An immersed boundary formulation for simulating high-speed compressible viscous flows with moving solids

Yegao Qu a,b, Ruchao Shi a, Romesh C. Batra a,*

a Department of Biomedical Engineering and Mechanics, MC 0219, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
b State Key Laboratory of Mechanical System and Vibration, Shanghai jiao Tong University, Shanghai 200240, China

A R T I C L E   I N F O

Article history:
Received 11 March 2017
Received in revised form 6 September 2017
Accepted 24 October 2017
Available online 26 October 2017

Keywords:
Navier–Stokes equations
Fluid–solid interface
Immersed boundary method
Constrained moving least-squares method
Flow discontinuities

A B S T R A C T

We present a robust sharp-interface immersed boundary method for numerically studying high speed flows of compressible and viscous fluids interacting with arbitrarily shaped either stationary or moving rigid solids. The Navier–Stokes equations are discretized on a rectangular Cartesian grid based on a low-diffusion flux splitting method for inviscid fluxes and conservative high-order central-difference schemes for the viscous components. Discontinuities such as those introduced by shock waves and contact surfaces are captured by using a high-resolution weighted essentially non-oscillatory (WENO) scheme. Ghost cells in the vicinity of the fluid–solid interface are introduced to satisfy boundary conditions on the interface. Values of variables in the ghost cells are found by using a constrained moving least squares method (CMLS) that eliminates numerical instabilities encountered in the conventional MLS formulation. The solution of the fluid flow and the solid motion equations is advanced in time by using the third-order Runge–Kutta and the implicit Newmark integration schemes, respectively. The performance of the proposed method has been assessed by computing results for the following four problems: shock-boundary layer interaction, supersonic viscous flows past a rigid cylinder, moving piston in a shock tube and lifting off from a flat surface of circular, rectangular and elliptic cylinders triggered by shock waves, and comparing computed results with those available in the literature.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The interactions of high-speed compressible viscous flows with irregularly shaped objects are commonly encountered in aerospace applications. These interactions may encompass various flow phenomena including shock wave reflection and diffraction, as well as shock–shock, shock-vortex and shock-boundary layer interactions. In addition to the challenge of representing various discontinuities in a high-speed flow, simulating an irregular-shaped solid moving in a compressible viscous flow is very challenging. Numerical methods used to solve such problems employ either a finite-difference or a finite volume grid to represent the computational domain and a variety of algorithmic approaches to satisfy continuity conditions at the fluid–solid interface.

An ideal numerical method for simulating high-speed flows should be accurate and free from numerical dissipation in smooth parts of the flow, and must robustly capture flow discontinuities without significant Gibbs ringing that can lead...
to nonlinear instability [1]. The shock-capturing upwind-biased schemes commonly employed to suppress the Gibbs oscillations include, but are not limited to, the total variation diminishing (TVD) methods with flux or slope limiters [2], the monotonicity-preserving (MP) methods [3–5], the essentially non-oscillatory (ENO) [6,7] methods and the weighted essentially non-oscillatory (WENO) [8,9] methods. The TVD methods can be easily implemented but they reduce to first-order accuracy at the local extrema of solutions and can be numerically very diffusive for computing solutions involving oscillatory waves. The MP method proposed by Suresh and Huynh [3] generalizes the TVD schemes and shows very good performance in preserving both the accuracy in smooth flow regions as well as the monotonicity near discontinuous flow regions [4,5].

The ENO schemes determine the numerical flux from a high-order reconstruction over an adaptive stencil that is selected to minimize interpolation across discontinuities and hence diminish Gibbs oscillations. However, the ENO schemes are found to be less stable for computing steady flows than the TVD techniques since the TVD condition is not rigorously satisfied in the ENO schemes [10,11]. In a WENO scheme, a high-order numerical flux is constructed by using a convex linear combination of lower-order polynomial reconstructions over a set of staggered stencils, with weights selected to achieve the maximum formal order of accuracy in smooth regions, and nearly zero weight assigned to reconstructions on stencils crossed by discontinuities. The WENO schemes improve robustness, convergence and efficiency of the ENO schemes, and tend to have uniform higher-order accuracy in smooth regions and maintain the essentially non-oscillatory properties near shock waves. Several WENO methods including the third-, the fifth- and the higher-order have been developed [12,13]. For high-speed flows, the WENO schemes seem to have superseded other shock-capturing methods in the last decade and have proved to be extremely accurate and robust in the presence of strong shock waves and complex shock interactions [1].

For numerically simulating flows interacting with complex-shaped solids, the correct enforcement of boundary conditions on the fluid–solid interface is important for the accuracy and stability of the numerical method. The body-fitted grid methods [14–16] have been commonly employed for such problems that transform governing equations and boundary conditions of the fluid into body-fitted coordinate systems with either a structured or an unstructured grid thereby easily enforcing boundary conditions on the fluid–solid interface for stationary solids with smooth boundaries. However, the mesh generation for a complex-shaped solid is cumbersome. Moreover, for flows involving moving solids, transient re-meshing strategies are required which further increases the computational and algorithmic complexity of the body-fitted grid methods. A different approach that retains most of the favorable properties of structured grids but also provides a high level of flexibility in handling irregular-shaped geometry is the immersed boundary method [17,18]. In this method, the requirement of the grid conforming to a solid boundary is relaxed by using a non-conforming grid, and the effect of a complex object on the flow is considered through proper treatment of the solution variables at the grid cells in the vicinity of the body. This method can tackle flows with complex stationary or moving boundaries with relative ease. However, as the solid boundary can arbitrarily cut through the underlying mesh, one needs to treat the boundary in a way that does not adversely impact the accuracy and conservation property of the underlying solver.

Based on the representation of the fluid–solid interface, the immersed boundary methods may be classified as either diffused or sharp interface [18]. In diffused interface methods, an immersed boundary is smeared by distributing singular forces to the surrounding background grid nodes using discrete delta functions [19] or mask functions for penalty methods [20]. The diffused interface methods can be formulated independent of the spatial discretization, and therefore can be easily implemented in an existing fluid solver. However, they produce a “diffused” boundary, and the boundary conditions on an immersed surface are not precisely satisfied at its actual location but within a localized region around the boundary. The so called “sharp interface” methods include, to name a few, the ghost-cell [21–23], the cut-cell [24,25] and the immersed interface methods [26], which strongly depend upon the spatial discretization of the immersed boundary and in which a solid boundary is precisely tracked. The sharp interface methods are preferred because of accuracy, particularly for flows with thin boundary layers. The ghost-cell methods are considered to be less accurate than the cut-cell methods at the same resolution of an underlying Cartesian grid due to its inherent implicit representation of the solid boundary. However, they can be easily implemented and are computationally efficient as it is not necessary to modify flux calculations of an existing Cartesian-grid solver. Moreover, the complicated cell reshaping procedure required in a cut-cell method is not needed in a ghost-cell method.

A critical issue in a ghost-cell immersed boundary method is the accuracy of the reconstruction solution at nodes near the immersed interface via appropriate interpolation schemes using known values on the solid surface and the information from the interior of the flow. The accuracy of the interpolation/extrapolation is an important aspect of an immersed boundary method since it directly influences the number of computational cells required to resolve a flow field as economically as possible. A classical scheme computing the ghost-cell values is the bilinear interpolation for two-dimensional (2-D) problems [21] (trilinear interpolation for 3-D problems [22]). However, when the interpolation point is very close to the boundary, all neighboring points required for the interpolation may not be in the fluid domain. In such cases, the information at the desired point can be found either by using a reduced-order interpolation scheme or by employing body-intersecting points for the interpolation. The inverse distance weighting interpolation method has also been used to construct the fluid values in sharp interface immersed boundary methods [21,27,28]. This scheme is stable for reconstructing variables that smoothly vary without exhibiting large maximum values. Toja-Silva [29] developed an immersed boundary method based on radial basis functions for the interpolation of the near-boundary cells that had convergence rate of one. Note that the accuracy of the above-mentioned methods is at most second order. Interpolations based on higher-order polynomials are expected to be more accurate, but they often lead to numerical instabilities and the determination of appropriate stencils for such interpolations is very difficult. Seo and Mittal [30] applied a moving least-squares (MLS)
method based on a high-order approximating polynomial to construct the flow values near immersed boundaries. Their results showed that the convergence rate of the method is about 5 (3) for very fine (coarse) grids.

The above brief review indicates that one way to simulate the interaction of a high-speed compressible viscous flow with irregularly shaped solids is to combine an accurate shock-capturing WENO scheme with the sharp interface immersed boundary method. Chaudhuri et al. [28] and Pasquariello et al. [31] used WENO-based sharp interface immersed boundary methods to analyze an inviscid flow interacting with irregular-shaped solids. However, we have not found a WENO scheme used for analyzing compressible viscous flows involving irregular-shaped moving solids. In addition, for simulating a high-speed viscous flow around a moving solid, the correct information about surface tractions and boundaries is important. This requires an accurate boundary formulation to minimize the dispersion/dissipation errors at the boundary layer. Though the MLS reconstruction scheme enables higher-order boundary formulations while allowing for a high degree of flexibility with respect to the interpolation stencil, this method can be numerically unstable [30,32]. These considerations have motivated us to propose a robust ghost-cell immersed boundary method for solving the fluid–solid interaction problems for high-speed compressible viscous flows interacting with moving bodies. In order to efficiently capture flow discontinuities, we employ a spatial flux discretization for the inviscid fluxes using the fifth-order WENO scheme together with a Lax–Friedrichs flux function. A conservative fourth-order central-difference scheme is adopted to discretize viscous terms in the compressible Navier–Stokes equations. A constrained moving least-squares (CMLS) sharp interface method is proposed to enforce boundary conditions on the fluid–solid interface. The Robin-type boundary conditions are considered to unify the Dirichlet- and the Neumann-type boundary conditions on the fluid–solid interface which are satisfied by using a penalty term in the constrained MLS formulation. The proposed technique preserves all advantages of the conventional MLS method, and provides an alternative and better approach devoid of numerical instabilities. The solution of the fluid flow and the solid motion equations is advanced in time by using, respectively, the third-order Runge–Kutta and the implicit Newmark integration schemes. A strong coupling between the fluid and the structural solutions is achieved iteratively within each time step until the prescribed convergence criteria have been satisfied. Results for four test problems stated in the Abstract are presented and compared with data available in the literature to establish the robustness and the accuracy of the proposed method.

The remainder of the paper is organized as follows. Equations governing the compressible viscous flow and motion of the solid are described in Section 2. Numerical methods, including the discretization of the Navier–Stokes equations and the constrained MLS ghost-cell immersed boundary method are presented in Section 3. Numerical results for the example problems are presented and discussed in Section 4 to demonstrate the accuracy and the robustness of the present method. Finally, conclusions are summarized in Section 5.

2. Problem formulation

2.1. Problem description

Let $\Omega \subset \mathbb{R}^2$ be a two-dimensional region with $\Omega = \Omega_\text{s} \cup \Omega_f$, where at time $t$, the region $\Omega_\text{s}$ is occupied by a rigid solid and $\Omega_f$ by a fluid as depicted in Fig. 1. The fluid and the solid intersect at a common immersed boundary, $\Gamma = \partial \Omega_f \cap \partial \Omega_\text{s}$. The unit normal vector $\mathbf{n} = \mathbf{n}(\mathbf{x}_f, t)$ points from the solid into the fluid domain, and $\mathbf{x}_f = \mathbf{x}_f(t)$ denotes a material point on the surface of the solid.

2.2. Governing equations for the fluid

The transient motion of the compressible viscous fluid is governed by Navier–Stokes equations written in conservation form as

$$\begin{align*}
\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x_1} + \frac{\partial \mathbf{F}}{\partial x_2} = \frac{1}{\text{Re}} \left( \frac{\partial \mathbf{G}}{\partial x_1} + \frac{\partial \mathbf{H}}{\partial x_2} \right)
\end{align*}$$

in which

\[\begin{align*}
\mathbf{Q} &= \begin{pmatrix} 
\rho \\
\rho \mathbf{u} \\
\rho e 
\end{pmatrix}, \\
\mathbf{E} &= \begin{pmatrix} 
\rho u \\
\rho u^2 + \frac{1}{2} \rho \mathbf{u}^2 \\
\rho u e + \rho \mathbf{u} \cdot \mathbf{E} 
\end{pmatrix}, \\
\mathbf{F} &= \begin{pmatrix} 
0 \\
\rho u v + \rho \mathbf{u} \cdot \mathbf{E} \\
\rho u e + \rho \mathbf{u} \cdot \mathbf{E} 
\end{pmatrix}, \\
\mathbf{G} &= \begin{pmatrix} 
\rho \frac{\partial u}{\partial x_1} \\
\rho u \frac{\partial u}{\partial x_1} + \frac{1}{2} \rho \frac{\partial \mathbf{u}^2}{\partial x_1} + \rho \mathbf{u} \cdot \mathbf{E} \\
\rho u \frac{\partial u}{\partial x_1} + \rho \mathbf{u} \cdot \mathbf{E} + \rho \mathbf{u} \cdot \mathbf{E} 
\end{pmatrix}, \\
\mathbf{H} &= \begin{pmatrix} 
\rho \frac{\partial v}{\partial x_2} \\
\rho u v + \rho \mathbf{u} \cdot \mathbf{E} \\
\rho u v + \rho \mathbf{u} \cdot \mathbf{E} 
\end{pmatrix}.
\end{align*}\]
\[ \mathbf{Q} = \begin{bmatrix} \rho \\
 \rho u_1 \\
 \rho u_2 \\
 E \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \rho u_1 \\
 \rho u_1^2 + p \\
 \rho u_1 u_2 \\
 (E + p)u_1 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u_2 \\
 \rho u_1 u_2 \\
 \rho u_2^2 + p \\
 (E + p)u_2 \end{bmatrix} \]
\[ \mathbf{G} = \begin{bmatrix} 0 \\
 \sigma_{11} \\
 \sigma_{12} \\
 \sigma_{11} u_1 + \sigma_{12} u_2 - q_1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 0 \\
 \sigma_{12} \\
 \sigma_{22} \\
 \sigma_{12} u_1 + \sigma_{22} u_2 - q_2 \end{bmatrix} \]

In Eqs. (1) and (2) \( \mathbf{Q} \) is the vector of conserved variables, \( \rho, p \) and \( E \) denote, respectively, the mass density, the pressure, and the total energy per unit present volume of the fluid, \( u_i \) is the velocity of a fluid particle in the coordinate direction \( x_i \), and \( E \) is related to the specific (i.e., per unit mass) internal energy \( e \) and the velocity \( u_i \) as: \( E = \rho e + \rho u_i u_i / 2 \). \( \mathbf{E} \) and \( \mathbf{F} \) are the inviscid flux vectors in the \( x_1 \) and \( x_2 \) directions, respectively, and \( \mathbf{G} \) and \( \mathbf{H} \) are the viscous flux vectors. The Reynolds number, \( \text{Re} \), based on the free stream flow properties is given by, \( \text{Re} = \rho_{\infty} u_{\infty} L / \mu_{\infty} \), where \( \rho_{\infty}, u_{\infty} \) and \( \mu_{\infty} \) are, respectively, the free-stream mass density, the velocity, and the dynamic viscosity, and \( L \) is the characteristic length. \( \sigma_{ij} \) and \( q_i \) are viscous stresses and heat fluxes, respectively.

We assume that the Newtonian fluid has zero bulk viscosity. Thus, the viscous stress tensor is written as:
\[ \sigma_{ij} = 2 \mu \left( S_{ij} - \frac{1}{3} S_{kk} \delta_{ij} \right) \]  
where the strain rate tensor \( S_{ij} \) is defined as: \( S_{ij} = (\partial u_i / \partial x_j + \partial u_j / \partial x_i) / 2 \). The non-dimensional dynamic viscosity coefficient, \( \mu \), is assumed to depend on the non-dimensional temperature, \( T \), by Sutherland’s empirical equation:
\[ \mu = \frac{1 + C / T_{\infty}}{T / T_{\infty}} \left( T_{\infty} \right)^{\frac{1}{2}} \]  
with \( C = 110.5 \) K and \( T_{\infty} \) the non-dimensional free stream temperature. Equation (4) is valid at moderate temperatures.

The heat flux \( q_i \) in Eq. (2) is expressed as:
\[ q_i = -\frac{\mu}{(\gamma - 1) M_{\infty}^2 \text{Pr}} \frac{\partial T}{\partial x_i} \]  
where \( M_{\infty} \) is the free stream Mach number, \( \text{Pr} \) is the Prandtl number, and \( \gamma \) is the specific heat ratio. Equations (1) through (5) are supplemented by the following equation of state for the fluid.
\[ p = (\gamma - 1) \rho e, \quad T = \frac{\gamma M_{\infty}^2 p}{\rho} \]

### 2.3. Rigid body dynamics

The translational and the rotational motions of the rigid solid satisfy, respectively, the following Newton’s equations
\[ \frac{d\mathbf{X}}{dt} = \mathbf{U}, \quad M \frac{d\mathbf{U}}{dt} = \mathbf{F}_s + \mathbf{F}_f \]  
and the Euler equations
\[ \frac{d\mathbf{\Theta}}{dt} = \mathbf{\omega}, \quad J \frac{d\mathbf{\omega}}{dt} = \mathbf{T}_s + \mathbf{T}_f \]
where \( \mathbf{X} \) is the translational displacement of the mass center of the solid, and \( \mathbf{\Theta} \) the angular displacement measured from a body-fixed reference line. \( \mathbf{U} \) and \( \mathbf{\omega} \) are the translational and the angular velocity vectors, respectively, \( M \) and \( J \) are the mass and the principal moment of inertia of the solid, respectively. \( \mathbf{F}_s \) and \( \mathbf{T}_s \) are the external force and the torque acting on the solid that are not associated with the fluid motion such as the mechanical and the gravitational forces. The resultant force and the resultant torque imparted by the fluid to the solid are denoted, respectively, by \( \mathbf{F}_f \) and \( \mathbf{T}_f \).

### 2.4. Coupling conditions on the fluid–solid interface

For the viscous fluid and the rigid solid considered here, the following no-slip conditions hold on their interface:
\[ \mathbf{u}(\mathbf{X}_f(t), t) = \mathbf{U}_c + \mathbf{\omega} \times (\mathbf{X}_f(t) - \mathbf{X}_t(t)) \]
where \( \mathbf{u} \) is the velocity vector of a fluid particle located at the position \( \mathbf{X}_f(t) \) on the fluid–solid interface. \( \mathbf{U}_c \) and \( \mathbf{X}_t \) denote, respectively, the velocity and the position vector of the mass center of the solid. In the inviscid case, the no-slip condition
is replaced by the requirement that the flow be tangent to the body, and Eq. (9) by \( \mathbf{u} \cdot \mathbf{n} = \mathbf{U}_k \cdot \mathbf{n} + [\omega \times (\mathbf{X}_r - \tilde{\mathbf{X}}_r)] \cdot \mathbf{n} \) that implies the continuity of the normal components of velocity of the contacting fluid and the solid particles; e.g., see [44].

The resultant force and torque exerted by the fluid on the solid are given by

\[
\mathbf{F}_f = \int \mathbf{f}_f dS = \int (-p \mathbf{I} + \sigma) \mathbf{n} d\Gamma \\
\mathbf{T}_f = \int [\mathbf{X}_r(t) - \tilde{\mathbf{X}}_r(t)] \times \mathbf{f}_f d\Gamma
\]

where \( \mathbf{f}_f \) represents the force per unit area exerted upon the solid surface by the fluid. For an inviscid fluid, \( \sigma = 0 \) in Eq. (10).

Solids are considered as non-conductors of heat and the energy equation is not solved in the solid interior.

3. Fluid numerical methods

3.1. Fluid field discretization

The Navier–Stokes equations of the compressible fluid are discretized on a uniform Cartesian grid. The semi-discrete form of Eq. (1) is written as

\[
\frac{d\mathbf{Q}_{i,j}}{dt} + \frac{\mathbf{E}_{i+1/2,j} - \mathbf{E}_{i-1/2,j}}{\Delta x_1} + \frac{\mathbf{F}_{i,j+1/2} - \mathbf{F}_{i,j-1/2}}{\Delta x_2} = \frac{1}{\text{Re}} \left( \frac{\mathbf{G}_{i+1/2,j} - \mathbf{G}_{i-1/2,j}}{\Delta x_1} + \frac{\mathbf{H}_{i,j+1/2} - \mathbf{H}_{i,j-1/2}}{\Delta x_2} \right)
\]

where \( \mathbf{Q}_{i,j} \) is the vector of conserved variables at the \((i, j)\)th grid node \((i, j)\) are the grid indices, \(\Delta x_i\) is the grid spacing in \(x_i\) direction, \(\mathbf{E}_{i\pm1/2,j}\) are the inviscid flux vectors at the right and the left cell boundaries, and \(\mathbf{F}_{i,j\pm1/2}\) are the inviscid flux vectors at the top and the bottom cell boundaries. To discretize the viscous flux vectors in the Navier–Stokes equations, a conservative central-difference scheme is adopted. \(\mathbf{G}_{i\pm1/2,j}\) and \(\mathbf{H}_{i,j\pm1/2}\) are interpolated viscous flux vectors at cell boundaries whose details will be discussed below.

The flux vectors \(\mathbf{E}_{i\pm1/2,j}\) and \(\mathbf{F}_{i,j\pm1/2}\) are split into positive and negative flux vectors as

\[
\mathbf{E}_{i\pm1/2,j} = \mathbf{E}_{i\pm1/2,j}^+ + \mathbf{E}_{i\pm1/2,j}^-,
\]

\[
\mathbf{F}_{i,j\pm1/2} = \mathbf{F}_{i,j\pm1/2}^+ + \mathbf{F}_{i,j\pm1/2}^-
\]

The Lax–Friedrichs flux splitting method [33] is employed to calculate the numerical flux vectors \(\mathbf{E}^\pm\) and \(\mathbf{F}^\pm\):

\[
\mathbf{E}^\pm = \frac{1}{2} \left( \mathbf{E} \pm \mathbf{R} \mathbf{A} (\mathbf{L} \mathbf{Q}) \right),
\]

\[
\mathbf{F}^\pm = \frac{1}{2} \left( \mathbf{F} \pm \mathbf{T} \tilde{\mathbf{A}} (\mathbf{S} \mathbf{Q}) \right)
\]

where \(\mathbf{A}\) and \(\tilde{\mathbf{A}}\) are diagonal matrices of real eigenvalues of matrices \(\partial \mathbf{E} / \partial \mathbf{Q}\) and \(\partial \mathbf{F} / \partial \mathbf{Q}\), respectively. The columns of \(\mathbf{R}\) and \(\mathbf{T}\) (\(\mathbf{L}\) and \(\mathbf{S}\)) are matrices of the right (left) eigenvectors of \(\partial \mathbf{E} / \partial \mathbf{Q}\) and \(\partial \mathbf{F} / \partial \mathbf{Q}\), respectively. Expressions of these matrices are given in Ref. [33]. The calculation of \(\mathbf{E}^\pm\) and \(\mathbf{F}^\pm\) requires interpolated values of the flux vectors \(\mathbf{E}\) and \(\mathbf{F}\), and the conserved variable \(\mathbf{Q}\) at the interfaces, which are determined by using the fifth-order WENO scheme [8,9].

A 1-D scalar problem is considered here to describe the fifth-order WENO scheme, and a generic variable \(\varphi\) is used to represent either the variable or the flux. The fifth-order WENO reconstruction for \(\varphi_{i+1/2}^k\) in the \(x\)-direction is implemented through the convex combination of interpolated values \(\hat{\varphi}_{i+1/2}^k\) \((k = 0, 1, 2)\), given as:

\[
\varphi_{i+1/2}^k = \sum_{k=0}^2 \alpha_k \hat{\varphi}_{i+1/2}^k
\]

in which

\[
\hat{\varphi}_{i+1/2}^0 = \frac{2\varphi_i^+ + 5\varphi_{i+1}^+ - \varphi_{i+2}^+}{6}, \quad \hat{\varphi}_{i+1/2}^1 = \frac{-\varphi_i^+ + 5\varphi_{i+1}^+ + 2\varphi_{i+2}^+}{6}, \quad \hat{\varphi}_{i+1/2}^2 = \frac{2\varphi_i^+ - 7\varphi_{i+1}^+ + 11\varphi_{i+2}^+}{6}
\]

The weight \(\alpha_k\) in Eq. (15) is defined by

\[
\alpha_k = \frac{d_k}{\sum_{j=0}^2 \alpha_j}, \quad \alpha_k = \frac{d_k}{(\varepsilon + \beta_k)^\nu}
\]

where \(d_k\) is taken as 0.3, 0.6 and 0.1 for \(k = 1, 2\) and 3, respectively. The parameter \(\nu\) is set equal to 2 to achieve fast convergence in non-smooth flow regions, the parameter \(\varepsilon = 10^{-6}\) is assigned a small value to prevent the denominators from becoming zero, and the smoothness indicators, \(\beta_k\), are defined as:
\[
\beta_0 = \frac{13}{12} \left( \varphi_i - 2\varphi_{i+1} + \varphi_{i+2} \right)^2 + \frac{1}{4} \left( 3\varphi_i - 4\varphi_{i+1} + \varphi_{i+2} \right)^2
\]
\[
\beta_1 = \frac{13}{12} \left( \varphi_{i-1} - 2\varphi_i + \varphi_{i+1} \right)^2 + \frac{1}{4} \left( \varphi_{i-1} - \varphi_{i+1} \right)^2
\]
\[
\beta_2 = \frac{13}{12} \left( \varphi_{i-2} - 2\varphi_{i-1} + \varphi_i \right)^2 + \frac{1}{4} \left( \varphi_{i-2} - 4\varphi_{i-1} + 3\varphi_i \right)^2
\]

Since linear combinations in Eq. (15) define the numerical flux at a cell face of the domain, an index-shift by \(-1\) gives \(\varphi_j^{n+1/2}\). Multidimensional problems are similarly analyzed.

To discretize viscous terms in the Navier–Stokes equations, the stencil width related to the viscous flux discretization should not exceed the WENO stencil width. Otherwise, the advantage of the WENO scheme in achieving a high-order interpolation within a compact stencil is lost. If the dynamic viscosity coefficient \(\mu\) in Eq. (3) is constant, achieving a finite differencing scheme with both high-order accuracy and flux conservation is straightforward. This is not the case in the present analysis since \(\mu\) varies with the temperature. Here, we employ a conservative fourth-order accurate central differencing scheme [34] for discretization of the viscous flux vectors, and construct viscous flux vectors \(\boldsymbol{G}\) and \(\boldsymbol{H}\) introduced in Eqs. (2) and (12) as:

\[
\tilde{G}_{i-1/2,j} = \sum_{r=i-3/2}^{i+1/2} \xi_r G_{r,j}, \quad \tilde{H}_{i,j-1/2} = \sum_{r=j-3/2}^{j+1/2} \xi_r H_{r,j}
\]

The coefficients \(\xi_r\) are defined as: \(\xi_{i-3/2} = \xi_{j-3/2} = -1/24, \xi_{i-1/2} = \xi_{j-1/2} = 26/24\) and \(\xi_{i+1/2} = \xi_{j+1/2} = -1/24\).

The viscous flux derivative in the \(x_1\)-direction is taken as an example to describe the discretization scheme. In order to achieve the highest order accuracy of \(G_{r,j}\) \((r = i-3/2, r = i-1/2\) and \(r = i+1/2\) within the stencil width of the fifth-order WENO scheme, the approximation of a generic variable \(\varphi\) in \(G_{r,j}\) is expressed as:

\[
\varphi_{r,j} = \sum_{l=m}^{n} \xi_l^r \varphi_{i+l,j}
\]

with coefficients \(\xi_l^r\) defined in Table 1.

The first-order derivative of \(\varphi\) in \(G_{r,j}\) is approximated as:

\[
\frac{\partial \varphi}{\partial x_1} |_{r,j} = \frac{1}{\Delta x_1} \sum_{l=-3}^{2} \xi_l^r \varphi_{i+l,j}, \quad \frac{\partial \varphi}{\partial x_2} |_{r,j} = \sum_{l=-2}^{1} \xi_l^r \frac{\partial \varphi}{\partial x_2} |_{i+l,j}
\]

in which

\[
\frac{\partial \varphi}{\partial x_2} |_{i,j} = \frac{1}{\Delta x_2} \sum_{l=-2}^{2} \xi_l^r \varphi_{i+l,j}
\]

and coefficients \(\xi_l^r\) and \(\xi_l^r\) are found from the Taylor series expansion. Values of \(\xi_l^r\) are listed in Table 2, and \(\xi_{-2} = 1/12, \xi_{-1} = -8/12, \xi_0 = 0, \xi_1 = 8/12, \xi_2 = -1/12\).

The third-order TVD Runge–Kutta scheme is employed for the time integration of Eq. (12).

### 3.2. Constrained MLS immersed boundary method

The ghost-cell MLS immersed boundary method is employed to enforce the interface continuity conditions on the fluid–solid interface. The essence of this method is to compute flow variables at several layers of cells inside and adjacent to the
immersed boundary (the so called “ghost cells”) such that boundary conditions on the immersed boundary are satisfied. The irregular-shaped immersed solid moving on a fixed Cartesian grid is tracked using the Lagrangian description of motion for them employing a series of body-markers, and linear segments are extended between neighboring body-markers to discretize the immersed boundary, as illustrated in Fig. 2. Computational grid nodes outside and inside of the immersed body are identified as fluid and solid nodes, respectively. A solid node next to the fluid-solid interface included in the finite-difference stencil is termed as a ‘ghost point’ (GP). The overall approach now is to assign appropriate values of the fluid variables at these GPs, which implicitly satisfy boundary conditions on the immersed boundary. In order to accomplish this, we extend a line segment from each GP in the fluid to an “image point” (denoted by IP) such that it is normal to the immersed boundary and the boundary intercept (denoted by BI) is midway between the GP and the IP. Once the BI and the IP of each GP have been identified, values of the fluid variables at the IP may be determined using solutions of the fluid field equations. Subsequently, an extrapolation scheme is used to obtain the value for the GP using the information of the IP and the boundary conditions at the BI.

Recalling Eq. (9), the no-slip condition requires the fluid velocity to equal the local solid body velocity; this is categorized as the Dirichlet-type boundary condition. Following Udaykumar et al. [24] a Neumann boundary condition for the pressure is obtained from the momentum equation by projecting the pressure gradient onto the surface-normal direction and dropping the viscous terms, which results in

\[
\frac{\partial p}{\partial n} \bigg|_{\Gamma(t)} \approx -\left(\rho \frac{\partial u}{\partial t} \cdot n \right) \bigg|_{\Gamma(t)}
\]  

(25)

where D/\(\partial t\) represents the material time derivative. The material time derivative of the velocity is then approximated from the known boundary velocities. This obviates the approximation of the convective term.

We assume the solid surface to be fully insulated, i.e., the heat flux across it is zero:

\[
\frac{\partial T}{\partial n} \bigg|_{\Gamma(t)} = 0
\]  

(26)

Both Eqs. (25) and (26) are categorized as Neumann-type boundary conditions. For a generic variable \(\varphi\), the Dirichlet- and the Neumann-type boundary conditions can be written in a unified form, i.e., the Robin-type boundary condition

\[
\alpha \varphi + \chi \frac{\partial \varphi}{\partial n} = \hat{\gamma}
\]  

(27)

where \(\hat{\gamma}\) is a given function; \(\alpha = 1, \ \chi = 0\) for the Dirichlet-type boundary condition, and \(\alpha = 0, \ \chi = 1\) for the Neumann-type boundary condition.

A CMLS method is proposed to approximate the value of \(\varphi\) in a local support region \(\Omega_l\) surrounding an IP; see Fig. 2. A local coordinate system \((\xi_1, \xi_2)\) is introduced at the IP, and the position of an arbitrary field point \(x \in \Omega_l\) in the local coordinate system is related to its position in the global coordinate system by \(\xi = x - x_\text{IP}\), where \(x_\text{IP}\) denotes the location of the IP. In the local coordinate system, the distribution of \(\varphi(\xi)\) over a number of fluid nodes \(\{\xi_i\} (i = 1, 2, \ldots, N, \ \xi_i \in \Omega_l)\), may be expressed as

\[
\varphi^h(\xi) = \sum_{i=1}^{m} \mathcal{G}_i(\xi) \eta_i(\xi) = \mathbf{G}^T(\xi) \eta(\xi), \quad \forall \xi \in \Omega_l
\]  

(28)
where \( \phi^h(\xi) \) is the approximant of \( \phi(\xi) \). \( \mathbf{G}_i(\xi) \) and \( \eta_i(\xi) \) are \( m \) basis functions and unknown coefficients, respectively. We write \( \mathbf{G}(\xi) = [\mathbf{G}_1(\xi), \ldots, \mathbf{G}_m(\xi)]^T \) and \( \eta(\xi) = [\eta_1(\xi), \ldots, \eta_m(\xi)]^T \).

For the two-dimensional problems considered here, the following basis functions can be used:

Linear: \[ \mathbf{G}^T(\xi) = [1, \xi_1, \xi_2] \] (29)

Quadratic (incomplete): \[ \mathbf{G}^T(\xi) = [1, \xi_1, \xi_2, \xi_1 \xi_2] \] (30)

Quadratic (complete): \[ \mathbf{G}^T(\xi) = [1, \xi_1, \xi_2, \xi_1^2, \xi_1 \xi_2, \xi_2^2] \] (31)

Cubic: \[ \mathbf{G}^T(\xi) = [1, \xi_1, \xi_2, \xi_1^2, \xi_1 \xi_2, \xi_2^2, \xi_1^3, \xi_1^2 \xi_2, \xi_1 \xi_2^2, \xi_2^3] \] (32)

The approximation of the value of \( \phi \) at an IP is affected by the field solutions of the fluid domain and the boundary conditions on the fluid–solid interface, the relative importance of which depends upon the distance \( \delta = |x_{IP} - x_{BI}| \) between the IP and the boundary intercept \( x_{BI} \). More specifically, if the IP is closer to the immersed boundary, the boundary conditions may play a more important role in the approximation of the values at the IP. On the other hand, if the distance \( \delta \) is relatively large compared to the grid spacing \( \Delta x \), the approximation of \( \phi \) at the IP is dominated by the solution of the field equations rather than the boundary conditions on the immersed interface. Substituting Eq. (28) into Eq. (27), the Robin-type boundary condition at the boundary intercept \( x_{BI} \) may be expressed as:

\[
\mathbf{G}^T(\xi) \eta(\xi) = \gamma
\]

which is subjected to the constraint defined in Eq. (33) since the boundary conditions on the fluid–solid interface may affect the approximation of \( \phi, \phi_i \) are values of the variables at the fluid nodes \( x_i \) in the support domain \( \Omega_i \).

In the local support region \( \Omega_i \) surrounding an IP, to achieve the best approximation of \( \phi \) in a least-squares sense, the coefficient vector \( \eta(\xi) \) in Eq. (28) is selected to minimize the following functional:

\[
H(\eta) = \sum_{i=1}^{N} \mathcal{W}_i(\xi - \xi_i)[\mathbf{G}^T(\xi) \eta(\xi) - \gamma]^2
\]

which is subjected to the constraint defined in Eq. (33) since the boundary conditions on the fluid–solid interface may affect the approximation of \( \phi, \phi_i \) are values of the variables at the fluid nodes \( x_i \) in the support domain \( \Omega_i \).

Minimizing \( H(\eta) \) under the constraint expressed by Eq. (33) is solved by using the penalty method. In doing so, the following modified functional is constructed:

\[
\tilde{H}(\eta) = \sum_{i=1}^{N} \mathcal{W}_i(\xi - \xi_i)[\mathbf{G}^T(\xi) \eta(\xi) - \gamma]^2 + \kappa [\mathbf{G}^T(\xi) \eta(\xi) - \gamma]^2
\]

where \( \kappa \) is a preassigned penalty parameter.

Minimizing \( \tilde{H}(\eta) \) with respect to \( \eta(\xi) \) leads to the following set of equations:

\[
\mathcal{A}(\xi) \eta(\xi) = \mathcal{B}(\xi) + \mathcal{C}
\]

where \( \mathcal{A}(\xi) = \mathbf{G}^T \mathcal{W} \mathbf{G} + \kappa \mathbf{G}^T \mathbf{G} \), \( \mathcal{B}(\xi) = \mathbf{G}^T \mathcal{W} \phi \), \( \mathcal{C} = \kappa \gamma \mathbf{G}^T \).

Once \( \eta(\xi) \) is determined from Eq. (37), one substitutes \( \eta(\xi) \) into Eq. (28) and obtains the following relation:

\[
\phi^h(\xi) = \mathbf{G}^T(\xi) \mathcal{A}^{-1}(\xi) \mathcal{B}(\xi) + \mathbf{G}^T(\xi) \mathcal{A}^{-1}(\xi) \mathcal{C}
\]

After having found values of the flow variables at the IP, values of the variables at the corresponding GP are determined by using the boundary conditions on the immersed boundary. A linear interpolation along the normal is employed and the value of the primitive variable at the GP is given by [22]:

\[
\varphi_{GP} = \varphi_{IP} + Q_{BI}
\]
where $\varphi_{\text{GP}}$ and $\varphi_{\text{IP}}$ are the flow variables at the GP and the IP, respectively, $Q_{\text{BI}}$ is evaluated at the boundary intercept, and $E$ is a parameter. For the Dirichlet boundary condition, $Q_{\text{BI}} = 2\varphi_{\text{BI}}$ and $E = -1$, and for the Neumann boundary condition, $E = 1$ and $Q_{\text{BI}} = \Delta l(\partial \varphi / \partial n)|_{\text{BI}}$, where $\Delta l$ is the distance between the IP and the BI; see Fig. 2.

**Remark 1.** The CMLS approximation is well defined only when the matrix $A(\zeta)$ in Eq. (37) is non-singular. This holds if the rank of matrix $\mathcal{G}$ equals $m$ and at least $m$ weight functions are non-zero at the point $x_i$. In the traditional MLS method [27,30], the size of the local support domain $\Omega_i$ of any geometrical shape should be large enough to cover a sufficient number of fluid nodes to ensure the regularity of the matrix. However, in the present constrained MLS method, the penalty term will always ensure a regular $A(\zeta)$ as long as the number of fluid nodes in $\Omega_i$ is not less than $m$ and the penalty parameter $\kappa$ is properly chosen. The present method does not suffer from ill-conditioning as may occur in the traditional MLS method. This will be discussed in Section 4.

**Remark 2.** The weight functions $\{W_i(\zeta - \zeta_i)\}$ play important roles in the CMLS approximation. In practice, $W_i(\zeta - \zeta_i)$ is chosen so that it is non-zero on the local support domain $\Omega_i$. In this work, we use a circular region of radius $r_s$ centered at an IP $\zeta_i$ as the local support of $W_i(\zeta - \zeta_i)$; see Fig. 2. In the numerical implementation, the following cubic spline functions [35] are used as weight functions.

$$W_i(\zeta - \zeta_i) = \begin{cases} \frac{3}{2} - 4r_i^2 + 4r_i^3, & \text{if } r_i \leq 0.5 \\ \frac{4}{3} - 4r_i + 4r_i^2 - \frac{4}{3}r_i^3, & \text{if } 0.5 < r_i \leq 1.0 \\ 0, & \text{if } 1.0 < r_i \end{cases}$$

(42)

These cubic splines are monotonically decreasing with the distance from $\zeta_i$ and are smooth in the support domain. $r_i = \|\zeta - \zeta_i\|/r_s$ is the normalized radius, and $r_s = \beta \sqrt{\Delta x_1^2 + \Delta x_2^2}$ is the radius of the local support domain. Here $\beta$ is a scaling parameter.

**Remark 3.** As mentioned previously, the closer (farther) an IP is to the immersed boundary, the more dominant is the effect of the boundary condition (solution of the field equations) on the interpolated values at the IP. Keeping this in mind, we use the CMLS formulation for an IP close to the immersed boundary, and the conventional MLS method (i.e., $\kappa = 0$ in Eq. (40)) otherwise. With $\delta = \|\mathbf{x}_{\text{IP}} - \mathbf{x}_{\text{BI}}\|$, the distance between the IP and the BI, the CMLS approximation is used for $\delta \leq \min(\Delta x_1, \Delta x_2)$.

**Remark 4.** The immersed boundary method is applicable to non-moving solid problems provided that the grid-interface relation and the flow variables at the GPs are updated at each time step. However, the role of a fixed Cartesian grid node near the immersed interface may vary as the solid body moves across a node. As illustrated in Fig. 3, an interior fluid node at time $\mathbf{t}_{n-1}$ may become a newly emerged GP at time $\mathbf{t}_n$ as the boundary intrudes into the fluid region, or a ghost node at $\mathbf{t}_{n-1}$ may become a newly emerged fluid node (called “fresh point”, i.e., FP) at $\mathbf{t}_n$ as the boundary withdraws from the fluid region. For the first case, the solution for the fluid can be advanced in time since all flow variables in the fluid region at time $\mathbf{t}_n$ are well defined. The computation of values at each newly emerged GP follows the same procedure as that for a normal GP. For the second case since the newly emerged fresh nodes were previously in the solid domain and have no history in the fluid phase at the current time step $\mathbf{t}_n$, the flow variables at the fresh nodes are determined by a field-extension procedure [37]. As shown in Fig. 3b, a normal intercept extended from a FP to the boundary intersects it at the boundary intercept (BI). Since the time step size for integration of the fluid flow equations is limited by the Courant–Friedrichs–Lewy (CFL) condition, the moving velocities of the immersed boundary are also subject to a similar constraint. Consequently, at any given time step, the distance between a FP and the extended BI is less than the minimum grid spacing, i.e., $\delta = \|\mathbf{x}_{\text{FP}} - \mathbf{x}_{\text{BI}}\| < \min(\Delta x_1, \Delta x_2)$. The values of flow variables at each FP are interpolated by using the CMLS method, and the interpolation procedure for the FP is similar to that of an IP. The local support domain used for the interpolation of a FP may include a number of FPs (see Fig. 3b) which are excluded in the interpolation process.

**Remark 5.** In the immersed boundary method, the computational grid and the surface of the solid are almost never aligned, introducing complications for computing the resultant forces and torques imparted by the fluid to the solid. For the fluid–solid interaction analysis considered here, these forces and torques are determined at each time step. The computation of local forces on the surface of the solid for incompressible viscous fluids has been discussed by Lai and Peskin [36], and Yang and Balaras [37]. To compute the resultant forces and torque in Eqs. (10) and (11), one needs to calculate

$$f_i = \left( -p \delta_{ij} + 2\mu S_{ij} - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) n_j$$

(43)

where $f_i$ is the surface traction component in the $x_i$ direction.

For finding $f_i$, the knowledge of $p$, $\partial u_i/\partial x_j$, and $\mu$ on the body surface is required. The computation of $\partial u_i/\partial x_j$ at a point on the body surface (called “surface point”) is straightforward. The surface point is treated as an interpolation point.
which is surrounded by a local support domain $\Omega_i$, the CMLS method is applied to approximate the velocity field in $\Omega_i$. Replacing $\varphi$ in Eq. (40) by $u_i$, and differentiating Eq. (40) with respect to $x_j$, the velocity derivatives $\partial u_i / \partial x_j$ are obtained. To compute the pressure, $p$, at a surface point, a simple interpolation from the interior may result in large errors. After testing several strategies it was found that the CMLS interpolation together with Eq. (25) as a constraint provides very good values of $p$. For computing $\mu$ at a surface point, we interpolate the temperature $T$ at the point using the CMLS method, and the adiabatic wall condition, Eq. (26), as the constraint. Subsequently $\mu$ is calculated by using Sutherland’s Eq. (4). The resultant forces and torques on the solid surface are determined by integrating $f_i$ and its moment over all element sides on the surface.

3.3. Strongly coupled iteration algorithm

A strongly coupled serially staggered procedure is applied at each time step to numerically solve equations of the fluid flow and the solid motion. Implicit coupling of these two sets of equations is achieved by a subiteration procedure. During each subiteration the fluid forces in the solid equations are updated and the new surface displacements of the solid are provided to the fluid solver. Using this approach the temporal lag between the fluid and the solid equations can be effectively eliminated, and a complete synchronization of the fluid–solid equation set is achieved. The solution procedure for the entire domain is summarized below.

Assume that the solution for the coupled system is known at time $t_n$. Set $Q^k_n = Q_n$, $X^k_{c,n} = X_n$ and $X^k_{r,n} = X_{r,n}$ for $k = 0$. The solution at $t_{n+1}$ is calculated as follows:

1) Determine fluid nodes, solid nodes, GPs, IPs, BLs, and the newly emerging GPs and FPs due to the boundary movement.
2) Construct values of flow variables at all newly emerging FPs using the CMLS method.
3) Implement the CMLS method to calculate values of flow variables at all IPs, and determine values of variables at the GPs using Eq. (41).
4) Solve the compressible Navier–Stokes equations, and compute $Q^{k+1}_n$ for the fluid.
5) Compute surface force component $f_i$ in Eq. (43) using the CMLS method, and determine the resultant forces $\mathbf{F}_f$ and torque $\mathbf{T}_f$ in Eqs. (10) and (11).
6) Solve Eqs. (7) and (8) for the rigid solid by using the implicit Newmark integration scheme.
7) Update the position, the velocity and the acceleration of the solid.
8) Check convergence of the solution for the solid:

$$\|X^{k+1}_{r,n} - X^k_{r,n}\| < \bar{\varepsilon}$$

where $\| \cdot \|$ is the infinity norm and $\bar{\varepsilon}$ represents the prescribed tolerance, taken here as $10^{-6}$.
9) If converged, set $X_{r,n+1} = X^{k+1}_{r,n}$, $X_{c,n+1} = X^{k+1}_{c,n}$, $Q_{n+1} = Q^{k+1}_n$, and proceed to the next time step. Otherwise, go back to step 1) above.

4. Results and discussion

Solutions of several test problems are discussed in this section to demonstrate the accuracy and robustness of the proposed method. Initially, the flow solver is verified via the shock/boundary-layer interaction problem. Subsequently, supersonic viscous flows of different Mach numbers past a rigid cylinder, and the problem of a moving piston in a shock tube are
studied. The movement of the piston is triggered by a prescribed velocity. Finally, the lift-off movements of circular, rectangular, and elliptical cylinders caused by shock waves are investigated. Results from the present method are compared with either analytical or numerical solutions available in the literature. Unless otherwise stated, monomials defined in Eq. (30) are adopted in the MLS interpolation due to their simplicity. Even though higher-order basis functions (e.g., cubic basis functions) can be employed in the present analysis, they generally require more interpolation nodes [30], and the computational expense associated with the implementation of the boundary conditions on the fluid–solid interface may significantly increase. Furthermore, numerical tests revealed that the complete quadratic basis do not significantly improve the accuracy of the present method but require more interpolation points than the incomplete quadratic basis.

Unless stated otherwise, we set $\kappa = 10^2$ in the CMLS formulation, $\beta = 2.5$ in the radius of the local support, $r_i = \beta \sqrt{\Delta x_i^2 + \Delta y_i^2}$, the Prandtl number, $\text{Pr} = 0.72$, and the specific heat ratio $\gamma = 1.4$. For problems involving viscous flows interacting with rigid solids, the heat flux across the solid surface is assumed to be zero.

4.1. Shock/boundary-layer interaction

The interaction of shock waves with boundary layers has been studied, for example, in Refs. [38–40]. Here, a shock tube of square cross-section with side of unit length and insulated walls is considered. A membrane with a shock Mach number of 2.37 located at the tube center ($x_1 = 0.5$) and separating the two fluids of different states is removed at time $t = 0$ to study their interaction. This is a standard shock tube problem for an inviscid fluid, and would give a 1-D wave structure that can be analyzed by solving a Riemann problem. However, if the compressible Navier–Stokes equations with no slip boundary conditions at the adiabatic walls are considered, there will be complex 2-D shock/boundary-layer interactions that depend on the Reynolds number, $\text{Re}$. The initial states of the fluid are assumed to be: $\rho = 120$, $p = 120/\gamma$, $u_1 = u_2 = 0$ for the fluid on the left-side of the membrane, and $\rho = 1.2$, $p = 1.2/\gamma$, $u_1 = u_2 = 0$ for the fluid on the right-side of the membrane. The reference velocity is based on the initial speed of sound corresponding to the Mach number, $M_\infty = 1$, and we set $\text{Pr} = 0.73$ and $\text{Re} = 200$. The fluid viscosity is assumed to be constant and independent of temperature. When the membrane is ruptured at time $t = 0$, a thin boundary layer at the bottom surface is created during the propagation of the contact and shock waves. The shock wave reflected from the right wall first interacts with the boundary layer near the bottom-right corner and modifies the flow structure near there. The generated recirculating flows initiate the boundary layer detachment and form separated “bubbles”. Since the walls are aligned with the mesh, the conventional ghost node method is employed to impose the boundary conditions along the walls. The contours of the mass density at different times presented in Fig. 4 are very close to those obtained by Daru and Tenaud [39].

The distributions of the computed mass density and the skin friction coefficient along the bottom wall at $t = 1$ are presented in Fig. 5. The skin friction coefficient is defined as: $C_f = \tau / (0.5 \rho_{\infty} U_{\infty}^2)$, where $\tau$ is the shear stress evaluated at the bottom wall. Simulations with increasing grid resolutions have been performed to verify that the computed solution is grid-independent. Daru and Tenaud’s [39] results computed using a high-order monotonicity preserving scheme on a $1000 \times 500$ grid are also included in the figures for comparison. It is observed that the present solutions converge very rapidly with the grid refinement, and results from $800 \times 400$ and $1000 \times 500$ grids are almost identical. Moreover, the mass density

Fig. 4. Density contours of the fluid at different times: top left, $t = 0.4$; top right, $t = 0.6$; bottom left, $t = 0.8$; bottom right, $t = 1.0$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
converges faster than the skin friction coefficient. The present converged solutions of the density distribution and skin friction coefficient along the bottom wall agree well with those reported by Daru and Tenaud [39]. The slight discrepancy in results of the skin friction coefficient between the two methods may be attributed to different finite-difference schemes employed to calculate the shear stress at the bottom wall. We have used a one-sided sixth-order finite-difference scheme to find the shear stress at the wall from the field solutions of the fluid domain.

4.2. Supersonic viscous flow past rigid cylinder

The supersonic viscous flow past a stationary circular cylinder has been analyzed by using the computational domain $30D \times 20D$, where $D$ is the cylinder diameter and the geometric center of the cylinder is located at $[x_1, x_2] = [8D, 20D]$. For $Re = 300$ based on the cylinder diameter and the freestream values, results have been computed for the inflow Mach numbers, $M_\infty = 1.2$ and $M_\infty = 2.0$, no-slip wall boundary conditions on the cylinder surface, and outflow boundary conditions on all other boundaries except for the inlet surface. Results computed using a uniform $600 \times 400$ Cartesian grid are presented in Fig. 6. For both values of $M_\infty$, the freestream flow encounters a bow shock ahead of the cylinder that accelerates along the cylinder surface forming a supersonic flow region which envelops the subsonic recirculation region behind the cylinder and generates two symmetric tail shocks. These density contours agree very well with those found by Takahashi et al. [41] and Qiu et al. [42].

In Fig. 7 we have displayed the pressure coefficient distributions along the surface of the cylinder computed using the two uniform discretizations, $500 \times 330$ and $600 \times 400$, and those found by Takahashi et al. [41] and Qiu et al. [42]. The pressure coefficient is defined as: $C_p = (p - p_\infty)/(0.5\rho_\infty u_\infty^2)$. It is observed that the present results obtained by using the $600 \times 400$ grid compare well with those of Refs. [41] and [42]. This confirms the validity of the present method.
However, requires obtained on $\kappa = \beta$ be generally unconstrained distributions Fig. 8. 

For $M_{\infty} = 2.0$, the $600 \times 400$ uniform mesh and $\kappa = 0, 0.1, 1, 10^2, 10^5$ and $10^8$, results of the pressure coefficient distributions on the cylinder surface are presented in Fig. 8. As mentioned in Section 3, $\kappa = 0$ corresponds to using the unconstrained MLS approach. However, numerical results did not converge for $\kappa = 0$. It is because for $\kappa = 0$, the interpolation of the values at the IPs are based on the field solutions of the fluid domain, and boundary conditions on the fluid–solid interface are not considered in the interpolation process. This may lead to inaccurate interpolated values at the IPs especially when they are very close to the fluid–solid interface. The value of $\kappa$ may significantly affect the solution of the present method. As mentioned in Sec. 3.2, a local coordinate system $(\zeta_1, \zeta_2)$ is employed to describe the fluid field in the local support region $\Omega_l$ surrounding an IP. Values of elements of $\tilde{\mathbf{G}}^\top \mathbf{W} \tilde{\mathbf{G}}$ in $\mathbf{A}(\zeta) = \tilde{\mathbf{G}}^\top \mathbf{W} \tilde{\mathbf{G}} + \kappa \tilde{\mathbf{G}}^\top \mathbf{G}$ (see Eq. (38)) are generally smaller than 1.0. If $\kappa \ll 1.0$, the term $\kappa \tilde{\mathbf{G}}^\top \mathbf{G}$ does not dominate in $\mathbf{A}(\zeta)$, and the boundary conditions may not be appropriately enforced by the penalty method. When $\kappa = 0$ or a very small value, the mass density and the pressure can have negative values at the IPs and the corresponding GPs. This terminates the numerical computation. It is observed from Fig. 8 that the CMLS method is stable for $1.0 \leq \kappa \leq 10^8$. Slight discrepancy exists between the results obtained by using $\kappa = 0.1$ and $\kappa \geq 1$. Here we have set $\kappa = 10^2$ for all problems studied in the paper.

For $M_{\infty} = 2.0$, $600 \times 400$ uniform mesh, and $\beta = 2.0, 3.0, 4.0$ and $6.0$, the computed pressure coefficient distributions on the cylinder surface are exhibited in Fig. 9. It is clear that reasonably accurate values of the pressure coefficient are obtained by using support domains of small radii (e.g., $\beta = 2.0$), and an increase in the radius of the support domain does not significantly affect the pressure distributions on the cylinder surface. As mentioned above, the key to successful implementation of the traditional MLS method is to maintain the well-posedness of the least-squares error problem which requires a relatively large support domain to cover a sufficient number of fluid points for interpolation; see Ref. [30]. However, in the proposed CMLS method, the flow variables near to and on the solid surface are mostly affected by the
4.3. Moving piston in a shock tube

We now analyze the fluid–solid interaction problem of a rigid rectangular piston moving at a constant velocity in a two-dimensional shock tube of length $L = 1$ and width $B = 0.2$. The piston is of length 0.04 and width 0.2, with its center initially located at $x = 0.42$ and $y = 0.1$. The piston moves with a constant speed, $u = 300$, from left to right into an initially quiescent fluid of density $\rho = 1.0$ and pressure $p = 10^5$. For an inviscid fluid, the problem has an analytical solution and it has been discussed in the literature that a fully conservative scheme is essential for accurately predicting the shock formation in front of the piston. The computed results of the mass density and the axial velocity distribution along the $x$-axis at time $t = 8 \times 10^{-4}$ are compared with the exact solution [43] in Fig. 10. The problem is solved using $200 \times 80$, $400 \times 80$, and $800 \times 80$ uniform grids to determine the convergence of the present method. It is observed from results depicted in Fig. 10 that even a relatively coarse $200 \times 80$ grid yields results that are in excellent agreement with the exact solution, and the present approach predicts the correct shock location.

A grid resolution study is performed to analyze the accuracy of the CMLS method. The computed $L_2$ and $L_\infty$ error-norms of the density are illustrated in Fig. 11. The $L_2$ error-norm is a good measure of the global error, and the $L_\infty$ error-norm effectively captures the local error around the immersed boundary. A line representing the second-order accuracy is also included for comparison purpose. The results show that the present method is globally and locally second-order accurate.
4.4.2. 

\[ \text{with}\]

comparable imposed shock \( X_c \).

Fig. 12. Pressure contours at time (top) \( t = 0.140 \) (100 contours from 0 to 28), and (bottom) \( t = 0.255 \) (100 contours from 0 to 22). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4. Lift-off of rigid cylinders

4.4.1. Inviscid fluid

We consider lifting off of a rigid circular cylinder initially at rest on the floor of a rectangular channel by a shock wave of Mach 3. We employ the computational domain \([0, 1] \times [0, 0.2]\) discretized using the 800 \( \times \) 160 uniform grid with the initial shock front positioned at 0.08 from the left boundary. The remaining domain is initially filled with the ideal gas having \( \rho = 1, \ p = 1, \) and \( u_i = 0. \) The cylinder of mass density 7.6 and radius 0.05 is initially located at \((0.15, 0.05)\). In order to compare the present results with those available in the literature, we assume that the fluid is inviscid. The reflective solid wall boundary conditions are applied to the top and the bottom surfaces of the domain, the left boundary is fixed at the post shock state, a zero-gradient outflow boundary condition is applied at the right edge, and slip boundary conditions are imposed on the fluid–solid interface. Fig. 12 shows a snapshot of the pressure contours at time \( t = 0.140 \) and 0.255. The conventional GPs are employed to enforce boundary conditions at the top and the bottom walls, and the CMLS method is used to impose boundary conditions on the fluid–solid interface. Our results compare favorably with those of Monasse et al. [5]. The presently computed final horizontal and vertical positions of the center of cylinder are: \( X_c^1 \approx 0.644 \) and \( X_c^2 \approx 0.146 \), which are very close to \( X_c^1 \approx 0.643 \) and \( X_c^2 \approx 0.147 \) reported by Monasse et al. [5].

Using the same parameters as those for a circular cylinder, we have studied the lifting off of the rigid square cylinder of side length 0.07, mass \( 5.85 \times 10^{-4} \), and the principal moment of inertia \( 4.777 \times 10^{-7} \) initially centered at \((0.15, 0.052)\) with one of its diagonal lines parallel to the bottom wall of the channel. In Fig. 13 are shown snapshots of the pressure contours at \( t = 0.045, 0.15 \) and 0.25. It should be noted that at the corners of the square cylinder, GPs may be situated at comparable distances from two space directions. However, no special treatment is needed in the present analysis, since the storage of the geometrical data of GPs are maintained separately for each space direction.

4.4.2. Viscous fluid

We now study the lifting off of an elliptic cylinder of major and minor radii 0.05 and 0.04, respectively. The cylinder has mass \( 5.85 \times 10^{-4} \), moment of inertia \( 5.996 \times 10^{-7} \) and its center of mass is initially positioned at \((0.125, 0.0515)\) with the major axis perpendicular to the channel axis. We study the viscous flow of \( Re = 100 \) (based on the maximum
dimension of the cylinder and the reference velocity $= 1.0$). The pre-shock conditions are prescribed as $\rho = 1$, $u = v = 0$ and $p = 1$ for $x \geq 0.08$, and the post-shock conditions for $x < 0.08$ as $\rho = 3.857$, $u = 2.629$, $v = 0$, $p = 10.33$. The symmetric boundary conditions are applied to the top and the bottom surfaces of the fluid domain, and the left boundary is at the post shock state and the right boundary has a zero-gradient outflow condition. No slip and adiabatic boundary conditions are applied on the surface of the elliptic cylinder. Time histories of the resultant forces and moment exerted by the fluid on the cylinder computed with $700 \times 140$, $800 \times 160$ and $900 \times 180$ uniform grids are shown in Fig. 14. It is observed that computations using the grids $800 \times 160$ and $900 \times 180$ give nearly identical results. The computational time required by a strongly-coupled partitioned method is more than that for a weakly coupled one since iterations are needed in every time step. The present method converges rapidly, and the implicit coupling of the fluid and the solid equations is achieved in about two subiterations in each time step. Since the computational time required for solving the solid motion equations is negligible, the fluid solver consumes most of the computational time in each subiteration. The relative computational cost of the strongly-coupled algorithm is defined as the ratio of the total computational time dedicated to the coupling method in each time step to the sum of computational time required by the fluid and the solid solvers in each subiteration. This ratio is about 2.0 for the present algorithm. For the $900 \times 180$ grid, we have plotted in Fig. 15 the instantaneous pressure contours at times 0.05, 0.15 and 0.25.

As for the problem studied in subsection 4.2, we have exhibited in Fig. 16 the pressure distributions along the elliptic cylinder surface at $t = 0.25$ computed with the $900 \times 180$ grid. The numerical results did not converge for the unconstrained MLS method (i.e., $\kappa = 0$) but converged for the CMLS method and are stable for $1.0 \leq \kappa \leq 10^8$. Very small discrepancy exists between the results obtained by using $\kappa = 1.0$ and $\kappa \geq 10$. Reasonably accurate results are obtained by using $\kappa = 10^2$.

In Fig. 17 we have exhibited time histories of the resultant forces and moment exerted by the fluid on the cylinder computed for different values of $\kappa$. It is clear that the resultant forces and moment for $\kappa = 0.1$ differ from those computed using $\kappa \geq 1$. Compared with the discrepancies of the resultant forces and moment determined with different grids (see Fig. 14), the deviation of the results due to a small value of $\kappa$ can be significant. It is also observed from Fig. 17 that converged results of the resultant forces and moment can be obtained by the CMLS method for $1.0 \leq \kappa \leq 10^8$. It is not required that $\kappa$ be large enough for the boundary conditions on the fluid–solid interface to be well satisfied, and $\kappa = 10^2$ is conservatively recommended to obtain converged results.

Fig. 18 shows the effect of the support domain size, varied by setting $\beta = 2.0, 3.0, 4.0$ and 6.0, on the resultant forces exerted by the fluid on the elliptic cylinder computed with the $900 \times 180$ uniform grid. It is clear that reasonably accurate values of the resultant forces can be obtained with $\beta = 2.0$, and an increase in its value does not significantly affect the resultant forces. The present method can potentially work for computing flows with small gaps without using local grid refinement. Note that the values of the force in the inset of the left Fig. 17 vary from 0.0310 to 0.0312 and that in the right Fig. 17 from 0.0066 to 0.00665. Thus a very small difference in the value of the force is highly magnified.

We note that for fluid–solid interaction problems involving moving bodies, spurious pressure oscillations may occur for all immersed boundary methods including the flow-reconstruction based methods as well as discrete and distributed forcing methods [25]. It is because for moving boundary problems, the role of some grid cells (or grid points) in the computational domain changes in time leading to spurious mass sources/sinks [25]. It also holds for the present CMLS immersed boundary

![Fig. 13. Pressure contours at time (top), $t = 0.045$ (100 contours from 0 to 42.7), (middle) $t = 0.15$ (100 contours from 0 to 29.7), and (bottom) $t = 0.25$ (100 contours from 0 to 20.8). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](image-url)
Fig. 14. Time histories of the resultant forces and moment exerted by the fluid on the cylinder: (a) resultant force in the x-direction; (b) resultant force in the y-direction; (c) resultant torque. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 15. Contours of the mass density at (top) $t = 0.05$ (100 contours from 0 to 8.11); (middle) $t = 0.15$ (100 contours from 0 to 7.12); $t = 0.25$ (100 contours from 0 to 5.83). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Fig. 16. At time $t = 0.245$, pressure distributions along the elliptic cylinder surface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 17. Time histories of the resultant forces and moment determined by using different values of $\kappa$: (a) resultant force in the $x$-direction; (b) resultant force in the $y$-direction; (c) resultant torque. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

method. However, it is found that the issue can be alleviated by appropriately selecting the computational time step as should be evident from results exhibited in Figs. 14, 16 and 17.

For all problems studied herein, we have not computed the error in satisfying the conservation equations (1) and (2).
5. Conclusions

A robust sharp interface immersed boundary method has been developed to simulate interactions of high-speed compressible viscous flows with complex-shaped moving objects. The fluid flow equations are discretized by the Lax–Friedrichs flux splitting scheme, and the spatial derivatives are approximated by using the fifth-order weighted essentially non-oscillatory finite difference scheme in conjunction with a conservative fourth-order central-difference time integration scheme. The immersed boundary interface is delineated by using a ghost-cell based immersed boundary method. A constrained moving least-squares (CMLS) interpolation method is employed to robustly interpolate the flow variables at an image point corresponding to those at a ghost point, as well as to satisfy the boundary conditions. The present immersed boundary method allows computations for sharp interfaces between the fluid and the solid bodies, preserves advantages of the conventional MLS method, and provides an alternative approach that does not suffer from numerical instabilities encountered in the unconstrained MLS method. The robustness of the method has been demonstrated by solving shock-boundary layer interaction, supersonic viscous flow past a rigid body, piston moving in a shock tube, and lift-off of rigid cylinders triggered by shock waves. The results from the proposed method agree well with their solutions available in the literature. The approach can be extended to studying three-dimensional fluid–solid interaction problems by taking the local support domain around an image point as a sphere.

Acknowledgements

RCB’s work was partially supported by the US Office of Naval Research grant no. N00014-16-1-2309 to Virginia Polytechnic Institute and State University with Dr. Y.D.S. Rajapakse as the program manager. Views expressed in the paper are those of the authors and not of the US ONR.

References


