A Numerical Framework for Fast Transient Compressible Flows Using Lattice Boltzmann and Immersed Boundary Methods

Hippolyte Lerogeron, Pierre Boivin, Vincent Faucher, Julien Favier

International Journal for Numerical Methods in Engineering

WILEY

RESEARCH ARTICLE OPEN ACCESS

A Numerical Framework for Fast Transient Compressible Flows Using Lattice Boltzmann and Immersed Boundary Methods

Hippolyte Lerogeron[1,2] | Pierre Boivin[1] | Vincent Faucher[2] | Julien Favier[1]

[1]Aix Marseille Univ, CNRS, Centrale Med, M2P2Marseille, France | [2]CEA, DES, IRESNE, Nuclear Technology Department, Center of Cadarache, Saint-Paul-lez-Durance, France

Correspondence: Hippolyte Lerogeron (hippolyte.lerogeron@cea.fr)

Received: 7 June 2024 | **Revised:** 8 October 2024 | **Accepted:** 2 December 2024

Funding: This work was performed using the ProLB software. Centre de Calcul Intensif d'Aix-Marseille is acknowledged for granting access to their high-performance computing resources. This project was provided with computer and storage resources by GENCI at TGCC thanks to the grant 2023-A0152A07679 on the supercomputer Joliot Curie SKL/ROME partition. Isabelle Cheylan, Jérôme Jacob and Song Zhao are gratefully acknowledged for the insightful discussion they proposed and the technical expertise they offered.

Keywords: immersed boundary method | Lattice Boltzmann method | thermal boundary condition | transient compressible flow

ABSTRACT

This article is dedicated to the development of a model to simulate fast transient compressible flows on solid structures using immersed boundary method (IBM) and a lattice Boltzmann solver. Ultimately, the proposed model aims at providing an efficient algorithm to simulate strongly-coupled fluid-structure interactions (FSI). Within this goal, it is necessary to propose a precise and robust numerical framework and validate it on stationary solid cases first, which is the scope of the present study. Classical FSI methods, such as body-fitted approaches, are facing challenges with moving or complex geometries in realistic conditions, requiring computationally expensive re-meshing operations. IBM offers an alternative by treating the solid structure geometry independently from the fluid mesh. This study focuses on the extension of the IBM to compressible flows, and a particular attention is given to the enforcement of various thermal boundary conditions. A hybrid approach, combining diffuse forcing for Dirichlet-type boundary conditions and ghost-nodes forcing for Neumann-type boundary conditions is introduced. Finally, a simplified model, relying only on diffuse IBM forcing, is investigated to treat specific cases where the fluid solid interface is considered as adiabatic. The accuracy of the method is validated through various test cases of increasing complexity.

1 | Introduction

The general purpose of the proposed research is set in the context of transient fluid-structure interactions (FSI), with applications across diverse industrial sectors, where predicting the movement or deformation of a solid structure submitted to a fluid flow is crucial. Particularly, there has been interest in the simulation of transient fluid-structure dynamics for efficiency or safety purposes. Typical examples include applications like: the aeroelasticity of slender bodies such as aircraft wings, wind turbine blades suffering severe winds, and energy infrastructures exposed to blast loading.

These systems involve the interaction of a compressible fluid with a deformable solid structure, leading to rapid transient phenomena. The simulation of these systems is specifically interesting

to prevent structure deformations that may lead to severe damage. This is why efforts have been directed towards developing a robust and efficient algorithm in order to enable accurate simulations for a reasonable computational cost.

This article focuses on the first mandatory step towards the general objectives above, dedicated to the design and implementation of a both robust and accurate method to handle immersed structural interfaces within a high-performance compressible fluid solver based on the Lattice Boltzmann Method (LBM). The latter is chosen in particular for its inherent low-diffusive properties and the local character of its algorithm yielding efficient computational properties in parallel architectures.

Imposing boundary conditions directly on the first layer of cells along the solid surface of the conformal grid is straightforward. Moreover, body-fitted mesh offers easy control of the grid resolution near the solid body. In this framework the Arbitrary Lagrangian–Eulerian (ALE) method has been explored to model moving fluid-structure interface [1, 2]. The fluid-solid interface is modelized as a set of Lagrangian nodes, while the fluid domain is represented by Eulerian nodes. As shown in previous work [3–5], the ALE method has proven its capacity to accurately model fast compressible flows involving FSI. However, the ALE approach relies on an update of the fluid domain with either a rezoning of the fluid nodes or a complete remeshing process. These operations exhibit both a significant computational cost and a lack of robustness regarding large structural motion. In contrast, including moving solid structures using the Immersed Boundary Method (IBM) is simple, but the enforcement of the boundary condition is not as obvious. In the IBM framework, the solid structure boundary is represented as a set of discrete point within a fixed cartesian grid representing the fluid mesh. The immersed boundary (IB) can move freely on the Eulerian mesh. As a result, remeshing process is not necessary, which makes it competitive, in terms of computing performance and robustness. Thus, this alternative method is usually preferred in the case of large solid motion or deformation. The use of IBM in industrial applications involving fast transient FSI was demonstrated for instance in [6, 7], both articles exhibiting simulations of implosion/explosion in tank containers.

The immersed boundary method was first introduced by Peskin [8] as he formalized its numerical framework and introduced a method to spread the effect of the immersed structure on the fluid grid. Later, the spreading and interpolation functions, that connect the immersed structure to the fluid grid, were refined to improve accuracy [8–11]. Simultaneously, new definitions of the solid force acting on the fluid grid were proposed to reduce spurious oscillations [12], or to provide direct enforcement of the boundary condition at the interface [13, 14]. Then, the Ghost-Cell or Ghost-Nodes methods [15–19] emerged to handle thin boundary layer in high Reynolds flows, allowing a sharper definition of the fluid-solid interface. A fundamental review of the subject has been carried out by Mittal and Iaccarino [20]. Feng et al. combined an Immersed Boundary with a Lattice Boltzmann solver [21] in order to solve problems related to previously-used boundary conditions in the LBM framework. More recently, Lu et al. [22] introduced an Immersed Boundary Lattice Boltzmann Model based on multiple relaxation times, which reduces numerical boundary slip compared to traditional models. Gsell, D'Ortona, and Favier [23] addressed the viscosity-related errors in the computed Immersed Boundary (IB) force in LBM simulations. They proposed an explicit and viscosity-independent Immersed Boundary scheme that corrects these errors without additional computational time, ensuring the accuracy of the simulations. Recent studies by Abaszadeh et al. [24] and Afra, Delouei, and Tarokh [25] provided advancements in the application of IB-LBM to specialized areas such as radiative heat transfer and non-Newtonian fluid interaction with flexible structure, respectively. More recently, R. Mittal and R. Bhardwaj reviewed the application of IBM to thermal fluid simulations [26].

We propose here to differentiate two main approaches: diffuse or sharp force. This distinction is based on the way the boundary condition is enforced to the fluid near the immersed solid boundary. In the diffuse forcing approach, the boundary condition is spread from the solid interface to the neighboring nodes through a smooth continuous function. In contrast, the sharp forcing approach consist in strictly enforcing the desired boundary conditions on the first layer of fluid nodes inside the immersed boundary. In the following, let us delve into specific details of these approaches within the framework of compressible flows and fast transient FSI.

The first approach, diffuse forcing, is formulated independently of the spatial mesh. Source terms are introduced into the continuous Navier–Stokes equations to model the influence of the solid. These terms are exchanged from the Lagrangian to the Eulerian domain through Interpolation and Spreading operations using a delta function. The diffuse forcing IBM was initially designed for incompressible flows. Its extension to compressible flow is challenging, as density variations and energy equations should be considered. A first attempt at using a diffuse forcing IBM for compressible flow can be found in [27], where the implicit velocity correction model of Wu and Shu [28] is extended to the correction of pressure and temperature. Validation of the method is carried out on various cases at transonic or supersonic regimes, demonstrating the capabilities of the diffuse forcing scheme to deal with compressible flows. Using a gas kinetic flux solver, the same approach was employed by Sun et al. [29]. Though, their approach relies on an implicit scheme and only Dirichlet-type boundary conditions are treated. This limits the choice of the thermal boundary conditions, since only cases where the desired body temperature is known can be handled.

In particular, in the fast transient FSI framework, a common choice for the thermal boundary condition consist in considering the fluid-solid interface as adiabatic. This requires, among other things, imposing a condition, not on the temperature, but on the temperature gradient, thus a Neumann-type boundary condition (NBC).

The extension of diffuse forcing IBM to NBC has been a subject of research for many years, not only in the framework of compressible flow simulations. The first attempts at using diffuse forcing with NBC aimed at solving thermal flow problems were performed by [30–32]. Among the proposed solutions, A. Hosseinjani and Ashrafizadeh [33] distinguished and compared direct and indirect Neumann diffuse forcing approaches. They conclude that the first one is more accurate, but also more expensive in terms of computational time. More recently, Guo et al. [34]

proposed an implicit heat flux correction-based IBM for thermal flows, using the cell's face centers as Eulerian points for the temperature gradient interpolation. Finally, Chen et al. [35] exploited previous research to propose an IBM that may enforce NBC using diffuse forcing in compressible flows. Their approach lies in the indirect forcing category [33].

To the best of the author's knowledge, the latest reported use of diffuse forcing with NBC can be found in [36–38].

Despite aforementioned work, the enforcement of NBC in the diffuse forcing framework still represents a non-negligible CPU cost. For instance, Hosseinjani and Ashrafizadeh [33] reported 2 to 3 more computational time when using diffuse forcing to enforce NBCs. Also, the diffuse nature of this approach does not suit well with the enforcement of gradients, in particular when the solid geometry exhibits sharp edges, resulting in inaccurate results.

To address these issues, R. Boukharfane et al. explored a hybrid method in [39], that involves a diffuse forcing only for Dirichlet boundary condition enforcement, and a sharp forcing IBM to handle Neumann boundary condition, referred to as sharp forcing IBM.

Sharp forcing IBM, consists in strictly enforcing the desired boundary conditions on the first layer of Eulerian nodes inside the immersed boundary. A summary of the main algorithms can be found in [40]. Among the existing sharp forcing approaches, let us focus on the ghost-nodes approach [15, 16, 19, 41–42]. This method implies first identifying the so-called "ghost nodes". Image points are then computed from a normal projection through the immersed boundary. Finally, interpolated variables at the image node are used to settle the value on the ghost nodes that respect the boundary condition. While the ghost-nodes approach offers direct and accurate control over the fluid near the boundary, it highly depends on the spatial discretization of the fluid domain. The strict enforcement of the velocity or temperature value on the ghost nodes may cause spurious oscillations leading to stability issues in the case of a moving boundary [43–46], referred to as the fresh/dead cells issue, contrary to the diffuse forcing approach. The advantages and drawbacks of each approach were discussed by Kang and Hassan [47] or by Qin, Yang, and Li [48]. Let us summarize the main ideas: the diffuse approach is cheaper in terms of computational time and more stable when dealing with moving fluid-solid interfaces. Though, it lacks of accuracy due to its diffuse nature, and it is not well-suited to the enforcement of NBC. On the other hand, the sharp forcing approach, such as the ghost-nodes method, enables a strict enforcement of any desired boundary condition, such as heat flux conditions. Though it may lack of numerical stability on moving or complex geometries. To address the limitations of each approach, hybrid methods combining both diffuse and sharp forcing IBM have been developed. These methods balance the strengths of each technique to address their respective limitations. A detailed example can be found in [39], as they suggested to use diffuse forcing to enforce Dirichlet-type boundary conditions (e.g., no-slip and isothermal) while sharp forcing is employed for Neumann-type boundary condition (in particular, heat-flux condition).

Based on the previously presented works, the present paper proposes a model for transient compressible flow using an immersed boundary method and an LBM solver. Similarly to [39], a diffuse forcing approach will be used to enforce all Dirichlet-type boundary conditions, while a ghost-nodes (GN) forcing will be employed to ensure Neumann-type boundary conditions. The accuracy of this method is assessed on various test cases.

While the coupling of the diffuse forcing to the GN approach enables the enforcement of necessary Neumann-type boundary conditions, it adds more computing steps to the algorithm and an added computational cost is thus expected.

To prevent this added CPU cost, an alternative simplified IBM model is also proposed in the present paper, designed specifically to deal with zero temperature gradient boundary conditions, also known as the adiabatic boundary condition. Indeed, it has been observed that in many cases of interest where fluid-solid heat exchange can be neglected, the strict enforcement of the adiabatic boundary condition is not necessary to ensure a correct simulation. This simplified model has already been used in [37]. In the present paper, it will be referred to as the T-Passive boundary condition. The relevance and accuracy of this approach will be evaluated in various cases.

The article is organized as follows: after a brief introduction in Section 1, the numerical framework is presented in Section 2. All necessary details on the developed immersed boundary method for compressible flow are given, and a simplified adiabatic IBM model is introduced. The accuracy of the model is assessed on various cases in Section 3. Details on the LBM solver adopted in the present work are given in Appendix A.

2 | Numerical Framework

2.1 | Governing Equations

Let us define a fluid domain Ω and a closed surface $\Gamma \subset \Omega$ representing a solid boundary immersed in the fluid domain.

The fluid motion is described by the conservation laws of mass, momentum, and energy expressed in the following form in the whole domain Ω as (i.e., compressible Navier-Stokes equations):

$$\partial_t \mathbf{Q} + \partial_\alpha \mathbf{F}_\alpha^C - \partial_\alpha \mathbf{F}_\alpha^V = \mathbf{S}^{\mathrm{IB}} \tag{1}$$

where an implicit summation is done over the index $\alpha \in \{x, y, z\}$, $\mathbf{Q}$ is the vector of the conservative variable, $\mathbf{F}_\alpha^C$ is the tensor of convective fluxes in the direction α, $\mathbf{F}_\alpha^V$ is the tensor of viscous fluxes in the direction α and $\mathbf{S}^{\mathrm{IB}}$ is the IB source term that represents the influence of the solid obstacle on the fluid. The conservative variables and fluxes write:

$$\mathbf{Q} = [\rho, \rho u_x, \rho u_y, \rho u_z, \rho \mathrm{E}]^T \tag{2}$$

$$\mathbf{F}_\alpha^C = [\rho u_\alpha, \rho u_\alpha u_x + p\delta_{\alpha x}, \rho u_\alpha u_y + p\delta_{\alpha y}, \rho u_\alpha u_z + p\delta_{\alpha z}, (\rho \mathrm{E} + p)u_\alpha]^T \tag{3}$$

$$\mathbf{F}_\alpha^V = [0, \tau_{\alpha x}, \tau_{\alpha y}, \tau_{\alpha z}, \tau_{\alpha\beta} u_\beta + q_\alpha]^T \tag{4}$$

where $\delta_{\alpha\beta}$ is the Kronecker delta. with ρ the density, u_α the α-component of the velocity, T the temperature, p the pressure and E the total energy of the fluid. $\tau_{\alpha\beta}$ is the stress sensor and q_α is the α-component of the conduction heat flux defined as:

$$\tau_{\alpha\beta} = \mu\left[\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} - \delta_{\alpha\beta}\frac{2}{3}\frac{\partial u_\gamma}{\partial x_\gamma}\right] \tag{5}$$

$$q_\alpha = -\lambda\frac{\partial T}{\partial x_\alpha} \tag{6}$$

where λ is the thermal conductivity coefficient and μ is the dynamic viscosity.

An equation of state (EOS) is required to close the system. The proposed IBM does not assume any particular shape for this equation and in the following, the ideal gas EOS is used: $p = \rho rT$ with r the specific gas constant defined as $r = \frac{R}{W}$ where R is the gas constant and W is the molar mass.

The total energy may be divided into kinetic and internal energy as:

$$\mathrm{E} = \frac{1}{2}u_\alpha^2 + C_v T \tag{7}$$

with C_v the specific heat capacity.

Finally, the source term $\mathbf{S}^{\mathrm{IB}}$ is related to the IBM, and will be described in Section 2.3.

2.2 | Hybrid LBM Solver

This section is dedicated to providing a concise description of the lattice Boltzmann solver used to solve (1) over Ω.

The lattice Boltzmann method aims at solving the discretized Boltzmann equation, which describes the evolution of a velocity distribution function $f(x, c, t)$. The following formulation of the equation accounting for the presence of body force is considered:

$$\frac{\partial f}{\partial t} + c_\alpha\frac{\partial f}{\partial x_\alpha} + \frac{F_\alpha}{\rho}\frac{\partial f}{\partial c_\alpha} = \Omega(f) \tag{8}$$

where ξ denotes the particle velocity and $\Omega(f)$ is the collision operator and $\mathbf{F}$ a generic forcing term. In the framework of an LBM solver, (8) is discretized in the physical space, in time, and in the velocity space by defining a set of c_i particle velocities, yielding the following scheme in the general formulation:

$$\begin{cases} f_i^{\mathrm{coll}}(x,t) = f_i(x,t) + \Omega(f_i(x,t)) + F_i(x,t) \\ f_i(x, t+\Delta t) = f_i^{\mathrm{coll}}(x - c_i\Delta t, t) \end{cases} \tag{9}$$

where $F_i(x,t)$ is the discretized forcing term, related to the IBM Force through the Guo's forcing scheme described in [49]. As mentioned before, the LBM scheme is applied in the present fluid solver to the mass and momentum equations. The macroscopic variables are recovered by computing the moments of f_i as:

$$\rho(x,t) = \sum_i f_i(x,t) \qquad \rho u_\alpha(x,t) = \sum_i c_{i,\alpha} f_i(x,t) + \frac{\Delta t}{2}F_\alpha \tag{10}$$

The collision kernel used in the present work relies on the unified model of Farag et al. [50], using a Hybrid Recursive Regularization (HRR) method of Jacob et al. [51]. This solver has been shown to successfully to simulate compressible flows around transonic to supersonic regimes [52], which are the subject of interest here. It consists of a hybrid approach in which the density and momentum equations are solved through a LBM scheme while a total energy equation is solved using a specifically designed finite volume approach, proposed by G. Wissocq et al. [53], to optimize the efficiency of the method in terms of computational cost and memory storage.

For all cases presented, parameters of the model were set as follows: $\kappa = 0, \xi = 0, \sigma = 0.98$ (see [50] for details), leading to the following formulation of the collision step in (9):

$$f_i^{\mathrm{coll}}(x,t) = f_i^{\mathrm{eq}}(x,t) + \left(1 - \frac{\Delta t}{\tau}\right)f_i^{\mathrm{neq}}(x,t) + \frac{\Delta t}{2}F_i(x,t) \tag{11}$$

τ is related to the fluid dynamic viscosity μ as $\mu + \rho\nu_{sc} = \left(\tau - \frac{\Delta t}{2}\right)\rho c_s^2$ and $c_s = \frac{1}{\sqrt{3}}\frac{\Delta x}{\Delta t}$. ν_{sc} is an artificial viscosity activated by a shock sensor only near discontinuities, its definition is given in Appendix A.

f_i^{eq}, the equilibrium distribution function and f_i^{neq}, the off-equilibrium distribution function, are expanded up to third order on the basis of Hermite polynomials, detailed expressions being provided in Appendix A. Definition of the equilibrium coefficients $a_{(\mathrm{eq})}^{(n)}$ related to the corresponding Hermite tensor $H_i^{(n)}$ are based on the macroscopic variable. According to the unified model [50], the zeroth order equilibrium coefficient is computed as:

$$a_{(\mathrm{eq})}^{(0)} = \rho + \frac{\omega_i - \delta_{0i}}{\omega_i}\rho(\theta - 1) \tag{12}$$

where $\theta = T/T_{\mathrm{ref}}$ is the dimensionless temperature coming from the resolution of an energy equation and the EOS. Similarly, definitions of the off-equilibrium coefficients $a_{(\mathrm{neq})}^{(2)}$ and $a_{(\mathrm{neq})}^{(3)}$ are computed following the Hybrid Recursive Regularized model.

2.3 | Diffuse Forcing Immersed Boundary Method

The general idea of the diffuse forcing IBM is to introduce a source term in the governing equations of the fluid in order to account for the influence of the solid over the fluid. In the general governing equations for the fluid introduced earlier (1), it is symbolized by the $\mathbf{f}$ term.

Each component of the vector $\mathbf{S}^{\mathrm{IB}}$ is expected to model the influence of the solid over the fluid in its respective equation. As a consequence, let us define the different components of the vector $\mathbf{S}^{\mathrm{IB}}$ here as:

$$\mathbf{S}^{\mathrm{IB}} = [0, f_x^{\mathrm{IB}}, f_y^{\mathrm{IB}}, f_z^{\mathrm{IB}}, \mathrm{W}^{\mathrm{IB}} + \mathrm{Q}^{\mathrm{IB}}]^{\mathrm{T}} \tag{13}$$

where:

- the first component, appearing in the continuity equation, is set to be zero since the solid structure is not expected to create/destroy any mass in the system.

- f_β^{IB} is included in the β component of the momentum equation and corresponds to the volume force of the solid acting on the fluid, with $\beta \in \{x, y, z\}$.
- the last component is identified as an energy source, and may be divided into two parts:
  - W^{IB} that accounts for the work of the structure over the fluid, which is zero in the case of a non-moving structure,
  - Q^{IB} that considers the heat exchange between the fluid and the solid in the case of a fluid-structure thermal coupling.

In the proposed method, the primary role of these forcing terms is to impose Dirichlet-type boundary conditions (BC) on the fluid near the solid boundary. Examples of such BC include no-slip and isothermal conditions, where the solid velocity and temperature are imposed on the fluid, which can be expressed respectively as:

$$u_\alpha = u_\alpha^B \tag{14}$$

$$T = T_B \tag{15}$$

with u_α^B the solid velocity and T_B the solid temperature.

2.3.1 | Derivation of an Expression for the Momentum Source Term for Dirichlet Boundary Condition

First, let us focus on the research of an explicit definition for f_β^{IB}, the β-component of the volume force that will enforce the desired BC to the fluid with $\beta \in \{x, y, z\}$ To accomplish this, the β component of the momentum equation in Equation (1) is discretized in time using finite difference discretization. This results in:

$$\frac{\rho u_\beta^{t+\Delta t} - \rho u_\beta^t}{\Delta t} + \frac{\partial(\rho^t u_\alpha^t u_\beta^t + p^t \delta_{\alpha\beta}^t)}{\partial x_\alpha} - \frac{\partial(\tau_{\alpha\beta}^t)}{\partial x_\alpha} = +f_\beta^{\mathrm{IB},t} \tag{16}$$

where the following convention $\phi^t = \phi(\mathbf{x}, t)$ has been used and an implicit summation is done over the index $\alpha \in \{x, y, z\}$.

A prediction step is considered, that consists in solving Equation (16) without considering the solid boundary as:

$$\frac{\rho u_\beta^* - \rho u_\beta^t}{\Delta t} + \frac{\partial(\rho^t u_\alpha^t u_\beta^t + p^t \delta_{\alpha\beta}^t)}{\partial x_\alpha} - \frac{\partial(\tau_{\alpha\beta}^t)}{\partial x_\alpha} = 0 \tag{17}$$

where u_β^* denotes the predicted velocity field.

By subtracting Equation (17) to Equation (16), an expression for the forcing term is derived: $f_\beta^{\mathrm{IB},t} = \frac{\rho u_\beta^{t+\Delta t} - \rho u_\beta^*}{\Delta t}$. Moreover, $u_\beta^{t+\Delta t}$ can be replaced by u_β^B in order for the source term to enforce the desired Dirichlet BC (14).

Finally, an explicit expression of the forcing is obtained as:

$$f_\beta^{\mathrm{IB},t} = \rho \frac{u_\beta^B - u_\beta^*}{\Delta t} \tag{18}$$

It can be noticed here that incorporating ρ as a factor in the expression above is only valid since the source term in the continuity equation is zero, resulting in $\rho^* = \rho^{t+\Delta t}$.

2.3.2 | Derivation of an Expression for the Energy Source Term for Dirichlet Boundary Condition in the Energy Equation

A similar approach is employed to derive explicit expressions for the source terms Q^{IB} and Q^{IB} in the energy equation.

$$\frac{(\rho \mathrm{E})^{t+\Delta t} - (\rho \mathrm{E})^t}{\Delta t} + \frac{\partial((\rho \mathrm{E}^t + p^t)u_\alpha^t)}{\partial x_\alpha} - \frac{\partial(\tau_{\alpha\beta}^t u_\alpha^t + q_\alpha^t)}{\partial x_\alpha} = \mathrm{W}^{\mathrm{IB},t} + \mathrm{Q}^{\mathrm{IB},t} \tag{19}$$

$$\frac{(\rho \mathrm{E})^* - (\rho \mathrm{E})^t}{\Delta t} + \frac{\partial((\rho \mathrm{E}^t + p^t)u_\alpha^t)}{\partial x_\alpha} - \frac{\partial(\tau_{\alpha\beta}^t u_\alpha^t + q_\alpha^t)}{\partial x_\alpha} = 0 \tag{20}$$

Subtracting the prediction Equation (20) to Equation (19) results in:

$$\mathrm{W}^{\mathrm{IB},t} + \mathrm{Q}^{\mathrm{IB},t} = \frac{(\rho \mathrm{E})^* - (\rho \mathrm{E})^t}{\Delta t} \tag{21}$$

The total energy ρE is then decomposed into kinetic and internal energies according to Equation (7):

$$\mathrm{W}^{\mathrm{IB},t} + \mathrm{Q}^{\mathrm{IB},t} = \frac{\left(\frac{1}{2}\rho u_\alpha^2\right)^{t+\Delta t} - \left(\frac{1}{2}\rho u_\alpha^2\right)^*}{\Delta t} + \frac{(\rho C_v T)^{t+\Delta t} - (\rho C_v T)^*}{\Delta t} \tag{22}$$

Similarly as before, $u_\alpha^{t+\Delta t}$ and $T^{t+\Delta t}$ are replaced according to the desired boundary conditions (14) while ρ and C_v are set as factors, resulting in:

$$\mathrm{W}^{\mathrm{IB},t} + \mathrm{Q}^{\mathrm{IB},t} = \frac{1}{2}\rho \frac{(u_\alpha^B)^2 - (u_\alpha^*)^2}{\Delta t} + \rho C_v \frac{T_B - T^*}{\Delta t} \tag{23}$$

The above expression is finally separated into two parts, resulting in the following explicit expressions for the source terms:

$$\mathrm{W}^{\mathrm{IB},t} = \frac{1}{2}\rho \frac{(u_\alpha^B)^2 - (u_\alpha^*)^2}{\Delta t} \tag{24}$$

$$\mathrm{Q}^{\mathrm{IB},t} = \rho C_v \frac{T_B - T^*}{\Delta t} \tag{25}$$

It can be noticed here that $\mathrm{W}^{\mathrm{IB},t} = \left(\frac{u_\alpha^B + u_\alpha^*}{2}\right) f_\alpha^{\mathrm{IB},t}$, which corresponds to the work of the force $f_\alpha^{\mathrm{IB},t}$ applied at the mean velocity between the predicted u_α^* and solid u_α^B velocities.

The diffuse forcing method presented above is then used to enforce any kind of Dirichlet boundary condition along the solid boundary. In the present solver, it is implemented through the following algorithm at every time step:

1. Prediction of u_α^*, T^* computed from (1) by neglecting solid source terms
2. Computation of the IBM source term from Equations (18), (24), and (25)
3. Solving the general equations (1)
4. Advance to the next time step $t = t + \Delta t$

A schematic view of the algorithm is shown in Figure 3, including sharp forcing, presented further in this article.

2.3.3 | Spatial Discretization of the IBM

In the present IBM framework, two distinct spatial domains are considered: Ω is the fluid domain and Γ is the solid boundary as shown in Figure 1, with the following specificities:

- Ω is discretized by a fixed cartesian mesh,
- Γ is discretized by a set of Lagrangian nodes which are free to move anywhere within the limits of Ω.

The nodes in Ω that lie inside the solid boundary are not treated differently from the ones outside, thus ensuring a strict independence of the two meshes and as expected, no update of the fluid mesh is required during the computation. Also, Ω is continuous through Γ, which will be exploited later to propose the simplified adiabatic model.

Since the boundary conditions are defined only on the solid domain Γ, which is not expected to coincide exactly with Ω, transfer operators must be defined and implemented in the numerical discretization. In the framework of diffuse forcing, the classical method consists of defining two reciprocal operators, named Interpolation and Spreading, such that:

$$\Phi(X,t) = \sum_{\Omega_f} \phi(x,t)\delta_{\mathrm{IB}}(x-X)\Delta x^3 \tag{26}$$

$$\phi(x,t) = \sum_{\Gamma_s} \Phi(X,t)\delta_{\mathrm{IB}}(x-X)\Delta q \Delta r \Delta s \tag{27}$$

Φ, X denotes variables from the solid boundary Γ while ϕ, x denotes variables from the fluid domain Ω. $\Delta q \Delta r \Delta s$ is the quadrature volume for the solid mesh.

The δ_{IB} function is a discretized approximation of the Dirac delta function. Here, the following expression is used:

$$\delta_{\mathrm{IB}}(x,y,z) = \frac{1}{\Delta x^3}\tilde{\delta_{\mathrm{IB}}}\left(\frac{x}{\Delta x}\right)\tilde{\delta_{\mathrm{IB}}}\left(\frac{y}{\Delta x}\right)\tilde{\delta_{\mathrm{IB}}}\left(\frac{z}{\Delta x}\right) \tag{28}$$

$$\tilde{\delta_{\mathrm{IB}}}(r) = \begin{cases} \frac{1}{2r}\left(1+\cos\left(\frac{\pi r}{1.5}\right)\right) & \text{if } r < 1.5 \\ 0 \text{ otherwise.} \end{cases} \tag{29}$$

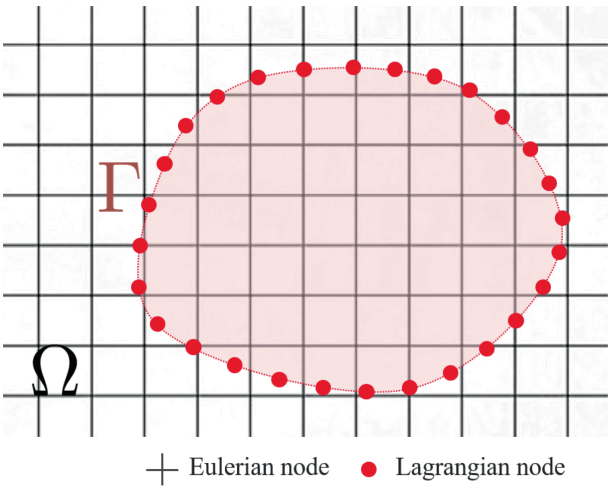

FIGURE 1 | Immersed boundary (red) on a cartesian grid.

These two operators must be reciprocal to ensure the conservation of forces on fluid and solid domains, and thus a correct coupling between fluid and solid dynamics. If this condition on the reciprocity of operators is not satisfied, the total forces computed in the solid domain will not equal the total force in the fluid domain, violating Newton third law of motion.

However, an issue arises in the definition of (27) since the quadrature volume is not entirely defined in the discretized space.

Indeed, Γ being a boundary domain, it has one dimension less than the dimension of Ω. As a consequence, at least one of the components in $\Delta q \Delta r \Delta s$ is unknown.

This unknown component, referred to as the Lagrangian weight $W(X)$, can be computed by a calibration procedure, as detailed by Gsell, D'Ortona, and Favier [23] and by Cheylan et al. [54]. The analytical formulation they proposed for the calculation of $W(X)$ is used here. This calibrating ensures a really high level of reciprocity between the two operators. In the case of a non-moving solid, this calibration is computed and performed only once at the first time step.

A simple test is realized to validate the correct calibration of $W(X)$: it involves the Spreading of a unity scalar field from Γ to Ω using (27), and the interpolation of this field back to Γ using (26). The gap between the initial and result field represents the so-called SPRINT (Spreading/Interpolation) error, denoted ϵ_{SP} of the method, on each Lagrangian node.

$$\epsilon_{SP}(X_k) = \sqrt{(\Phi_k - 1)^2} \tag{30}$$

$$\Phi_k = \sum_{\Omega_f} \phi_i \delta_{\mathrm{IB}}(x_i - X_k)\Delta x^3 \tag{31}$$

$$\phi_i = \sum_{\Gamma_s} 1 \times \delta_{\mathrm{IB}}(x_i - X_k)\Delta q \Delta r W(X_k) \tag{32}$$

In all simulations from the results section 3, the Lagrangian weight was calibrated at the first iteration. As a consequence, $\epsilon_{SP}(X_k) < 10^{-4}\ \forall X_k$.

2.4 | Hybridization With the Ghost-Nodes Forcing Method

The diffuse forcing IBM can not easily be expressed for boundary conditions that differ from Dirichlet-type (such as Neumann boundary conditions). It was attempted in [31, 32, 34] but this leaves too many arbitrary parameters (such as the distance of projection of the Lagrangian node) and it is not clear if the zero-gradient on temperature boundary condition is well applied. Therefore, as proposed by Boukharfane et al. in [39] a GN forcing method was implemented in addition to the current diffuse forcing IBM in order to ensure a strict imposition of boundary conditions of any kind.

The ghost-nodes forcing method is different from the diffuse forcing introduced earlier in the way the boundary condition is enforced. As a matter of fact, the GN forcing is usually classified as the "sharp interface" method in opposition to the "diffuse

interface" methods as proposed by Mittal and Iaccarino [20]. The general idea of GN forcing is to classify some Eulerian fluid nodes that stand near the boundary inside Γ and to enforce directly the desired boundary conditions on these nodes. The boundary condition may be of any type, though, in the present framework, it is used only to enforce Neumann boundary conditions corresponding to heat flux conditions.

In particular, imposing a zero-heat-flux boundary condition is a common choice when the thermal coupling between the solid structure and the fluid is neglected and the solid wall is considered adiabatic:

$$\partial_{\mathbf{n}} T = Q_B \tag{33}$$

with $\partial_{\mathbf{n}}$ the spatial derivative to the wall normal direction and Q_B the heat flux between the fluid and the solid. Imposing $Q_B = 0$ results in the adiabatic boundary condition, expressed as:

$$\partial_{\mathbf{n}} T = 0 \tag{34}$$

2.4.1 | Node Classification

The first step in the algorithm is to classify each Eulerian node into one of these categories (Figure 2):

- x_F, Fluid node: an Eulerian node that stands outside the solid boundary.
- x_S, Solid node: an Eulerian node that stands inside the solid boundary
- x_{GN}, Ghost node: an Eulerian node that is a solid node near the solid boundary. The enforcement of the desired BC through the GN method happens there.

Initially, the classification of the Eulerian nodes as x_F or x_S is unknown since the solid mesh is independent of the Eulerian fluid mesh. A method is required to determine for each node if it is outside or inside the solid boundary.

This problem simply comes down to a point-in-polygon problem from computational geometry. One of the simplest and more generic method to solve this is the ray casting algorithm. It consists in computing how many times a line (ray), starting from the candidate node into the x, y, z directions, would cross the boundary. This number is denoted C_n. If it is odd, then the node lies inside the boundary, otherwise, it is outside.

The identification of the ghost nodes x_{GN} is realized in a simple manner: among all Eulerian x_S, if a node has at least one neighbor which is of the type x_F, then it is flagged as ghost node, otherwise, it remains of the x_S type.

2.4.2 | Image Point

The next step of the GN method consists in projecting each ghost node in the direction normal to the solid boundary. The resulted point is denoted as the "image point" X_{IP}, which will usually not coincide with an Eulerian node. The boundary condition on x_{GN} will be enforced according to the value of the field at its corresponding image point. Thus, an interpolation should be realized around X_{IP}. In the present work, it is done with the previously introduced interpolation operator (26), but the nodes classified as x_{GN} or x_S are excluded from the interpolation stencil. In order to recover a correct interpolation, the results are normalized by the sum of the $\delta_{\text{IB}}(|x_i - X_{IP}|)$ as:

$$\Phi(X_{IP}, t) = \frac{\sum_{\Omega_{fluid}} \phi(x,t)\delta_{\text{IB}}(x_f - X_{IP})\Delta x^3}{\sum_{\Omega_{fluid}} \delta_{\text{IB}}(x_f - X_{IP})\Delta x^3} \tag{35}$$

2.4.3 | Enforcement of the Neumann Boundary Condition

Concerning the imposition of the boundary condition on the GN, let us focus only on the enforcement of Neumann BC on T here. The goal is to ensure the desired boundary condition (33) on the solid boundary by modifying the value of temperature. A 2nd order approximation of (33) results in:

$$\partial_{\mathbf{n}}(T) = \frac{T(X_{IP}) - T(x_{GN})}{2h} \quad \text{where } h \text{ is the distance between } x_{GN} \text{ and the solid boundary} \tag{36}$$

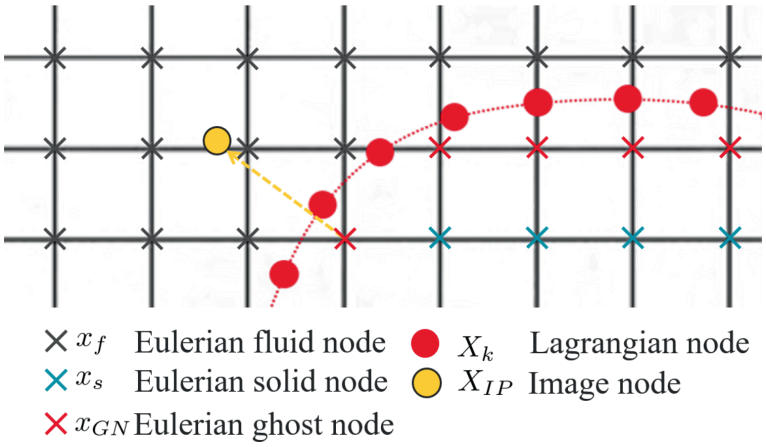

FIGURE 2 | Eulerian nodes status and projection of a ghost-node.

From (33), the value of temperature to enforce on the x_{GN} is deduced as:

$$T(x_{GN}) = 2hQ_B - T(X_{IP}) \tag{37}$$

In the specific case of an adiabatic boundary, $Q_B = 0$, which results in the copy of the temperature at the image point to the ghost node as: $T(x_{GN}) = T(X_{IP})$.

2.4.4 | Implementation of the Hybrid GN-IBM

The proposed numerical framework for a hybrid sharp/diffuse IBM is illustrated in Figure 3.

2.5 | Introduction of the T-Passive Bboundary Condition

The hybrid GN-IBM method presented in the previous section is useful to enforce strictly the Neumann boundary condition while preserving the advantages of the diffuse forcing for the Dirichlet boundary condition. Though, the additional computational effort required by the GN part of the algorithm, see Figure 3, is not negligible. This is particularly true in the case of a moving boundary, when the classification of nodes as solid or fluid has to be repeated each time step. The interpolation of the temperature field on the image point also represents an additional cost.

In the present section, the particular case of the adiabatic boundary condition is considered. That type of BC is of interest when the heat exchange between the fluid and the solid is negligible. As mentioned earlier, the adiabatic boundary condition is a special case of Neumann BC where the desired Temperature gradient should be zero.

When the solid structure is considered as adiabatic, another IBM model is proposed where no special treatment is applied to the temperature field, saving CPU time compared to the hybrid IBM-GN. The validity of this so-called "T-Passive" IBM model to

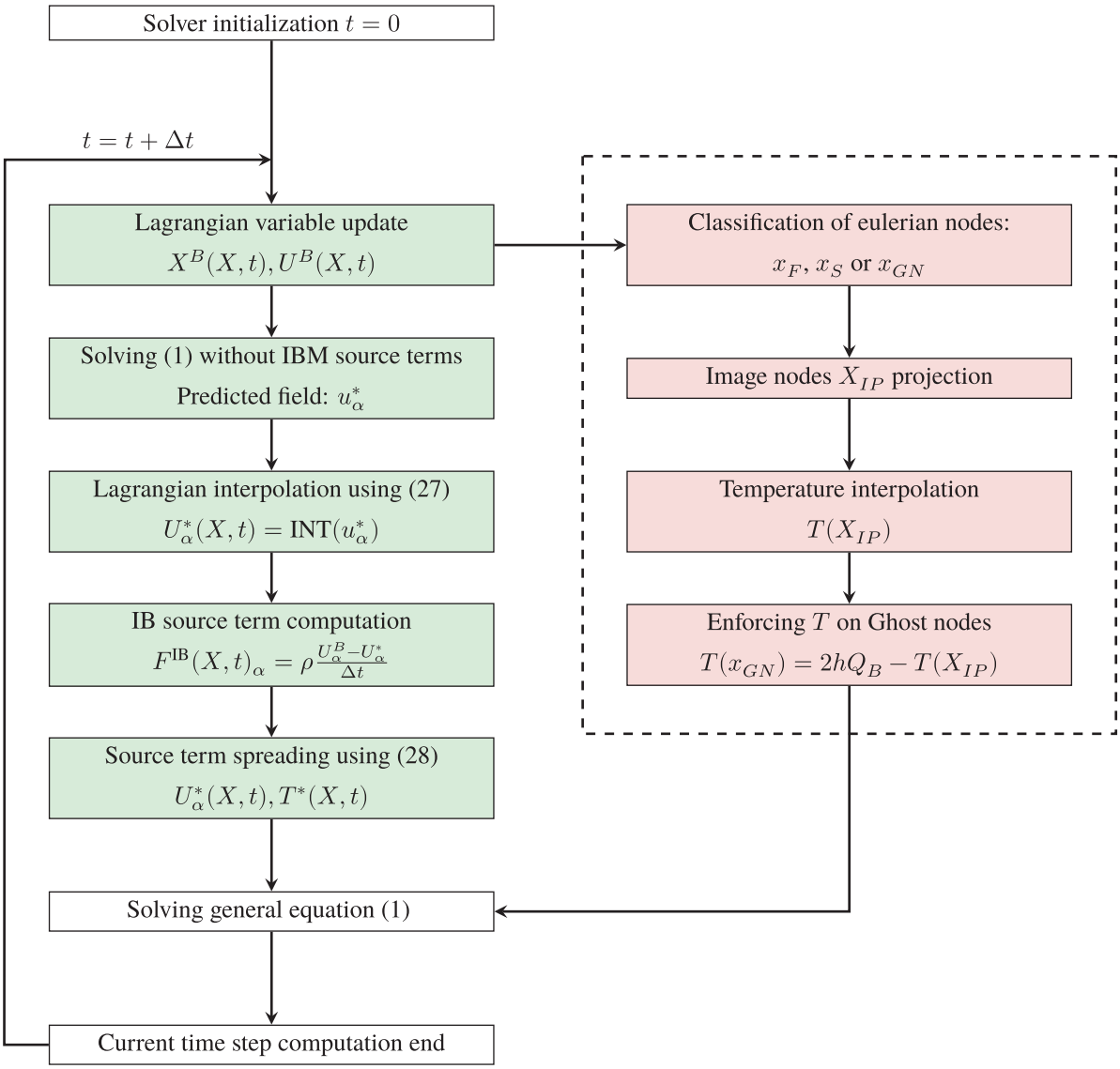

FIGURE 3 | Schematic view of the hybrid IBM-GN algorithm enforcing adiabatic FS interface. Diffuse and sharp forcing approaches are distinguished respectively as green and red.

simulate the interaction of the flow with adiabatic boundaries is investigated in the results section 3.

The relevance of this approach is supported by the different time scales involved in the energy conservation for the fluid close to the boundary. In most of the situations of interest, the change of temperature is dominated by convection and conduction and is only significant for much larger time scales. Taking into account that the forcing of the velocity field is sufficient to ensure the correct boundary condition for the energy convection, simply neglecting any conductive heat exchange through the selected boundary is likely to be enough to obtain a satisfactory adiabatic boundary condition.

In the framework of a T-passive boundary, the no-slip condition is still enforced through diffuse forcing and no additional treatment is applied to the temperature field, which is simply left unconstrained. This means that, contrary to the Hybrid IBM-GN, there is no need to compute the position of ghost nodes or to interpolate a field around any image point. This simplifies greatly the method and is expected to highly reduce the computational cost of thermal boundary condition management.

2.5.1 | Analysis of the Heat Conduction Term in the Dimensionless Energy Equation

In the following, a theoretical analysis is performed to assess the validity of the simplified T-passive model.

Firstly, the energy equation can be expressed under a dimensionless form in order to identify a term related to heat conduction. The validity of the simplified model depends on whether this term is negligible or not.

$$\partial_{\tilde{t}}(\tilde{\rho}\tilde{E}) + \partial_{\tilde{\alpha}}(\tilde{\rho}\tilde{E}\tilde{u_\alpha}) = -\partial_{\tilde{\alpha}}(\tilde{p}\tilde{u_\alpha}) + \frac{M_a^2}{R_e}\partial_{\tilde{\alpha}}(\tilde{\tau_{\alpha\beta}}.\tilde{u_\alpha}) - \frac{1}{R_e P_r}\partial_{\tilde{\alpha}}(\tilde{q_\alpha}) = 0 \tag{38}$$

where $\tilde{.}$ denotes dimensionless quantities, M_a is the Mach number, R_e is the Reynolds number, P_r is the Prandtl number and $\tilde{q_\alpha}$ is the dimensionless heat flux.

Here, the heat conduction term is identified as $\frac{1}{R_e P_r}\partial_{\tilde{\alpha}}(\tilde{q_\alpha})$. This means that the simplification is valid as long as the dimensionless variable $R_e P_r$ is large enough, and as long as the temperature delta throughout the boundary is small $\Rightarrow \tilde{q_\alpha} \approx 0$.

2.5.2 | Analysis of the Characteristic Time

Secondly, an analysis of the characteristic times related to convection and conduction is realized as:

$$\tau_{\text{convection}} = \frac{L}{u_\infty} \tag{39}$$

$$\tau_{\text{conduction}} = \frac{L^2}{\alpha} = \frac{\rho C_p L^2}{\lambda} \tag{40}$$

with τ a characteristic time, L a characteristic length, u_∞ a characteristic velocity, and α the thermal diffusivity.

In the case of a transient simulation, the simplified model is thus valid if the $\tau_{\text{conduction}}$ is large to $\tau_{\text{convection}}$ and the total simulation time. The ratio of these two characteristic times also reveals a dependence on the dimensionless number $R_e P_r$, defined from characteristic variables taken in the bulk flow, which is consistent with the previous analysis. Further investigations would be needed in cases where the T-Passive model fails due to particular phenomena encountered inside the boundary layer where these numbers can take different values. Such situations have not been encountered in the tests proposed in the current article.

$$\frac{\tau_{\text{conduction}}}{\tau_{\text{convection}}} = \frac{\rho u_\infty L C_p}{\lambda} = \frac{\rho u_\infty L}{\mu}\frac{\mu C_p}{\lambda} = RePr \tag{41}$$

3 | Numerical Results and Validation of the Proposed IB Model for Compressible Flow

In order to validate the proposed method, a set of 1D and 2D test cases is carried out, and the results are compared with various references.

Case 1 Subsonic unsteady flow around a cylinder at Re 100: standard validation for incompressible flow

Case 2 Thermal Couette flow at Ma 1.3: demonstration of the validity of the two IBM models in compressible supersonic conditions

Case 3 Supersonic laminar flow around a cylinder at Re 300 and Ma 2.0: comparison of the present results with other works from the IBM literature

Case 4 Supersonic turbulent flow on a wedge at Re 50000 and Ma 2.0: validation of the method for sharp geometries and turbulent flows

Case 5 Shock wave impact on a cylinder: Assessment of the ability of the IBM model to deal with rapid transient flows

3.1 | Case 1: Subsonic Unsteady Flow Around a Cylinder

Let us first consider an unsteady flow past a circular cylinder, neglecting the compressible effect by setting the Mach number to 0.1. A cylinder, of diameter D is centered at coordinates (0,0) in a rectangular domain of size $[-25\text{D}, 75\text{D}] \times [-30\text{D}, 30\text{D}]$, discretized by a cartesian mesh with $\text{D} = 80\Delta x$. The flow at the inlet is towards the x-direction and its velocity is set according to the Mach number $M_\infty = 0.1$ as: $U_\infty = M_\infty c_\infty$ with c_∞ the speed of sound.

The free-stream pressure and temperature respectively equal to $p_\infty = 1\text{bar}$ and $T_\infty = 292$ K. The Reynolds number is set to $Re_\infty = 100$, leading to an unsteady wake. The vortex generation frequency is analyzed by computing the Strouhal number:

$$S_t = \frac{f_{\text{req}} D}{U_\infty}$$

where f_{req} is the vortex shedding frequency.

The boundary condition imposed on the cylinder is: no slip isothermal, with $T_B = T_\infty$.

One advantage of the IBM framework employed here is that it makes it easy to compute the aerodynamic forces endured by the solid cylinder as:

$$C_d = \frac{\sum_{\Gamma_s} F_x^{\mathrm{IB}}}{\rho_\infty u_\infty S_{\mathrm{eff}}}$$

$$C_l = \frac{\sum_{\Gamma_s} F_y^{\mathrm{IB}}}{\rho_\infty u_\infty S_{\mathrm{eff}}}$$

where C_d is the drag coefficient and C_l is the lift coefficient.

F_x^{IB} and F_y^{IB} are the immersed boundary forces respectively in x and y directions, summed over the lagrangian nodes. S_{eff} is the effective aerodynamic surface, equal to $D\Delta \times \Delta x$ in the present case.

The lift coefficient is oscillating around a 0 mean value at the same frequency as the vortex shedding.

This case represents a classical experiment in the literature as a number of authors have studied the flow around a cylinder at Reynolds 100 in the past. Though, despite its simplicity, previous studies show variability in results, with no clear consensus on the exact value among numerical or experimental studies, as shown in [55–57]. Present results fall within the dispersion of results and agree with most recent IBM simulation reference results as illustrated in Table 1.

An instantaneous velocity field is shown in Figure 4.

3.2 | Case 2: Thermal Couette Flow With an Adiabatic Wall

After validating the present IBM model on a nearly incompressible flow, let us check now its ability to deal with compressible flow and to enforce thermal BC through the analysis of a thermal Couette flow at $M_\infty = 1.3$. The present case consists of the simulation of a 1D shear flow between two walls and is configured as follows:

- The fluid domain is a 1D line placed in the Y-Direction between the bottom wall at Y=0 and the top wall at Y=H
- The top wall is considered stationary and adiabatic, so that the boundary conditions for the fluid are $U = 0$ and $\frac{\partial T}{\partial y} = 0$
- The top wall is moving and isothermal, leading to the following BC: $U = U_B$ and $T = T_B$.

The fluid domain is discretized with H=100Δx. The final state is reached once the thermal dissipation balances out the viscous heat generated by the shear stress. This test case assesses the capability of the present IBM model to accurately reproduce the shear stress on walls, and compares the two thermal models in providing the correct boundary condition for energy on the adiabatic wall.

In practice, in order to properly assess the accuracy of the proposed IBM, the bottom and top walls need to be totally incorporated into the fluid domain. To satisfy this condition, a "double" Couette test is set up, with all the domain limits being periodic. Otherwise, the results would be influenced by the chosen boundary condition on the top and bottom limits of the

TABLE 1 | Cylinder at $M_\infty = 0.1$ and $Re_\infty = 100$: Comparison of the results.

	C_d	C_l	S_t
Liu, Zheng, and Sung [58]	1.35	±0.339	0.164
Zhou, So, Lam [59]	1.49	±0.248	0.162
Kim, Kim, and Choi [60]	1.33	±0.320	0.165
Bourguet and Lo Jacono [61]	1.32	±0.320	0.164
Gsell and Favier [62], IBM	1.37	±0.340	0.164
Cheylan et al. 2022 [54], IBM	1.41	±0.330	0.166
Ménez et al. 2023 [37], IBM	1.36	±0.297	0.161
Norberg [56] empirical	—	±0.227	0.164
present	1.31	±0.316	0.162

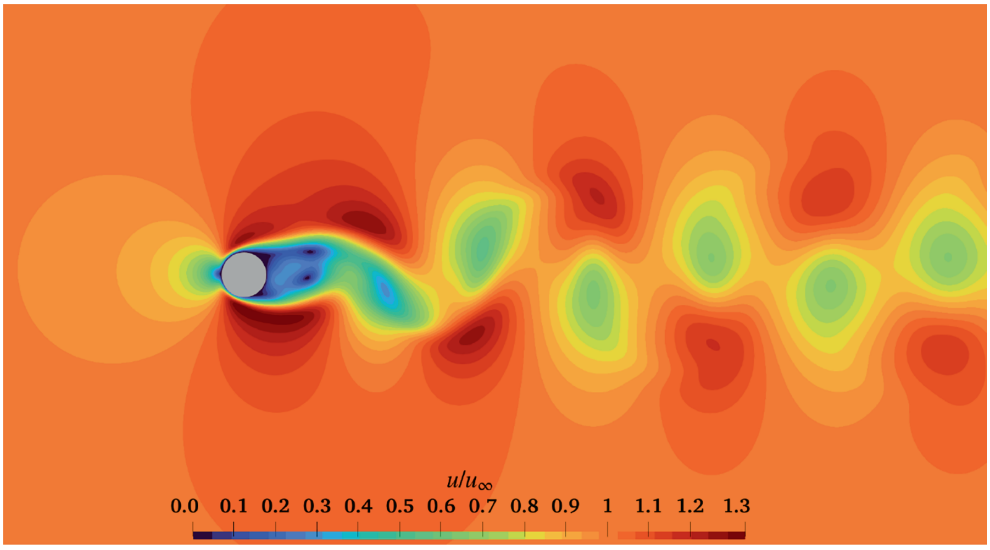

FIGURE 4 | Velocity field around a cylinder at $M_\infty = 0.1$ and $Re_\infty = 100$.

domain. In practice, only the $x = [0, H]$ section is of interest as shown in the figures.

The fluid is initialized by: Temperature $T_0 = 300$, pressure $P_0 = 8 \times 10^5$ Pa, heat capacity ratio $\gamma = 1.4$, viscosity $\mu_0 = 10^{-4}$, specific gas constant $r = 287$ Prandtl number $Pr = 0.71$. The Reynolds number computed from the channel height $H = 10^{-4}$, initial density $\rho_0 = \frac{P_0}{rT_0} = 9.29$ and top wall velocity $U_B = M_a * c_0 = 451.3$ is set to $Re = 4193$. The wall temperature of the top wall is $T_B = 300$K. The dynamic viscosity is following a power law as:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^n \tag{42}$$

Analytical solutions for velocity and temperature profiles can be found in [63]

$$\frac{T}{T_B} = 1 + \zeta\left[1 - \left(\frac{u_x}{U_B}\right)^2\right] \tag{43}$$

$$\frac{u_x}{U_B} = \left(1 + \frac{2}{3}\zeta\right)\frac{y}{H} - \zeta\left[\frac{u_x}{U_B} - \frac{1}{3}\left(\frac{u_x}{U_B}\right)^3\right] \tag{44}$$

Where $\zeta = Pr(\gamma - 1)/2\, M_\infty^2$.

The simulation was performed with the hybrid GN-IBM model and the simplified T-Passive model. The results are compared to reference solutions in Figure 5. It is found that both models yield satisfying results on velocity and temperature fields. One can notice that, because of the diffused nature of the IBM (which has a radius stencil of $1.5\Delta x$), the wall velocity is enforced in an area wider than expected. As a consequence, the shear stress is overestimated, leading to a more viscous heat. This explains the higher temperature reached by the fluid on the adiabatic wall, which can obviously be reduced by refining the mesh.

Finally, the two thermal IBM models are compared. First, as expected, the two velocity fields are exactly the same. One can then notice that the temperature field resulting from the simplified model is further from the reference solution than the one obtained with the strictly adiabatic model. Though, by computing the $\mathcal{L}_2$ error to the reference solution, it can be concluded that this difference remains negligible, showing the good ability of the T-Passive IBM to model an adiabatic wall in the present situation.

The $\mathcal{L}_2$ errors on temperature and velocity to the reference solution are given in Table 2 as:

$$\epsilon^{\phi}_{\mathcal{L}_2} = \sqrt{\frac{\sum_n (\phi^*(x_i) - \phi(x_i))^2}{\sum_n (\phi^*(x_i))^2}}$$

A grid convergence study is performed by varying the number of nodes between the two walls. Figure 6 shows a second-order convergence in temperature, while a first-order is observed for velocity.

3.3 | Case 3: Supersonic Laminar Flow Around a Cylinder

Now, let us analyze the configuration of a laminar supersonic flow around a circular cylinder. The Mach and Reynolds number are respectively set to $M_\infty = 2$ and $R_{e_\infty} = 300$. Pressure and temperature at the inlet are imposed as $p_\infty = 1$ bar and $T_\infty = 300$.

A circular cylinder of diameter D, centered on coordinates $(0; 0)$, is modeled by a set of Lagrangian nodes, in a rectangular domain of dimensions $[-2.5\text{D}, 17.5\text{D}] \times [-10\text{D}, 10\text{D}]$. A uniform mesh is used with $\Delta x = D/80$. The distance between two Lagrangian nodes is set up to be equal to Δx.

The imposed boundary condition on the IB is no slip and adiabatic $U_B = 0$, $\partial_{\mathbf{n}} T = 0$.

The density field obtained with the Hybrid IBM-GN method is shown in Figure 7. A Schlieren representation of the density

TABLE 2 | L2 error on the thermal Couette flow.

	$\epsilon^U_{\mathcal{L}_2}$	$\epsilon^T_{\mathcal{L}_2}$
T-Passive IBM	2.1×10^{-2}	3.4×10^{-3}
Hybrid GN-IBM	2.1×10^{-2}	3.3×10^{-3}

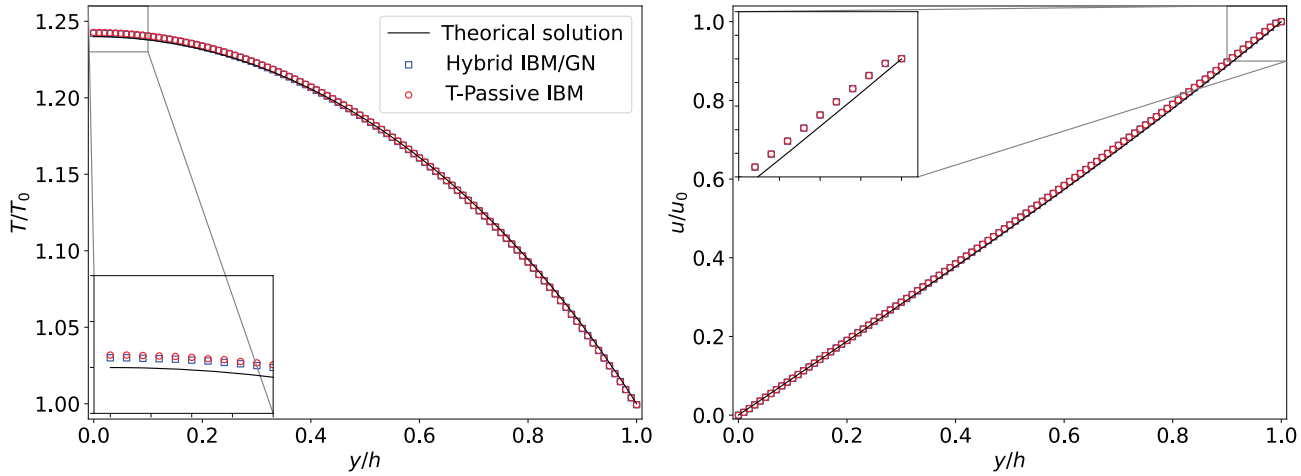

FIGURE 5 | Thermal Couette flow at Ma 1.3 with an adiabatic wall.

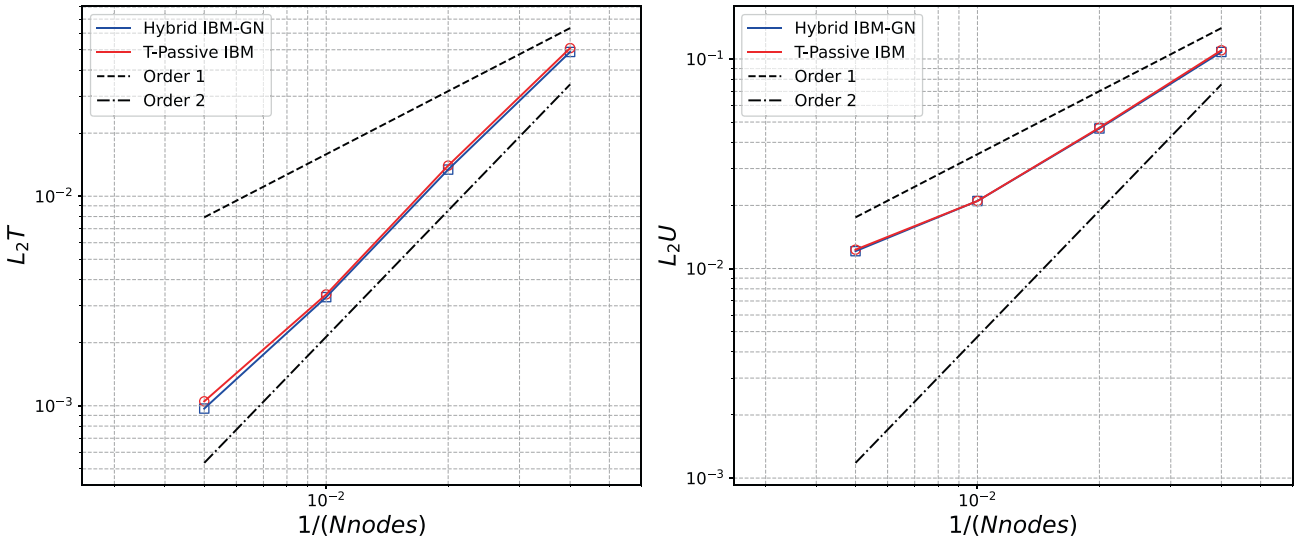

FIGURE 6 | Thermal Couette flow at Ma 1.3 convergence study.

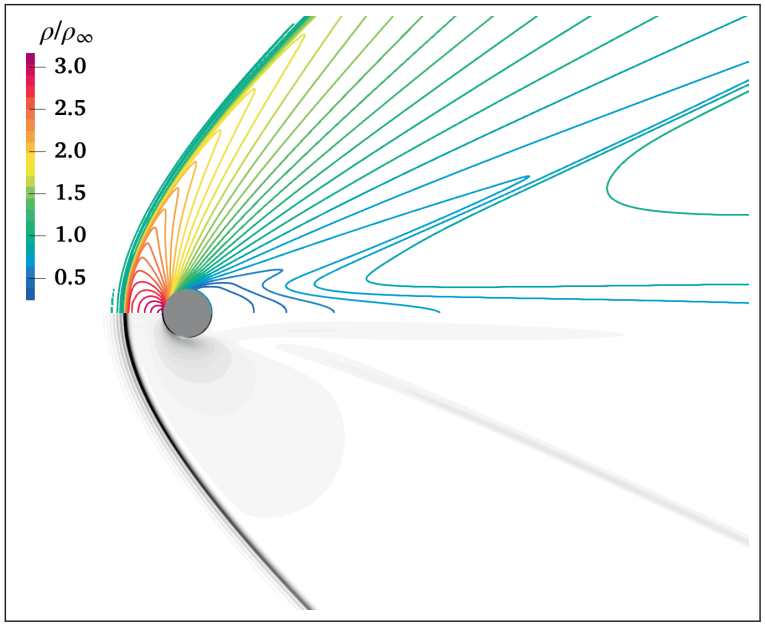

FIGURE 7 | Flow around a cylinder at Ma 2.0 Re 300, top: density field, bottom: Schlieren representation of the density gradient.

gradient is also shown in Figure 7 to illustrate the shock position relative to the solid cylinder.

The drag coefficient and shock stand-off distance Δ_{shock} are compared with the results of [27, 29, 37, 64, 65] in Table 3. The pressure coefficient along the cylinder surface and the pressure profile on the stagnation are compared and shown respectively in Figure 8.

Finally, the same simulation is performed using the simplified T-Passive model. As previously, the temperature field resulting from the two IBM models is very similar. An illustration is given in Figure 9 which is a center-line plot of the temperature. The position of the cylinder is illustrated by the two vertical gray lines.

A grid convergence study is performed by computing the error of the current simulations with a refined case ($D/\Delta_s x = 160$). Figure 10 shows a first-order convergence in temperature and velocity.

3.4 | Case 4: Supersonic Turbulent Flow on a Wedge

The next test case consists of a supersonic flow past a wedge of chord D and half-angle $\theta = 20°$. The behavior of the proposed IBM when dealing with sharp edges is investigated. Mach and Reynolds number are respectively $M_\infty = 2$ and $R_{e_\infty} = 5 \times 10^4$. R_{e_∞} is computed from the wedge length, noted D.

The triangle front is located at coordinates (0;0), on a rectangular domain of size $[-5\text{D}, 10\text{D}] \times [-5\text{D}, 5]$. A finer mesh is employed with $\Delta x = \text{D}/250$. As previously, the distance between two Lagrangian nodes is set up to be around Δx.

The boundary conditions on the surface of the wedge are no-slip and adiabatic $U_B = 0$, $\frac{\partial T}{\partial \vec{n}} = 0$. The T-Passive IBM is used, since it was shown that it is valid in this kind of configuration.

An oblique shock is formed at the tip of the wedge. Once the flow is steady, its angle is computed and compared with reference results [67, 68] in Table 4. The total drag coefficient is computed using the wedge length D as reference length, and compared to theoretical and numerical results of [67]. Here, let us notice that the theoretical drag value is evaluated using classical inviscid shock wave and Prandtl–Meyer theory, neglecting the viscous drag.

The Schlieren representation of the flow is obtained by computing the density gradient and is presented in Figure 11.

The pressure profile along the x-axis at $y = 0.88$D is also compared in Figure 12, and shows again that the present IBM agrees with the reference. Also, contrary to the observations of Ménez et al. [37], the downstream flow computed with IBM is accurate. In particular, the tail shock behind the wedge does converge towards the center line as expected. The difference between the IBM employed by [37] and the present work lies in the definition of the forcing term. Also, no calibration of the Lagrangian weight is mentioned, which might explain the lack of accuracy when dealing with sharp corners. In particular, the back face of the wedge may suffer from important leaks, degrading the backflow accuracy.

By enforcing a perfect reciprocity of the Spreading and Interpolation operators through calibration of the Lagrangian weight, the present immersed boundary method is indeed capable of enforcing strictly the desired boundary condition. The computed streamline illustrates how the present method reduces the fluid leaks through the immersed boundary, which yields accurate results for the simulation of the flow behind the wedge.

TABLE 3 | Cylinder at $M_\infty = 2.0$ and $Re_\infty = 300$: Comparison of the results.

		Δ_{shock}/D	C_d
Takahashi, Nonomura, and Fukuda [64]	Body-fitted	—	1.55
Takahashi, Nonomura, and Fukuda [64]	Ghost Nodes	—	1.53
Qiu et al. 2016 [27]	IBM	—	1.54
Riahi et al. 2018 [65]	IBM	0.69	1.51
Kumar, Sharma, and Singh [66]	IBM	—	1.56
Ménez et al. 2023 [37]	Penalization	0.73	1.59
Ménez et al. 2023 [37]	IBM	0.72	1.60
Present	Hybrid IBM-GN	0.72	1.60

3.5 | Case 5: Shock Wave Impact on a Cylinder

The final test case is the impact of a shock wave on an adiabatic circular cylinder. The goal is to analyze the accuracy of the T-Passive model in the case of rapid transient phenomena. A circular cylinder of diameter D is centered at coordinates (0,0) in a rectangular domain of size $[-3.75\text{D}, 3.75\text{D}] \times [-3.75\text{D}, 3.75\text{D}]$. The mesh size is fixed as $\Delta x = \text{D}/160$.

The fluid is initialized in order to generate a normal shock wave traveling at a Mach number of $M_s = 2.81$ in the x-positive direction. The two initial fluid states are set up as:

$$\begin{pmatrix} P_1 = 9.06 \times 10^5 \text{ Pa} \\ \rho_1 = 3.68 \text{ kg/m}^3 \\ U_1 = 7643.36 \text{ m/s} \end{pmatrix} \quad \begin{pmatrix} P_2 = 1.00 \times 10^5 \text{ Pa} \\ \rho_2 = 1.0 \text{ kg/m}^3 \\ U_2 = 0.0 \text{ m/s} \end{pmatrix} \tag{45}$$

The shock wave is reflected by the cylinder, forming a bow shock upstream. Along the surface of the solid, the reflected wave merges with the incident one, generating a Mach wave.

Trajectories of triple points, where the three shock waves meet, are observed and compared to [69, 70] in Figure 13.

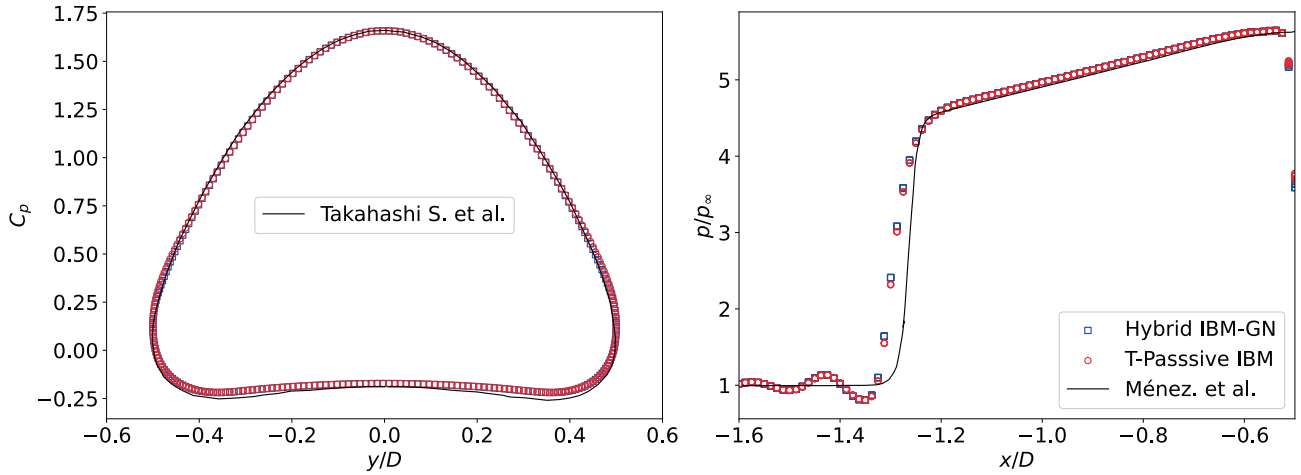

FIGURE 8 | Cylinder at Ma 2.0 Re 300, left: pressure Coefficient on the cylinder surface, right: pressure profile along the stagnation line in front of the cylinder.

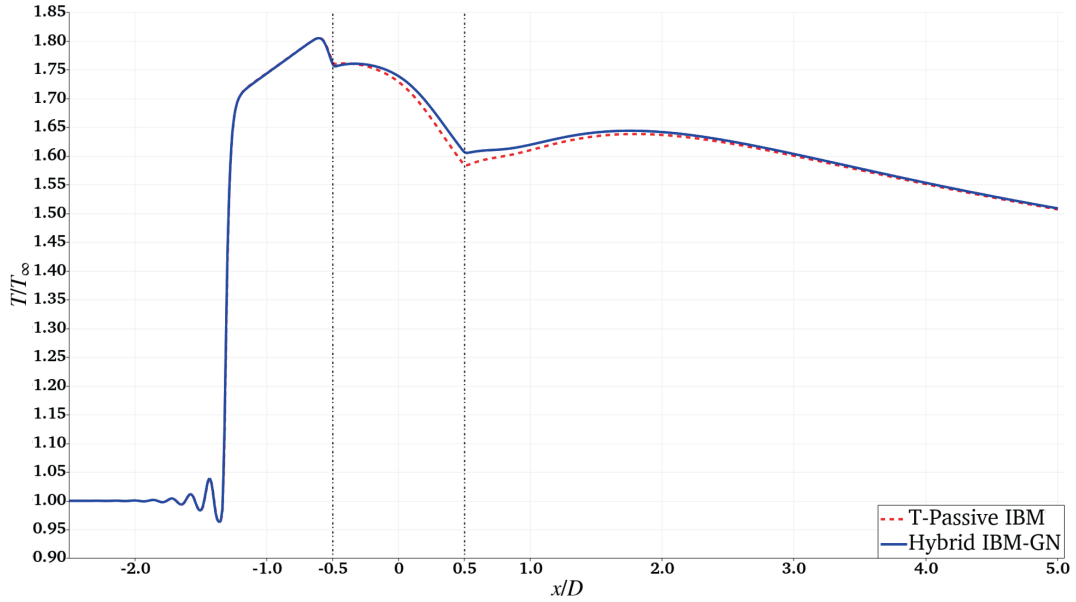

FIGURE 9 | 1D center line plot of the Temperature field around a cylinder at Ma 2.0 Re 300 (Position of the IB is symbolized by two vertical dash lines).

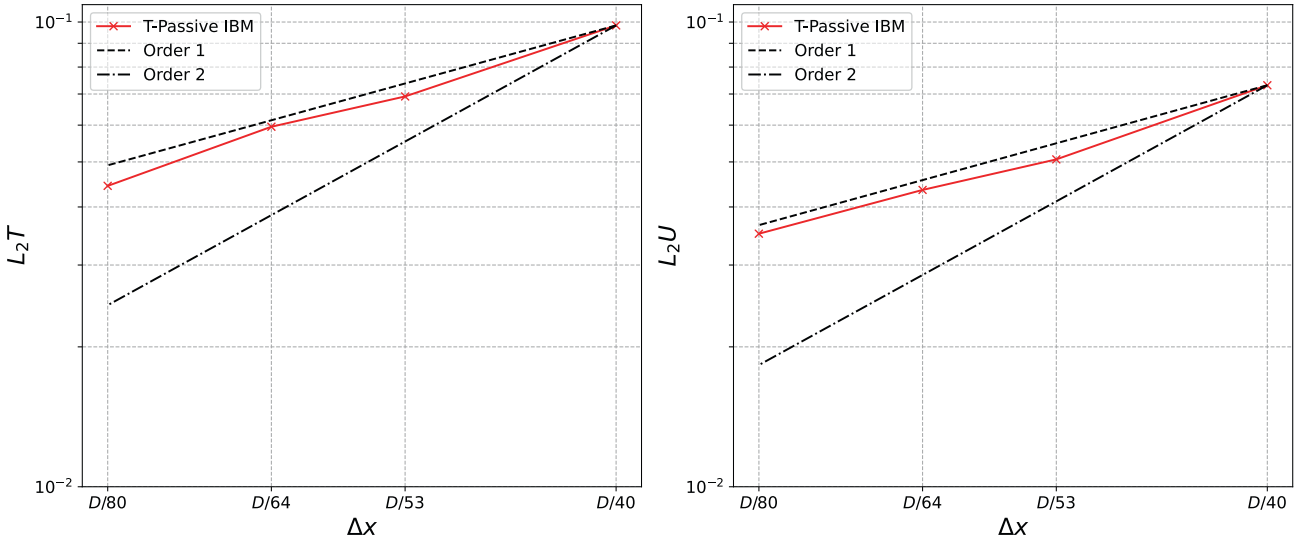

FIGURE 10 | Laminar bow shock convergence study.

TABLE 4 | Wedge $M_\infty = 2.0$: Comparison of the shock angle.

	β	C_d
Theory [67]	53.5∘	0.68
Boiron, Chiavassa, and Donat [67]	54.1∘	0.73
Abgrall, Beaugendre, and Dobrzynsk [68]	53.8∘	—
Ménez et al. [37] IBM	54.8∘	—
Ménez et al. [37] VP	54.5∘	—
Present	54.6∘	0.720

An excellent agreement is obtained between the obtained results and the reference experiments and simulations. Also, no major difference is observed between the hybrid strictly adiabatic GN/IBM and the T-Passive simplified model as shown in Figure 14. An analysis of the characteristic time ratios, proposed in 2.5.2, reveals why the simplified model performs well in this particular case:

$$\frac{\tau_{\text{conduction}}}{\tau_{\text{convection}}} > 10^4 \tag{46}$$

As suggested by the Figure 3, the Hybrid GN-IBM adds a non-negligible computational cost (measured on a mono processor simulation): the CPU time required for the Hybrid IBM-GN simulation was 1.56 times longer than the one using the T-Passive. Note that this gap is expected to increase with the number of Lagrangian nodes and the complexity of the solid shape.

4 | Conclusion

The present study proposes an immersed boundary model designed for compressible flows computed with the lattice Boltzmann method. A particular focus is given to enforcing Neumann boundary conditions for the temperature field.

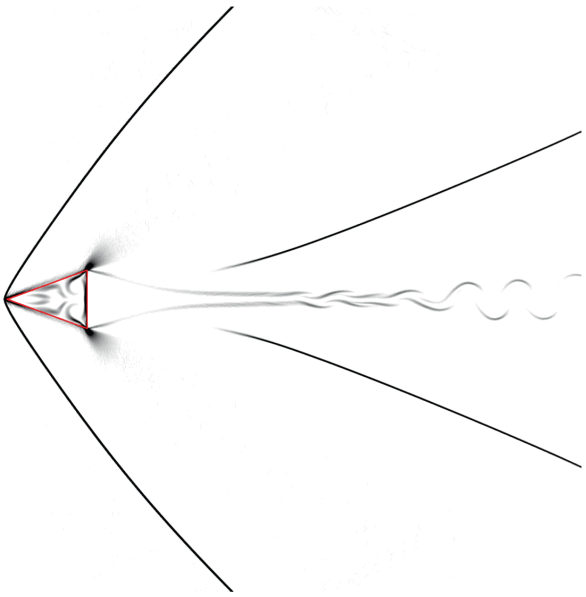

FIGURE 11 | Schlieren representation of the flow over a wedge at Ma 2.0.

In the usual case with the given fast transient dynamics context where the thermal coupling between the solid and the fluid is negligible, the boundary condition on the solid boundary is considered adiabatic. Particular attention is given to the best method to enforce such a condition, with a reference solution obtained by an original combination of diffuse forcing for the velocity and ghost nodes approach for the temperature. A simplified IBM model for temperature management with adiabatic boundary, taking into account the different time scales between convection and conduction and referred to as "T-Passive" IBM, is also explored and carefully assessed for the sake of computational efficiency.

Both models are validated and compared in various cases, ranging from laminar subsonic to turbulent supersonic regimes. The results demonstrate the capacity of the simplified T-Passive model to yield similar results compared to the strictly adiabatic GN method. However, the T-Passive model is only valid when thermal fluid-structure coupling can be neglected. In high Mach flows, such as hypersonic regimes where temperature gradients are extreme, this assumption may not be valid anymore. Additionally, as suggested in Section 2.5.2, even in cases where the FS heat exchange is neglected, if the Reynolds-Prandtl (RePr) number is too low, the T-Passive model may fail to model accurately the FS thermal interface, limiting its applicability. The proposed IBM model coupled to a compressible flow LBM solver is demonstrated to give promising performances, to tackle fast transient flows with fluid-structure interaction.

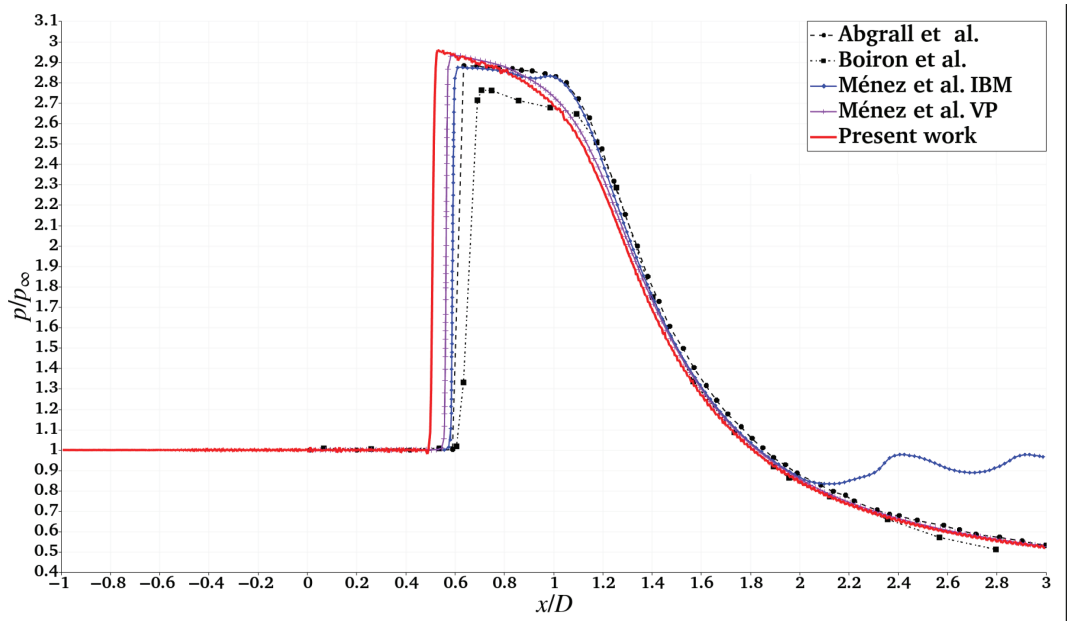

FIGURE 12 | Pressure profile on an horizontal line at $y/d = 0.88$ – flow over a wedge at Ma 2.0.

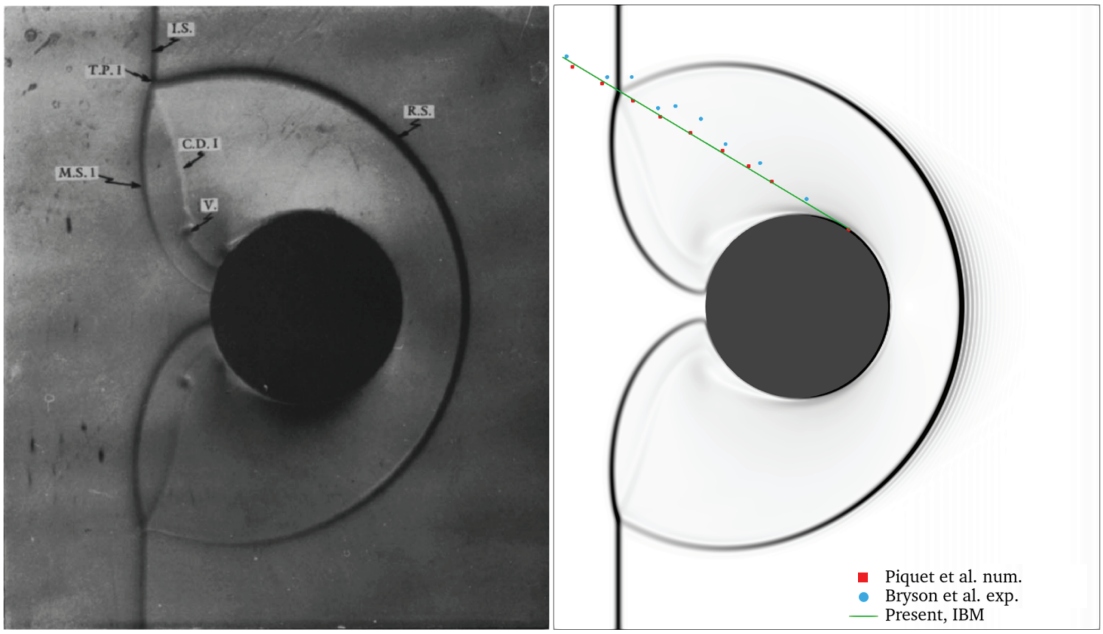

FIGURE 13 | Schlieren representation of the impact of a shock wave on a circular cylinder at Ma 2.81 Left: experimental result of [69] Right: present numerical result and comparison of triple point trajectories with [69, 70].

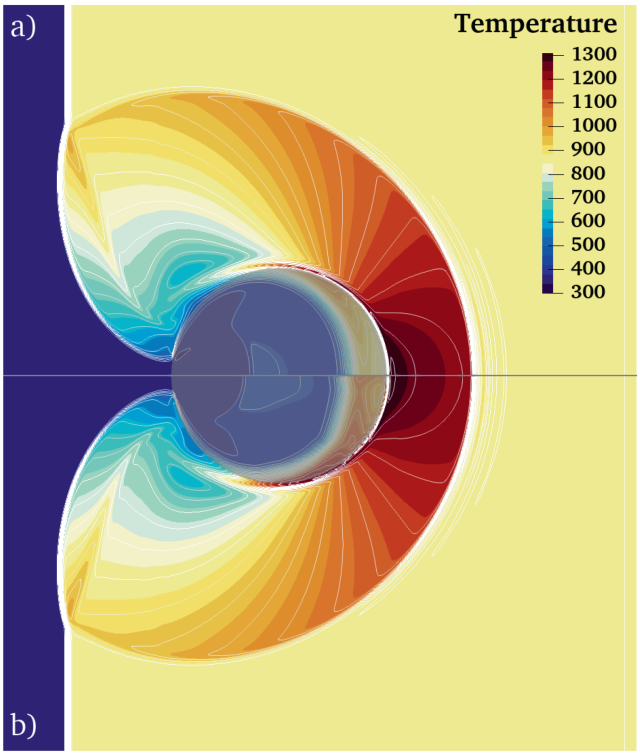

FIGURE 14 | Temperature field of the impact of a shock wave on a circular cylinder at Ma 2.81 Comparison of a) the T-Passive model and b) the Hybrid IBM-GN model.

Future work will focus on the coupling of the present fluid solver with a solid structure solver in order to demonstrate the applicability of the present IBM approach in coupled FSI scenarios. Challenges related to the stability and accuracy of the coupling scheme should be addressed in the framework of fast transient FSI.

Acknowledgments

This work was performed using the ProLB software. Centre de Calcul Intensif d'Aix-Marseille is acknowledged for granting access to their high-performance computing resources. This project was provided with computer and storage resources by GENCI at TGCC thanks to the grant 2023-A0152A07679 on the supercomputer Joliot Curie SKL/ROME partition. Isabelle Cheylan, Jérôme Jacob and Song Zhao are gratefully acknowledged for the insightful discussion they proposed and the technical expertise they offered.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. C. W. Hirt, A. A. Amsden, and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for all Flow Speeds," *Journal of Computational Physics* 14, no. 3 (1974): 227–253, https://linkinghub.elsevier.com/retrieve/pii/0021999174900515.

2. J. Donea, A. Huerta, J. P. Ponthot, and A. Rodríguez-Ferran, "Arbitrary Lagrangian–Eulerian Methods: Part 1 Fundamentals," in *Encyclopedia of Computational Mechanics*, 1st ed. (Hoboken, NJ: Wiley, 2004), https://onlinelibrary.wiley.com/doi/10.1002/0470091355.ecm009.

3. J. Donea, S. Giuliani, and J. P. Halleux, "An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Interactions," *Computer Methods in Applied Mechanics and Engineering* 33, no. 1–3 (1982): 689–723, https://linkinghub.elsevier.com/retrieve/pii/0045782582901281.

4. E. Kuhl, S. Hulshoff, and R. De Borst, "An Arbitrary Lagrangian Eulerian Finite-Element Approach for Fluid–Structure Interaction Phenomena," *International Journal for Numerical Methods in Engineering* 57, no. 1 (2003): 117–142, https://onlinelibrary.wiley.com/doi/10.1002/nme.749.

5. A. J. Barlow, P. H. Maire, W. J. Rider, R. N. Rieben, and M. J. Shashkov, "Arbitrary Lagrangian–Eulerian Methods for Modeling High-Speed

Compressible Multimaterial Flows," *Journal of Computational Physics* 322 (2016): 603–665, https://linkinghub.elsevier.com/retrieve/pii/S0021999116302807.

6. C. Farhat, K. G. Wang, A. Main, et al., "Dynamic Implosion of Underwater Cylindrical Shells: Experiments and Computations," *International Journal of Solids and Structures* 50, no. 19 (2013): 2943–2961, https://linkinghub.elsevier.com/retrieve/pii/S002076831300200X.

7. O. Jamond and V. Faucher, "Regularized Immersed Boundary-Type Formulation for Fast Transient Dynamics With Fluid-Structure Interaction," *Advances in Engineering Software* 108 (2017): 1–23, https://linkinghub.elsevier.com/retrieve/pii/S0965997816304148.

8. C. S. Peskin, "Flow Patterns Around Heart Valves: A Numerical Method," *Journal of Computational Physics* 10, no. 2 (1972): 252–271, https://www.sciencedirect.com/science/article/pii/0021999172900654.

9. R. P. Beyer and R. J. LeVeque, "Analysis of a One-Dimensional Model for the Immersed Boundary Method," *SIAM Journal on Numerical Analysis* 29, no. 2 (1992): 332–364, https://epubs.siam.org/doi/abs/10.1137/0729022.

10. E. M. Saiki and S. Biringen, "Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method," *Journal of Computational Physics* 123, no. 2 (1996): 450–465, https://linkinghub.elsevier.com/retrieve/pii/S0021999196900364.

11. M. C. Lai and C. S. Peskin, "An Immersed Boundary Method With Formal Second-Order Accuracy and Reduced Numerical Viscosity," *Journal of Computational Physics* 160, no. 2 (2000): 705–719, https://linkinghub.elsevier.com/retrieve/pii/S0021999100964830.

12. D. Goldstein, R. Handler, and L. Sirovich, "Modeling a no-Slip Flow Boundary With an External Force Field," *Journal of Computational Physics* 105, no. 2 (1993): 354–366, https://linkinghub.elsevier.com/retrieve/pii/S0021999183710818.

13. J. Mohd-Yusof, *Combined Immersed Boundary Bspline Methods for Simulations of Flow in Complex Geometries* (Center for Turbulence Research: Annual Research Briefs, 1997).

14. E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations," *Journal of Computational Physics* 161, no. 1 (2000): 35–60, https://linkinghub.elsevier.com/retrieve/pii/S0021999100964842.

15. R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, "A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)," *Journal of Computational Physics* 152, no. 2 (1999): 457–492, https://linkinghub.elsevier.com/retrieve/pii/S0021999199962368.

16. R. P. Fedkiw, "Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation With the Ghost Fluid Method," *Journal of Computational Physics* 175, no. 1 (2002): 200–224, https://linkinghub.elsevier.com/retrieve/pii/S0021999101969359.

17. Y. H. Tseng and J. H. Ferziger, "A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry," *Journal of Computational Physics* 192, no. 2 (2003): 593–623, https://linkinghub.elsevier.com/retrieve/pii/S0021999103004108.

18. R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, and A. Von Loebbecke, "A Versatile Sharp Interface Immersed Boundary Method for Incompressible Flows With Complex Boundaries," *Journal of Computational Physics* 227, no. 10 (2008): 4825–4852, https://linkinghub.elsevier.com/retrieve/pii/S0021999108000235.

19. C. Farhat, A. Rallu, and S. Shankaran, "A Higher-Order Generalized Ghost Fluid Method for the Poor for the Three-Dimensional Two-Phase Flow Computation of Underwater Implosions," *Journal of Computational Physics* 227, no. 16 (2008): 7674–7700, https://linkinghub.elsevier.com/retrieve/pii/S0021999108002568.

20. R. Mittal and G. Iaccarino, "Immersed Boundary Methods," *Annual Review of Fluid Mechanics* 37, no. 1 (2005): 239–261, https://www.annualreviews.org/doi/10.1146/annurev.fluid.37.061903.175743.

21. Z. G. Feng and E. E. Michaelides, "The Immersed Boundary-Lattice Boltzmann Method for Solving Fluid–Particles Interaction Problems," *Journal of Computational Physics* 195, no. 2 (2004): 602–628, https://linkinghub.elsevier.com/retrieve/pii/S0021999103005758.

22. J. Lu, H. Han, B. Shi, and Z. Guo, "Immersed Boundary Lattice Boltzmann Model Based on Multiple Relaxation Times," *Physical Review E* 85 (2012): 016711 https://link.aps.org/doi/10.1103/PhysRevE.85.016711.

23. S. Gsell, U. D'Ortona, and J. Favier, "Explicit and Viscosity-Independent Immersed-Boundary Scheme for the Lattice Boltzmann Method," *Physical Review E* 100, no. 3 (2019): 033306, https://link.aps.org/doi/10.1103/PhysRevE.100.033306.

24. M. Abaszadeh, A. Safavinejad, A. Amiri Delouei, and H. Amiri, "Analysis of Radiative Heat Transfer in Two-Dimensional Irregular Geometries by Developed Immersed Boundary–Lattice Boltzmann Method," *Journal of Quantitative Spectroscopy and Radiative Transfer* 280 (2022): 108086, https://linkinghub.elsevier.com/retrieve/pii/S0022407322000231.

25. B. Afra, A. A. Delouei, and A. Tarokh, "Flow-Induced Locomotion of a Flexible Filament in the Wake of a Cylinder in Non-Newtonian Flows," *International Journal of Mechanical Sciences* 234 (2022): 107693, https://linkinghub.elsevier.com/retrieve/pii/S0020740322005744.

26. R. Mittal and R. Bhardwaj, "Immersed Boundary Methods For Thermofluids Problems," *Annual Review of Heat Transfer* 24, no. 1 (2022): 33–70, https://www.dl.begellhouse.com/references/5756967540dd1b03,3ae07302147f45b7,2e0aa8eb280d726b.html.

27. Y. L. Qiu, C. Shu, J. Wu, Y. Sun, L. M. Yang, and T. Q. Guo, "A Boundary Condition-Enforced Immersed Boundary Method for Compressible Viscous Flows," *Computers & Fluids* 136 (2016): 104–113, https://linkinghub.elsevier.com/retrieve/pii/S004579301630189X.

28. J. Wu and C. Shu, "Implicit Velocity Correction-Based Immersed Boundary-Lattice Boltzmann Method and Its Applications," *Journal of Computational Physics* 228, no. 6 (2009): 1963–1979, https://linkinghub.elsevier.com/retrieve/pii/S0021999108006116.

29. Y. Sun, L. Yang, C. Shu, and Y. Chen, "A Diffuse-Interface Immersed Boundary Method for Simulation of Compressible Viscous Flows With Stationary and Moving Boundaries," *International Journal for Numerical Methods in Fluids* 92, no. 3 (2020): 149–168, https://onlinelibrary.wiley.com/doi/10.1002/fld.4777.

30. N. Zhang, Z. C. Zheng, and S. Eckels, "Study of Heat-Transfer on the Surface of a Circular Cylinder in Flow Using an Immersed-Boundary Method," *International Journal of Heat and Fluid Flow* 29 (2008): 1558–1566.

31. W. Ren, C. Shu, and W. Yang, "An Efficient Immersed Boundary Method for Thermal Flow Problems With Heat Flux Boundary Conditions," *International Journal of Heat and Mass Transfer* 64 (2013): 694–705, https://linkinghub.elsevier.com/retrieve/pii/S0017931013004109.

32. Y. Wang, C. Shu, and L. M. Yang, "Boundary Condition-Enforced Immersed Boundary-Lattice Boltzmann Flux Solver for Thermal Flows With Neumann Boundary Conditions," *Journal of Computational Physics* 306 (2016): 237–252, https://linkinghub.elsevier.com/retrieve/pii/S002199911500786X.

33. A. A. Hosseinjani and A. Ashrafizadeh, "An Extended Iterative Direct-Forcing Immersed Boundary Method in Thermo-Fluid Problems With Dirichlet or Neumann Boundary Conditions," *Journal of Central South University* 24, no. 1 (2017): 137–154, http://link.springer.com/10.1007/s11771-017-3416-x.

34. T. Guo, E. Shen, Z. Lu, Y. Wang, and L. Dong, "Implicit Heat Flux Correction-Based Immersed Boundary-Finite Volume Method for

Thermal Flows With Neumann Boundary Conditions," *Journal of Computational Physics* 386 (2019): 64–83, https://linkinghub.elsevier.com/retrieve/pii/S0021999119301330.

35. Y. Chen, C. Shu, Y. Sun, L. M. Yang, and Y. Wang, "A Diffuse Interface IBM for Compressible Flows With Neumann Boundary Condition," *International Journal of Modern Physics B* 34, no. 14n16 (2020): 2040070, https://www.worldscientific.com/doi/abs/10.1142/S0217979220400706.

36. H. Yu and C. Pantano, "An Immersed Boundary Method With Implicit Body Force for Compressible Viscous Flow," *Journal of Computational Physics* 459 (2022): 111125, https://linkinghub.elsevier.com/retrieve/pii/S0021999122001875.

37. L. Ménez, P. Parnaudeau, M. Beringhier, and E. Goncalves Da Silva, "Assessment of Volume Penalization and Immersed Boundary Methods for Compressible Flows With Various Thermal Boundary Conditions," *Journal of Computational Physics* 493 (2023): 112465, https://linkinghub.elsevier.com/retrieve/pii/S0021999123005600.

38. B. Wu, J. Lu, H. Lee, C. Shu, and M. Wan, "An Explicit Boundary Condition-Enforced Immersed Boundary-Reconstructed Thermal Lattice Boltzmann Flux Solver for Thermal–Fluid–Structure Interaction Problems With Heat Flux Boundary Conditions," *Journal of Computational Physics* 485 (2023): 112106, https://linkinghub.elsevier.com/retrieve/pii/S0021999123002012.

39. R. Boukharfane, F. H. Eugênio Ribeiro, Z. Bouali, and A. Mura, "A Combined Ghost-Point-Forcing/Direct-Forcing Immersed Boundary Method (IBM) for Compressible Flow Simulations," *Computers & Fluids* 162 (2018): 91–112, https://linkinghub.elsevier.com/retrieve/pii/S0045793017304267.

40. J. Yang, "Sharp Interface Direct Forcing Immersed Boundary Methods: A Summary of Some Algorithms and Applications," *Journal of Hydrodynamics* 28 (2016): 713–730.

41. A. Chaudhuri, A. Hadjadj, and A. Chinnayya, "On the Use of Immersed Boundary Methods for Shock/Obstacle Interactions," *Journal of Computational Physics* 230, no. 5 (2011): 1731–1748, https://linkinghub.elsevier.com/retrieve/pii/S0021999110006248.

42. J. Ho and C. Farhat, "Discrete Embedded Boundary Method With Smooth Dependence on the Evolution of a Fluid-Structure Interface," *International Journal for Numerical Methods in Engineering* 122, no. 19 (2021): 5353–5383, https://onlinelibrary.wiley.com/doi/10.1002/nme.6455.

43. J. H. Seo and R. Mittal, "A Sharp-Interface Immersed Boundary Method With Improved Mass Conservation and Reduced Spurious Pressure Oscillations," *Journal of Computational Physics* 230, no. 19 (2011): 7347–7363, https://linkinghub.elsevier.com/retrieve/pii/S0021999111003524.

44. J. Lee and D. You, "An Implicit Ghost-Cell Immersed Boundary Method for Simulations of Moving Body Problems With Control of Spurious Force Oscillations," *Journal of Computational Physics* 233 (2013): 295–314, https://linkinghub.elsevier.com/retrieve/pii/S0021999112005128.

45. M. Belliard, M. Chandesris, J. Dumas, et al., "An Analysis and an Affordable Regularization Technique for the Spurious Force Oscillations in the Context of Direct-Forcing Immersed Boundary Methods," *Computers & Mathematics With Applications* 71, no. 5 (2016): 1089–1113, https://linkinghub.elsevier.com/retrieve/pii/S0898122116300207.

46. D. M. C. Martins, D. M. S. Albuquerque, and J. C. F. Pereira, "Continuity Constrained Least-Squares Interpolation for SFO Suppression in Immersed Boundary Methods," *Journal of Computational Physics* 336 (2017): 608–626, https://linkinghub.elsevier.com/retrieve/pii/S0021999117301195.

47. S. K. Kang and Y. A. Hassan, "A Comparative Study of Direct-Forcing Immersed Boundary-Lattice Boltzmann Methods for Stationary Complex Boundaries," *International Journal for Numerical Methods in Fluids* 66, no. 9 (2011): 1132–1158, https://onlinelibrary.wiley.com/doi/10.1002/fld.2304.

48. J. Qin, X. Yang, and Z. Li, "Hybrid Diffuse and Sharp Interface Immersed Boundary Methods for Particulate Flows in the Presence of Complex Boundaries," *Communications in Computational Physics* 31, no. 4 (2022): 1242–1271.

49. Z. Guo, C. Zheng, and B. Shi, "Discrete Lattice Effects on the Forcing Term in the Lattice Boltzmann Method," *Physical Review* 65, no. 4 (2002 Apr): 046308, https://link.aps.org/doi/10.1103/PhysRevE.65.046308.

50. G. Farag, S. Zhao, T. Coratger, P. Boivin, G. Chiavassa, and P. Sagaut, "A Pressure-Based Regularized Lattice-Boltzmann Method for the Simulation of Compressible Flows," *Physics of Fluids* 32, no. 6 (2020): 066106, http://aip.scitation.org/doi/10.1063/5.0011839.

51. J. Jacob, O. Malaspinas, and P. Sagaut, "A New Hybrid Recursive Regularised Bhatnagar–Gross–Krook Collision Model for Lattice Boltzmann Method-Based Large Eddy Simulation," *Journal of Turbulence* 19, no. 11–12 (2018): 1051–1076, https://www.tandfonline.com/doi/full/10.1080/14685248.2018.1540879.

52. T. Coratger, G. Farag, S. Zhao, P. Boivin, and P. Sagaut, "Large-Eddy Lattice-Boltzmann Modeling of Transonic Flows," *Physics of Fluids* 33, no. 11 (2), http://doi.org/10.1063/5.0064944.

53. G. Wissocq, T. Coratger, G. Farag, S. Zhao, P. Boivin, and P. Sagaut, "Restoring the Conservativity of Characteristic-Based Segregated Models: Application to the Hybrid Lattice Boltzmann Method," *Physics of Fluids* 34, no. 4 (2022 Apr): 046102, https://aip.scitation.org/doi/10.1063/5.0083377.

54. I. Cheylan, T. Fringand, J. Jacob, and J. Favier, "Analysis of the Lagrangian Weight in the Direct-Forcing Immersed Boundary Method for Fluid-Structure Interaction," *SSRN Electronic Journal* 492 (2022): 112418, https://www.ssrn.com/abstract=4222782.

55. C. Norberg, "Flow Around A Circular Cylinder: Aspects of Fluctuating Lift," *Journal of Fluids and Structures* 15, no. 3–4 (2001): 459–469, https://linkinghub.elsevier.com/retrieve/pii/S0889974600903670.

56. C. Norberg, "Fluctuating Lift on a Circular Cylinder: Review and New Measurements," *Journal of Fluids and Structures* 17, no. 1 (2003): 57–96, https://linkinghub.elsevier.com/retrieve/pii/S0889974602000993.

57. L. Qu, C. Norberg, L. Davidson, S. H. Peng, and F. Wang, "Quantitative Numerical Analysis of Flow Past a Circular Cylinder at Reynolds Number Between 50 and 200," *Journal of Fluids and Structures* 39 (2013): 347–370, https://linkinghub.elsevier.com/retrieve/pii/S0889974613000388.

58. C. Liu, X. Zheng, and C. H. Sung, "Preconditioned Multigrid Methods for Unsteady Incompressible Flows," *Journal of Computational Physics* 139, no. 1 (1998): 35–57, https://linkinghub.elsevier.com/retrieve/pii/S0021999197958599.

59. C. Y. Zhou, R. M. C. So, and K. Lam, "Vortex-Induced Vibrations of an Elastic Circular Cylinder," *Journal of Fluids and Structures* 13, no. 2 (1999): 165–189, https://linkinghub.elsevier.com/retrieve/pii/S0889974698901955.

60. J. Kim, D. Kim, and H. Choi, "An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries," *Journal of Computational Physics* 171, no. 1 (2001): 132–150, https://linkinghub.elsevier.com/retrieve/pii/S0021999101967786.

61. R. Bourguet and J. D. Lo, "Flow-Induced Vibrations of a Rotating Cylinder," *Journal of Fluid Mechanics* 740 (2014): 342–380, https://www.cambridge.org/core/product/identifier/S0022112013006654/type/journal_article.

62. S. Gsell and J. Favier, "Direct-Forcing Immersed-Boundary Method: A Simple Correction Preventing Boundary Slip Error," *Journal of Computational Physics* 435 (2021): 110265, https://linkinghub.elsevier.com/retrieve/pii/S0021999121001601.

63. H. Wolfgang Liepmann and A. Roshko, *Elements of Gas Dynamics* (Mineola, NY: Dover Publications Inc, 2001).

64. S. Takahashi, T. Nonomura, and K. Fukuda, "A Numerical Scheme Based on an Immersed Boundary Method for Compressible Turbulent Flows With Shocks: Application to Two-Dimensional Flows Around Cylinders," *Journal of Applied Mathematics* 2014 (2014): 1–21, http://www.hindawi.com/journals/jam/2014/252478/.

65. H. Riahi, M. Meldi, J. Favier, E. Serre, and E. Goncalves, "A Pressure-Corrected Immersed Boundary Method for the Numerical Simulation of Compressible Flows," *Journal of Computational Physics* 374 (2018): 361–383, https://linkinghub.elsevier.com/retrieve/pii/S0021999118304947.

66. V. Kumar, A. Sharma, and R. K. Singh, "Central Upwind Scheme Based Immersed Boundary Method for Compressible Flows Around Complex Geometries," *Computers & Fluids* 196 (2020): 104349, https://linkinghub.elsevier.com/retrieve/pii/S0045793019303081.

67. O. Boiron, G. Chiavassa, and R. Donat, "A High-Resolution Penalization Method for Large Mach Number Flows in the Presence of Obstacles," *Computers & Fluids* 38, no. 3 (2009): 703–714, https://linkinghub.elsevier.com/retrieve/pii/S0045793008001424.

68. R. Abgrall, H. Beaugendre, and C. Dobrzynski, "An Immersed Boundary Method Using Unstructured Anisotropic Mesh Adaptation Combined With Level-Sets and Penalization Techniques," *Journal of Computational Physics* 257 (2014): 83–101, https://linkinghub.elsevier.com/retrieve/pii/S0021999113005962.

69. A. E. Bryson and R. W. F. Gross, "Diffraction of Strong Shocks by Cones, Cylinders, and Spheres," *Journal of Fluid Mechanics* 10, no. 1 (1961): 1–16, https://www.cambridge.org/core/product/identifier/S0022112061000019/type/journal_article.

70. A. Piquet, O. Roussel, and A. Hadjadj, "A Comparative Study of Brinkman Penalization and Direct-Forcing Immersed Boundary Methods for Compressible Viscous Flows," *Computers & Fluids* 136 (2016): 272–284, https://linkinghub.elsevier.com/retrieve/pii/S0045793016301888.

71. O. Malaspinas, "Increasing Stability and Accuracy of the Lattice Boltzmann Scheme: Recursivity and Regularization," 2015, http://arxiv.org/abs/1505.06900.

Appendix A

Lattice-Boltzmann Solver

Details on the Hyrbid LBM solver are given in the following appendix. First, let us recall the Boltzmann Equation (8) accounting for the presence of body force as:

$$\frac{\partial f}{\partial t} + c_\alpha \frac{\partial f}{\partial x_\alpha} + \frac{F_\alpha}{\rho}\frac{\partial f}{\partial c_\alpha} = \Omega(f) \tag{A1}$$

where c denotes the particle velocity and $\Omega(f)$ is the collision operator and **F** a generic forcing term.

In the framework of an LBM solver, (8) is discretized in the physical space, in time, and in the velocity space by defining a set of c_i particle velocities. Here the usual D3Q19* lattice is used for which

$$\mathbf{c}_{i,x} = (0,1,1,0,-1,-1,-1,0,1,0,1,-1,1,-1,0,-1,1,0)\frac{\Delta x}{\Delta t} \tag{A2}$$

$$\mathbf{c}_{i,y} = (0,0,1,1,1,0,-1,-1,-1,0,1,-1,-1,1,1,-1,0,0)\frac{\Delta x}{\Delta t} \tag{A3}$$

$$\mathbf{c}_{i,z} = (0,0,0,0,0,0,0,0,0,1,-1,1,-1,1,-1,1,-1,1,-1)\frac{\Delta x}{\Delta t} \tag{A4}$$

As shown earlier in Equation (9), using a unified Hybrid Recursive Regularized approach from [50, 51, 53] the obtained scheme can be written in the general formulation:

$$\begin{cases} f_i^{\text{coll}}(x,t) = f_i^{\text{eq}}(x,t) + \left(1 - \frac{\Delta t}{\tau}\right) f_i^{\text{neq}}(x,t) + \frac{\Delta t}{2} F_i(x,t) \\ f_i(x, t+\Delta t) = f_i^{\text{coll}}(x - c_i \Delta t, t) \end{cases}$$

With τ being related to the fluid dynamic viscosity μ as:

$$\mu + \rho \nu_{sc} = \left(\tau - \frac{\Delta t}{2}\right)\rho c_s^2 \tag{A5}$$

An artificial viscosity ν_{sc} is used to handle discontinuities and is computed as:

$$\begin{aligned} \nu_{sc} &= s_c \frac{1}{3}\left|\delta_{p_x} + \delta_{p_y} + \delta_{p_z}\right| \\ \delta_{p_\alpha} &= \frac{p(x-\Delta x_\alpha, t) - 2p(x,t) + p(x+\Delta x_\alpha, t)}{p(x-\Delta x_\alpha, t) + 2p(x,t) + p(x+\Delta x_\alpha, t)} \end{aligned} \tag{A6}$$

where s_c is a free parameter that can be tuned to adjust the influence of the shock sensor.

Regarding the computation of f_i^{eq}, an expansion on the basis of Hermite polynomials is used:

$$\begin{aligned} f_i^{\text{eq}} = \omega_i \Bigg(& H_i^{(0)}(\mathbf{c}_i) a_{(\text{eq})}^{(0)} + \frac{H_{i,\alpha}^{(1)}(\mathbf{c}_i) a_{(\text{eq}),\alpha}^{(1)}}{c_s^2} + \frac{H_{i,\alpha\beta}^{(2)}(\mathbf{c}_i) a_{(\text{eq}),\alpha\beta}^{(2)}}{2c_s^4} \\ & + \frac{1}{6c_s^6}\Big[3(H_{i,xxy}^{(3)} + H_{i,yzz}^{(3)})(a_{(\text{eq}),xxy}^{(3)} + a_{(\text{eq}),yzz}^{(3)}) \\ & + (H_{i,xxy}^{(3)} - H_{i,yzz}^{(3)})(a_{(\text{eq}),xxy}^{(3)} - a_{(\text{eq}),yzz}^{(3)}) \\ & + 3(H_{i,xzz}^{(3)} + H_{i,xyy}^{(3)})(a_{(\text{eq}),xzz}^{(3)} + a_{(\text{eq}),xyy}^{(3)}) \\ & + (H_{i,xzz}^{(3)} - H_{i,xyy}^{(3)})(a_{(\text{eq}),xzz}^{(3)} - a_{(\text{eq}),xyy}^{(3)}) \\ & + 3(H_{i,yyz}^{(3)} + H_{i,xxz}^{(3)})(a_{(\text{eq}),yyz}^{(3)} + a_{(\text{eq}),xxz}^{(3)}) \\ & + (H_{i,yyz}^{(3)} - H_{i,xxz}^{(3)})(a_{(\text{eq}),yyz}^{(3)} - a_{(\text{eq}),xxz}^{(3)}) \Big] \Bigg) \end{aligned} \tag{A7}$$

where the Hermite polynomials in the velocity space are:

$$H_i^{(0)} = 1 \tag{A8}$$

$$H_i^{(1)} = c_{i\alpha} \tag{A9}$$

$$H_i^{(2)} = c_{i\alpha}c_{i\beta} - c_s^2 \delta_{\alpha\beta} \tag{A10}$$

$$H_i^{(3)} = c_{i\alpha}c_{i\beta}c_{i\gamma} - c_s^2(c_{i\alpha}\delta_{\beta\gamma} + c_{i\beta}\delta_{\gamma\alpha} + c_{i\gamma}\delta_{\alpha\beta}) \tag{A11}$$

And the Hermite coefficients are defined as:

$$a_{(\text{eq})}^{(0)} = \rho + \frac{\omega_i - \delta_{0i}}{\omega_i}\rho(\theta - 1) \tag{A12}$$

$$a_{(\text{eq}),\alpha}^{(1)} = \rho u_\alpha \tag{A13}$$

$$a_{(\text{eq}),\alpha\beta}^{(2)} = \rho u_\alpha u_\beta \tag{A14}$$

$$a_{(\text{eq}),\alpha\beta\gamma}^{(3)} = \rho u_\alpha u_\beta u_\gamma \tag{A15}$$

Similarly, a definition of f_i^{neq} expanded up to third order is proposed as:

$$f_i^{\text{neq}} = \omega_i \left(\frac{H_{i,\alpha\beta}^{(2)}(\mathbf{c}_i) a_{(\text{neq}),\alpha\beta}^{(2)}}{2c_s^4} + \frac{1}{6c_s^6} \right.$$

$$\Big[3(H^{(3)}_{i,xxy}+H^{(3)}_{i,yzz})(a^{(3)}_{(\mathrm{neq}),xxy}+a^{(3)}_{(\mathrm{neq}),yzz}) + (H^{(3)}_{i,xxy}-H^{(3)}_{i,yzz})(a^{(3)}_{(\mathrm{neq}),xxy}-a^{(3)}_{(\mathrm{neq}),yzz}) + 3(H^{(3)}_{i,xzz}+H^{(3)}_{i,xyy})(a^{(3)}_{(\mathrm{neq}),xzz}+a^{(3)}_{(\mathrm{neq}),xyy}) + (H^{(3)}_{i,xzz}-H^{(3)}_{i,xyy})(a^{(3)}_{(\mathrm{neq}),xzz}-a^{(3)}_{(\mathrm{neq}),xyy}) + 3(H^{(3)}_{i,yyz}+H^{(3)}_{i,xxz})(a^{(3)}_{(\mathrm{neq}),yyz}+a^{(3)}_{(\mathrm{neq}),xxz}) + (H^{(3)}_{i,yyz}-H^{(3)}_{i,xxz})(a^{(3)}_{(\mathrm{neq}),yyz}-a^{(3)}_{(\mathrm{neq}),xxz})\Big]\Big) \tag{A16}$$

The second order off-equilibrium Hermite coefficient $a^{(2)}_{(\mathrm{neq}),\alpha\beta}$ is computed according to the Hybrid Recursive Regularized model as:

$$a^{(2)}_{(\mathrm{neq}),\alpha\beta} = \sigma(\tilde{a}^{(2),(PR)}_{(\mathrm{neq}),\alpha\beta}) + (1-\sigma)(a^{(2),(FD)}_{(\mathrm{neq}),\alpha\beta}) \tag{A17}$$

Where $\tilde{a}^{(2),(PR)}_{(\mathrm{neq}),\alpha\beta}$ is defined by the projection of the distribution function onto the second-order Hermite polynomials, used in a traceless form:

$$\tilde{a}^{(2),(PR)}_{(\mathrm{neq}),\alpha\beta} = a^{(2),(PR)}_{(\mathrm{neq}),\alpha\beta} - \frac{\delta_{\alpha\beta}}{\mathrm{D}} a^{(2),(PR)}_{(\mathrm{neq}),\gamma\gamma} \tag{A18}$$

$$a^{(2),(PR)}_{(\mathrm{neq}),\alpha\beta} = \sum_i H^{(2)}_{i,\alpha\beta}(f_i - f_i^{eq}) \tag{A19}$$

And where $\tilde{a}^{(2),(FD)}_{(\mathrm{neq}),\alpha\beta}$ is computed directly from the macroscopic variable as shown in [51]. Finally, the third order moment is recursively computed from $a^{(2),(PR)}_{(\mathrm{neq}),\alpha\beta}$ as proposed by [71]:

$$a^{(3)}_{(\mathrm{neq}),\alpha\beta\gamma} = u_\alpha a^{(2)}_{(\mathrm{neq}),\beta\gamma} + u_\beta a^{(2)}_{(\mathrm{neq}),\alpha\gamma} + u_\gamma a^{(2)}_{(\mathrm{neq}),\alpha\beta} \tag{A20}$$

The generic forcing term F_i in (9) may include various corrections or external forcing. In the present method, it includes the immersed boundary source term from the momentum equation f^{IB}_α. Here the method proposed by Guo, Zheng, and Shi [49] is used, that relates f^{IB}_α which the IBM forcing term in the NS Equation (1) to F_i^{IB} the IBM forcing term in the Boltzmann Equation (8) as:

$$F_i^{IB} = \left(1-\frac{\Delta t}{2\tau}\right)\omega_i\left(\frac{c_{i,\alpha}-u_\alpha}{c_s^2} + \frac{c_{i,\beta}u_\beta}{c_s^4}c_{i,\alpha}\right) f^{\mathrm{IB}}_\alpha \tag{A21}$$

Simultaneously, the total energy equation is advanced in time using finite difference discretization as:

$$\frac{(\rho \mathrm{E})^{t+\Delta t}-(\rho \mathrm{E})^t}{\Delta t} + \frac{\partial((\rho \mathrm{E}^t + p^t)u^t_\alpha)}{\partial x_\alpha} - \frac{\partial(\tau^t_{\alpha\beta}u^t_\alpha + q^t_\alpha)}{\partial x_\alpha} = 0 \tag{A22}$$

The IBM source term has been ignored here for clarity. The convective flux of (A22) is computed by a 2D MUSCL-Hancock method. Thermal conduction and viscous heat are computed through second-order centered finite difference.