growing boundary layer with large velocity gradients is produced along the no-slip wall, while no boundary layer is presented in the slip wall case.

Fig. 5 further shows the x-velocity profiles at a vertical line segment from point $(0, 0.50)$ to point $(0, 0.55)$. A velocity profile that indicates a zero velocity at wall changing to local free-stream value away from the wall is presented in the no-slip wall case. A velocity profile that indicates a maximum velocity at wall decreasing to local free-stream velocity is presented in the slip wall case. These observations agree with the flow physics at this vertical line of flow over a cylinder.

According to the successful solutions of shock diffraction over cylinder with no-slip and slip wall boundary conditions, the developed immersed boundary method is

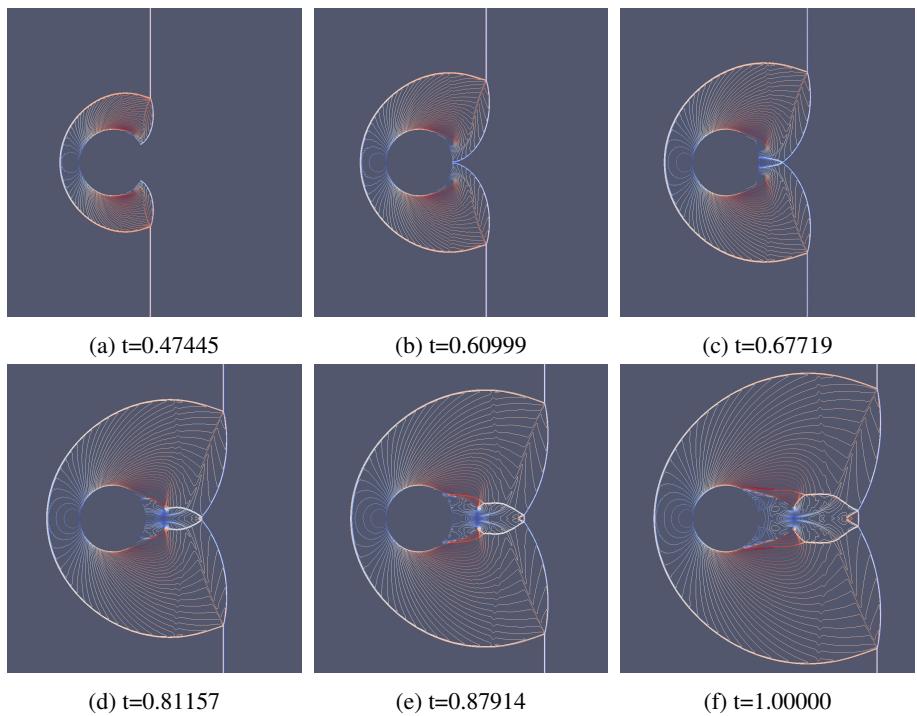


Figure 3: Density contour of shock diffraction over a cylinder. Colored by velocity.

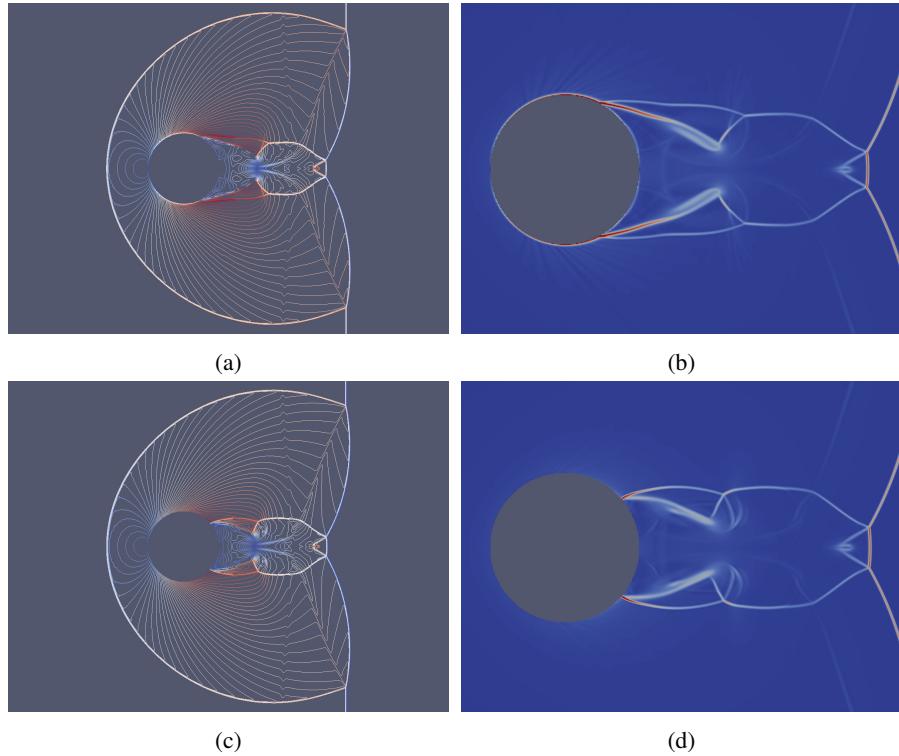


Figure 4: Numerical results with two types of wall boundary conditions. (a) Density contour colored by velocity, no-slip wall. (b) Velocity gradient color map, no-slip wall. (c) Density contour colored by velocity, slip wall. (d) Velocity gradient color map, slip wall.

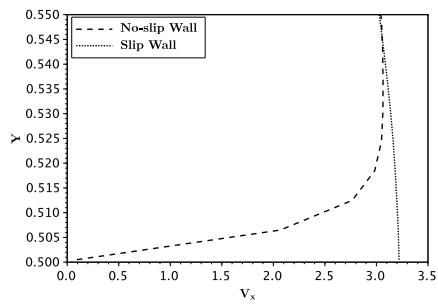


Figure 5: X-velocity profiles at a line segment from point $(0, 0.50)$ to point $(0, 0.55)$.

able to correctly enforce different types of physical boundary conditions.

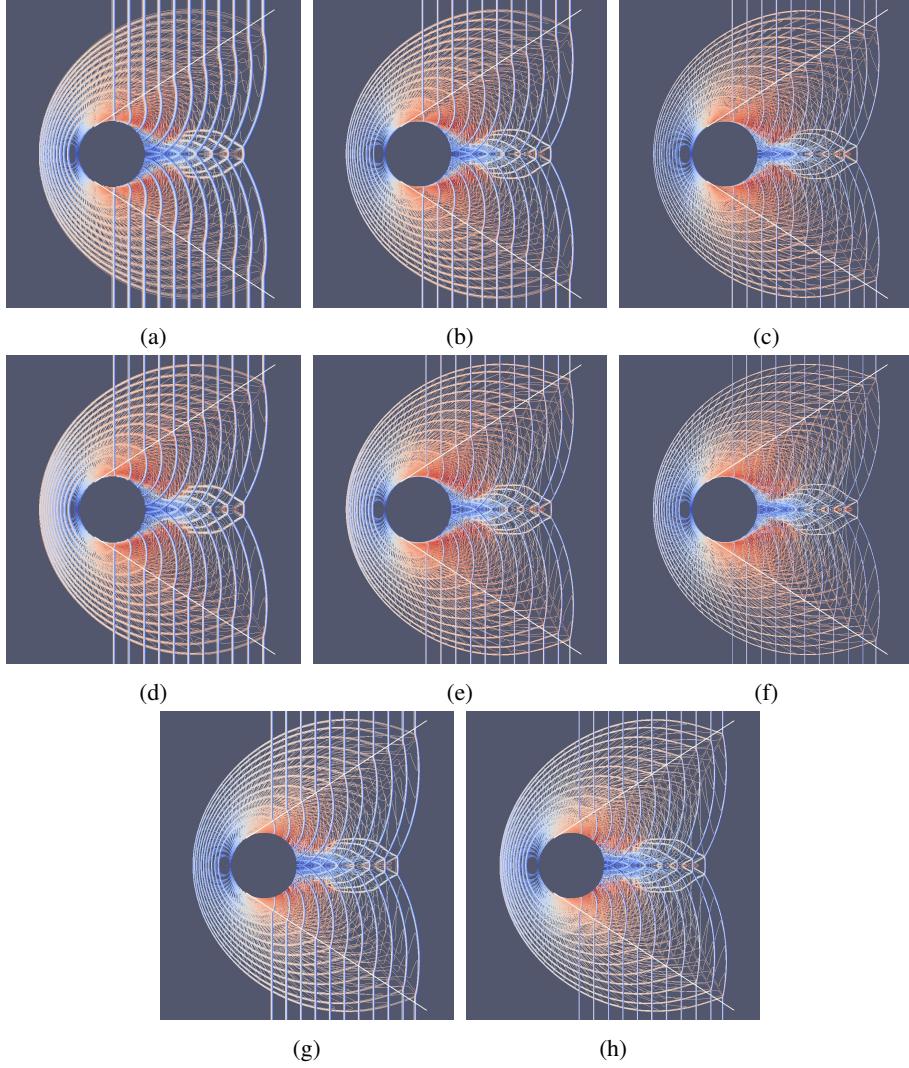


Figure 6: Superimposition of density contours (white lines are 33° tangent lines of the cylinder). (a) TVD, grid C. (b) TVD, grid M. (c) TVD, grid F. (d) WENO, grid C. (e) WENO, grid M. (f) WENO, grid F. (g) TVD, grid M, slip wall. (h) WENO, grid M, slip wall.

4.1.3 Shock detachment distance

As shown in Fig. 6, a bow-shaped shock is reflected from the cylinder when the incident shock hits the cylinder. In [51], the concepts of shock detachment distance and

nondimensional time after collision were introduced to describe the reflected incident shock. Fig. 7 shows the comparison of predicted detachment distance between present numerical results and the experimental results of [51].

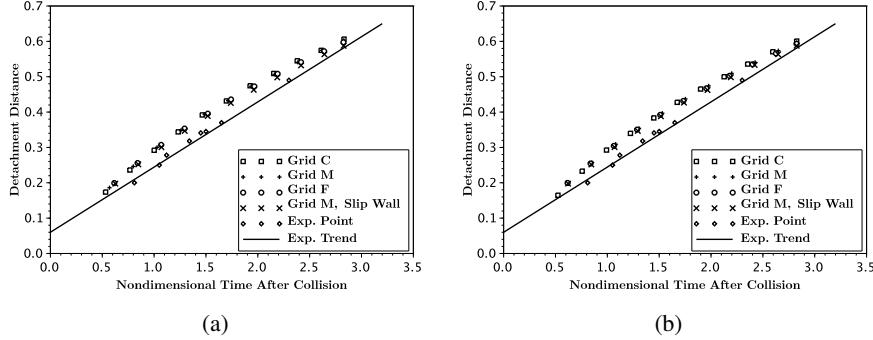


Figure 7: Comparison of detachment distance. (a) TVD. (b) WENO.

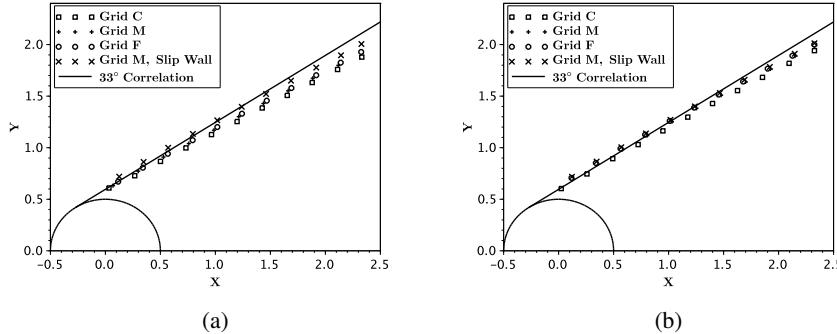


Figure 8: Comparison of triple-point path. (a) TVD. (b) WENO.

As it is pointed out in [52] that numerical studies of the detachment distance generally predicts a greater detachment distance, and the detachment distance subjects to a parabolic distribution with respect to the nondimensional time after collision rather than the linear behavior reported in [51]. This parabolic behavior of the detachment distance, which is clearly observed in the results of this paper, is also presented in polynomial reconstruction based results of [53], cut-cell based results of [56], and unstructured mesh based results of [52].

4.1.4 Triple-point path

The incident shock, reflected shock, and diffracted shock intersect and form an upper triple point on each side of the plane of symmetry. As shown in Fig. 6, this triple point

travels in space and produces an upper triple-point path. Interferometric measurements of [51] predict that this upper triple-point path is tangent to the cylinder at an angle of 33° for Mach numbers in the range of $1.42 - 5.96$.

The predicted triple-point paths of this paper and the experimental correlation of [51] are compared in Fig. 8. The least square linear regressions of the predicted triple-point paths of TVD with grid C , M , and F are about 29.0° , 29.5° , and 29.6° respectively, of WENO with grid C , M , and F are about 30.2° , 30.3° , and 30.3° respectively. For grid M with slip wall boundary condition, TVD and WENO predict 30.1° and 30.3° respectively. These results, which agree well with the experimental correlation of [51] and very well with polynomial reconstruction based results of [53], cut-cell based results of [56], and unstructured mesh based results of [52], demonstrate the accuracy of the developed method.

4.2 Explosive dispersal of zero-gap particles

To test the robustness of the proposed method, strongly irregular, concave, and changing geometries formed by initially zero-gap configured particles are used. The zero-gap particles may represent one of the most challenging geometries in fluid-solid interactions. As preliminary cases for studying particle jetting instabilities from multiphase explosive detonation [2, 57], no study case concerning zero-gap particles, which is the common setting when particles are packed in practice, has been found so far.

As shown in Fig. 9, eighteen identical particles are zero-gap configured in a 1×1 computational domain. The centers of particles are evenly distributed on a circle whose radius is equal to 0.2. A flow state $(\rho, u, v, p)^T = (3.67372, 0, 0, 9.04545)^T$ is initially positioned at a circular region centered in the domain, and the radius of the circular region is 0.1. The flow state at the rest of the region is set to $(\rho, u, v, p)^T = (1, 0, 0, 1)^T$. A grid with 2001×2001 nodes is used for the numerical solution. Newton's second law of motion is employed for evolving spatial distribution of particles, during which no collision and gravity effects are currently integrated.

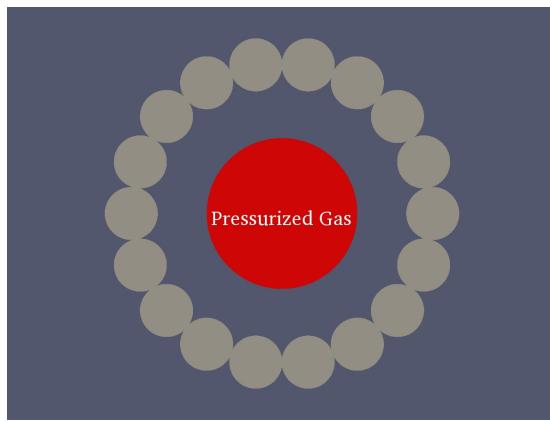


Figure 9: Illustration of computational configuration of zero-gap particles.

Numerical solutions of the explosive dispersal process are illustrated in Fig. 10. Complex shock-shock interaction, shock reflection and diffraction behaviors are clearly resolved in the numerical results. In addition, compression waves are formed in front of the moving particles, and high velocity fluid jets are generated at regions between moving particles. These physical and successful solutions of explosive dispersal of zero-gap particles prove the high robustness of the proposed immersed boundary method for strongly irregular, concave, and changing geometries.

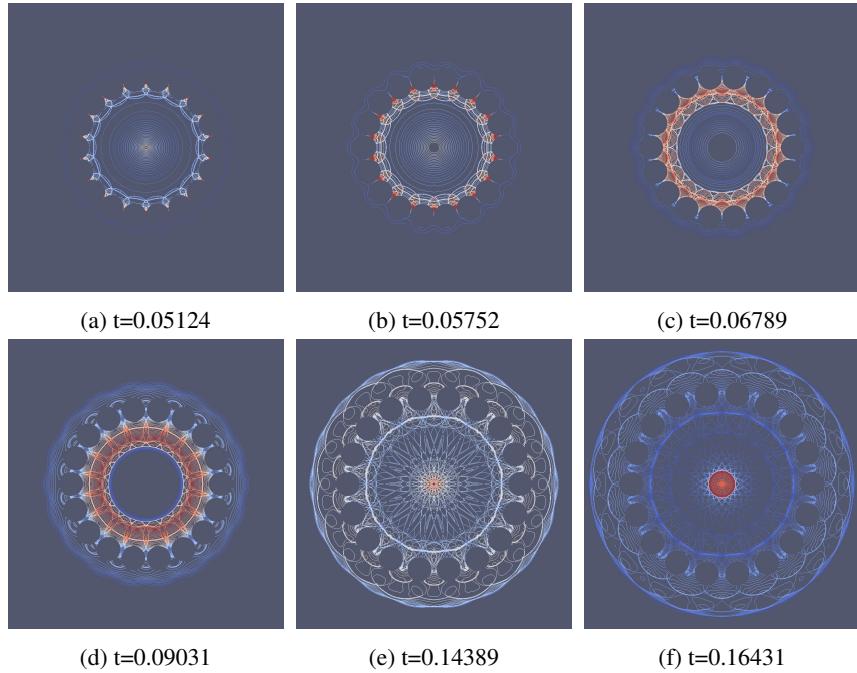


Figure 10: Explosive dispersal of zero-gap particles. Colored by pressure.

4.3 Shock diffraction over two partially overlapped spheres

A Mach 2.81 planar shock interacting with two stationary and partially overlapped spheres (Fig. 11a) is solved to test the proposed immersed boundary method for three-dimensional irregular and concave geometries. The diameter of these two identical spheres is $D = 1$, and the centers of them are located at $(0, 0, 0)$ and $(0.5D, 0, 0)$ respectively. The size of the computational domain is $6D \times 6D \times 6D$, and the computational grid is $251 \times 251 \times 251$ nodes.

Fig. 11b shows density contour on two perpendicular semi-planar slices. As shown in the two identical density contour slices, flow discontinuities resulted from the concave region are adequately resolved on this relatively coarse grid. Moreover, the axis symmetry property of this flow problem is able to be preserved.

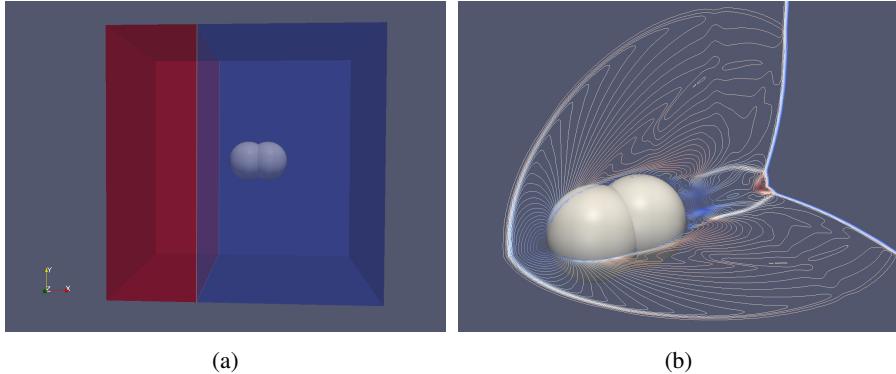


Figure 11: Shock interacting with two partially overlapped spheres. (a) Computational domain. (b) Density contour on two perpendicular semi-planar slices. Colored by velocity.

Due to using non-body conformal Cartesian grids, unphysical flux over the immersed boundary is a fundamental issue in immersed boundary methods [13], and considerable efforts to overcome this issue have been devoted in [33, 34]. Since Eq. (4) has established the convergence property of current method that the constructed ghost flow is able to converge to exact physical boundary conditions, we now numerically examine the unphysical flux under practical grid sizes by synthesizing the solved problems.

Fig. 12 presents the stream traces of shock diffraction problems. The solved stream traces by the developed immersed boundary method are closely aligned with geometry surfaces, even in the three-dimensional problem where a coarse grid is employed.

To quantitatively examine the unphysical flux over immersed boundary, the absolute flux of the shock diffraction over a cylinder problems at time $t = 1.0$ is examined and the flux is calculated on the first layer of ghost nodes.

$$\text{absolute flux} = \frac{1}{S} \iint_S |(\mathbf{V} - \mathbf{V}_S) \cdot \mathbf{n}| ds \quad (26)$$

The absolute flux distribution $|(\mathbf{V} - \mathbf{V}_S) \cdot \mathbf{n}|$ over the cylinder with a no-slip wall on grid M is plotted in Fig. 13a, in which a symmetrical distribution is presented. The main unphysical flux is observed at the angular range of $\pm[60^\circ, 100^\circ]$, where flow has a large velocity gradient at near wall region.

Fig. 13b plots the absolute flux of different cases, of which the average velocity and maximum velocity of flow fields are provided in Table 2. The slip wall and no-slip wall boundary conditions show similar flux values. When the grid is refined from 251×251 to 2001×2001 , the absolute flux of the no-slip wall case is effectively reduced from about 1.44×10^{-2} to about 4.19×10^{-3} , in which the latter is about 29.09% of the former and is about 0.21% of the average velocity of the flow field.

According to these qualitative and quantitative results, the developed immersed boundary method of this paper retains a sharp interface and is able to effectively alleviate unphysical flux over physical boundaries when grid resolution is improved.

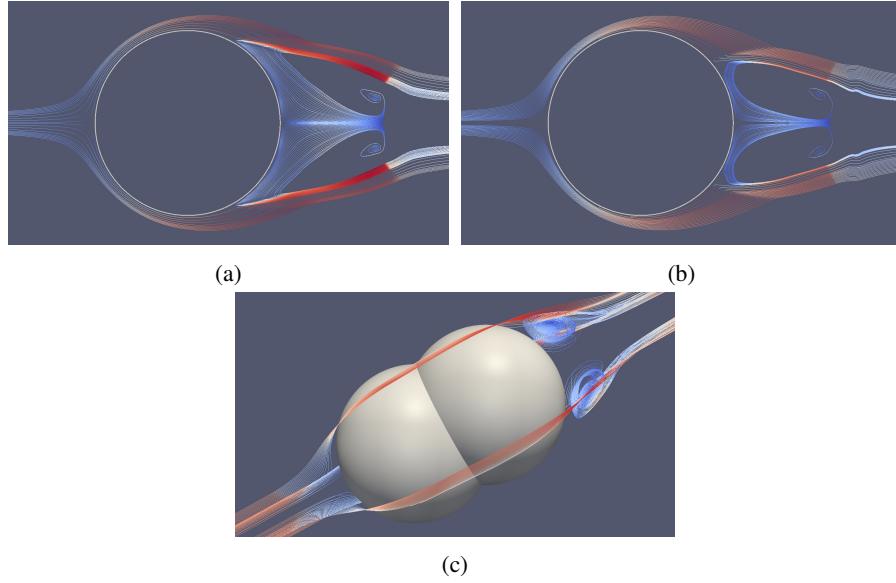


Figure 12: Stream traces colored by velocity with corresponding analytical geometry boundaries. (a) Shock diffraction over a cylinder, no-slip wall. (b) Shock diffraction over a cylinder, slip wall. (c) Shock diffraction over two partially overlapped spheres, no-slip wall.

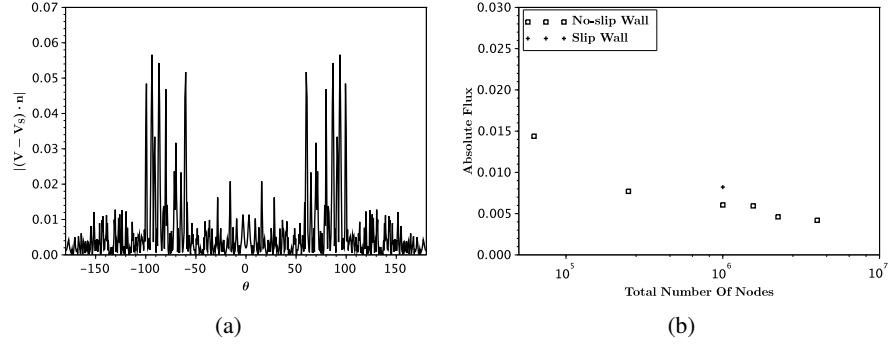


Figure 13: Absolute flux over analytical geometry boundaries of the shock diffraction over a cylinder problems. (a) Absolute flux distribution over the cylinder with no-slip wall on grid M . (b) Absolute flux over cylinder with no-slip or slip wall.

Table 2: Average velocity and maximum velocity of shock diffraction over a cylinder.

	No-slip wall	Slip wall
Average velocity of flow field	2.00	2.01
Maximum velocity in flow field	3.83	4.41

5 Conclusions

A sharp interface immersed boundary method is developed. This method enforces Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner and is able to provide efficient and robust boundary treatment for numerically solving flow with arbitrarily irregular and changing geometries while maintaining accuracy. The effectiveness of the proposed method is conformed by numerical experiments concerning flow of two and three space dimensions, stationary and moving objects, convex and concave geometries, no-slip and slip wall boundary conditions.

A A sample code for boundary treatment implementation

A.1 Ghost flow reconstruction function

```
/*
 * Flow reconstruction of Field_G[N] for N field variables at ghost node G.
 */
/* pre-estimate Field_I[N] in domain of dependence of image point I. */
compute weightedSum[N] and sumOfWeights by Appendix A.2.;
for (int n = 0; n < N; ++n) {
    Field_Istar[n] = weightedSum[n] / sumOfWeights;
}
/* enforce physical boundary conditions to determine Field_O[N] */
Field_O[0] = C[0] * Field_Istar[0] + R.R.H.S.[0];
.
.
.
Field_O[N-1] = C[N-1] * Field_Istar[N-1] + R.R.H.S.[N-1];
/* correction step to solve Field_I[N] */
for (int n = 0; n < N; ++n) {
    Field_I[n] = (weightedSum[n] + Field_O[n] * weight_O) /
        (sumOfWeights + weight_O);
}
/* apply the method of images to construct Field_G[N] at ghost node G */
for (int n = 0; n < N; ++n) {
    Field_G[n] = 2 * Field_O[n] - Field_I[n];
}
```

A.2 A search function for inverse distance weighting

```
/*
 * Search fluid nodes around node(kI, jI, iI) in the domain of dependence R
 * to apply inverse distance weighting. node(kI, jI, iI) is a computational
 * node whose node coordinates are derived from the corresponding spatial
 * coordinates of a image point I(xI, yI, zI).
 */
for (int kh = -R; kh <= R; ++kh) {
    for (int jh = -R; jh <= R; ++jh) {
        for (int ih = -R; ih <= R; ++ih) {
            if (Flag[kI+kh][jI+jh][iI+ih] != FLUID) { /* not a fluid node */
                continue;
            }
        }
    }
}
```

```

        }
        /* a valid stencil node, apply inverse distance weighting */
        compute weight for node(kI+kh, jI+jh, iI+ih) to I(xI, yI, zI);
        sumOfWeights = sumOfWeights + weight;
        for (int n = 0; n < N; ++n) {
            weightedSum[n] = weightedSum[n] +
                Field[kI+kh][jI+jh][iI+ih][n] * weight;
        }
    }
}

```

B Fluid-solid coupling

Currently, a simple fluid-solid coupling pattern is used. The interactions between fluid and solid are explicitly coupled by applying Strang splitting [46, 47] for physical process splitting as the following:

1. Evolve particle dynamics for $\frac{1}{2}\Delta t$
 2. Evolve fluid dynamics for Δt
 3. Evolve particle dynamics for $\frac{1}{2}\Delta t$

The evolution of particle dynamics can be expanded as the following procedures:

1. Integrate surface forces.
 2. Update the spatial distribution of particles by particle dynamics models.
 3. Detect ghost nodes that fall out the regions of their corresponding particles. These ghost nodes now change to fluid nodes, of which values of flow variables can be reconstructed by inverse distance weighting.
 4. Re-mesh computational domain, and apply boundary treatment.

Acknowledgements

Financial support of this work was provided by Natural Sciences and Engineering Research Council of Canada (NSERC) and Defence Research and Development Canada (DRDC). This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca) and Compute/Calcul Canada. The first author of this paper is grateful to Prof. Deliang Zhang for introducing Computational Fluid Dynamics and is thankful to Dr. Deyong Wen for discussions of flow visualization.

References

- [1] D. L. Frost, Y. Grégoire, O. Petel, S. Goroshin, F. Zhang, Particle jet formation during explosive dispersal of solid particles, *Phys. Fluids* 24 (9) (2012) 091109.
- [2] F. Zhang, R. Ripley, A. Yoshinaka, C. Findlay, J. Anderson, B. von Rosen, Large-scale spray detonation and related particle jetting instability phenomenon, *Shock Waves* (2014) 1–16.
- [3] V. Kedrinskiy, Hydrodynamic aspects of explosive eruptions of volcanoes: simulation problems, *Shock Waves* 18 (6) (2009) 451–464.
- [4] D. Lohse, R. Bergmann, R. Mikkelsen, Impact on soft sand: void collapse and jet formation, *Phys. Rev. Lett.* 93 (19) (2004) 198003.
- [5] F. Pacheco-Vázquez, J. Ruiz-Suárez, Impact craters in granular media: grains against grains, *Phys. Rev. Lett.* 107 (21) (2011) 218001.
- [6] F. Zhang, Shock Wave Science and Technology Reference Library: Heterogeneous Detonation, vol. 4, Springer Science & Business Media, 2009.
- [7] V. Rodriguez, R. Saurel, G. Jourdan, L. Houas, Solid-particle jet formation under shock-wave acceleration, *Phys. Rev. E* 88 (6) (2013) 063011.
- [8] C. Liu, S. R. Nagel, D. Schecter, S. Coppersmith, S. Majumdar, O. Narayan, T. Witten, Force fluctuations in bead packs, *SCIENCE* 269 (5223) (1995) 513–513.
- [9] H. M. Jaeger, S. R. Nagel, R. P. Behringer, Granular solids, liquids, and gases, *Rev. Mod. Phys.* 68 (4) (1996) 1259.
- [10] C. S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (2) (1972) 252–271.
- [11] E. Jung, C. S. Peskin, Two-dimensional simulations of valveless pumping using the immersed boundary method, *SIAM J. Sci. Comput.* 23 (1) (2001) 19–45.
- [12] C. S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [13] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [14] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161 (1) (2000) 35–60.
- [15] Y.-H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2) (2003) 593–623.
- [16] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2) (2005) 448–476.

- [17] Y. Mori, C. S. Peskin, Implicit second-order immersed boundary methods with boundary mass, *Comput. Methods Appl. Mech. Engrg.* 197 (25) (2008) 2049–2067.
- [18] A. Kapahi, S. Sambasivan, H. Udaykumar, A three-dimensional sharp interface Cartesian grid method for solving high speed multi-material impact, penetration and fragmentation problems, *J. Comput. Phys.* 241 (2013) 308–332.
- [19] J. Yang, F. Stern, A non-iterative direct forcing immersed boundary method for strongly-coupled fluid-solid interactions, *J. Comput. Phys.* 295 (2015) 779–804.
- [20] S. Schwarz, T. Kempe, J. Fröhlich, An immersed boundary method for the simulation of bubbles with varying shape, *J. Comput. Phys.* .
- [21] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, *CTR-Annual Research Briefs-Stanford Univ./NASA Ames* .
- [22] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, *Comput. Fluids* 33 (3) (2004) 375–404.
- [23] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* 152 (2) (1999) 457–492.
- [24] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, *J. Comput. Phys.* 175 (1) (2002) 200–224.
- [25] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, *Appl. Mech. Rev.* 56 (3) (2003) 331–347.
- [26] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid-structure interaction, *Prog. Aerosp. Sci.* 65 (2014) 1–21.
- [27] T. Gao, Y.-H. Tseng, X.-Y. Lu, An improved hybrid Cartesian/immersed boundary method for fluid-solid flows, *Int. J. Numer. Meth. Fluids* 55 (12) (2007) 1189–1211.
- [28] F. Gibou, C. Min, R. P. Fedkiw, High resolution sharp computational methods for elliptic and parabolic problems in complex geometries, *J. Sci. Comput.* 54 (2-3) (2013) 369–413.
- [29] R. Crockett, P. Colella, D. T. Graves, A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, *J. Comput. Phys.* 230 (7) (2011) 2451–2469.
- [30] T. Kempe, M. Lennartz, S. Schwarz, J. Fröhlich, Imposing the free-slip condition with a continuous forcing immersed boundary method, *J. Comput. Phys.* 282 (2015) 183–209.

- [31] C. Chu, A. Sereny, Boundary conditions in finite difference fluid dynamic codes, *J. Comput. Phys.* 15 (4) (1974) 476–491.
- [32] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.* 87 (1) (1990) 171–200.
- [33] A. Mark, B. G. van Wachem, Derivation and validation of a novel implicit second-order accurate immersed boundary method, *J. Comput. Phys.* 227 (13) (2008) 6660–6680.
- [34] J. H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, *J. Comput. Phys.* 230 (19) (2011) 7347–7363.
- [35] A. Gilmanov, F. Sotiropoulos, E. Balaras, A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, *J. Comput. Phys.* 191 (2) (2003) 660–669.
- [36] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, *J. Comput. Phys.* 207 (2) (2005) 457–492.
- [37] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proc. 23rd Nat. Conf. ACM, ACM, 517–524, 1968.
- [38] J. L. Junkins, G. W. Miller, J. R. Jancaitis, A weighting function approach to modeling of irregular surfaces, *J. Geophys. Res.* 78 (11) (1973) 1794–1803.
- [39] R. Franke, Scattered data interpolation: tests of some methods, *Math. Comput.* 38 (157) (1982) 181–200.
- [40] J. D. Anderson, J. Wendt, Computational Fluid Dynamics, vol. 206, Springer, 1995.
- [41] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (2) (1988) 439–471.
- [42] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Rev.* 43 (1) (2001) 89–112.
- [43] A. Harten, On a class of high resolution total-variation-stable finite-difference schemes, *SIAM J. Numer. Anal.* 21 (1) (1984) 1–23.
- [44] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1).
- [45] J. H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, vol. 3, Springer Berlin, 2002.
- [46] G. Strang, Accurate partial difference methods, *Numer. Math.* 6 (1) (1964) 37–46.

- [47] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (3) (1968) 506–517.
- [48] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1) (1984) 115–173.
- [49] A. Bryson, R. Gross, Diffraction of strong shocks by cones, cylinders, and spheres, *J. Fluid Mech.* 10 (01) (1961) 1–16.
- [50] T. Bazhenova, L. Gvozdeva, M. Nettleton, Unsteady interactions of shock waves, *Prog. Aerosp. Sci.* 21 (1984) 249–331.
- [51] J. Kaca, An interferometric investigation of the diffraction of a planar shock wave over a semicircular cylinder, *Tech. Rep., University of Toronto Institute for Aerospace Studies*, 269, 1988.
- [52] R. Ripley, F.-S. Lien, M. Yovanovich, Numerical simulation of shock diffraction on unstructured meshes, *Comput. Fluids* 35 (10) (2006) 1420–1431.
- [53] S. K. Sambasivan, H. Udaykumar, Ghost fluid method for strong shock interactions Part 2: immersed solid boundaries, *AIAA J.* 47 (12) (2009) 2923–2937.
- [54] W. L. Oberkampf, T. G. Trucano, Verification and validation in computational fluid dynamics, *Prog. Aerosp. Sci.* 38 (3) (2002) 209–272.
- [55] P. J. Roache, *Verification and validation in computational science and engineering*, Hermosa, 1998.
- [56] H. Ji, F.-S. Lien, E. Yee, Numerical simulation of detonation using an adaptive Cartesian cut-cell method combined with a cell-merging technique, *Comput. Fluids* 39 (6) (2010) 1041–1057.
- [57] R. Ripley, F. Zhang, Jetting instability mechanisms of particles from explosive dispersal, in: *J. Phys. Conf. Ser.*, vol. 500, IOP Publishing, 152012–152019, 2014.