A High-Fidelity Simulation Framework for Turbulent Flows with Complex (Metamaterial) Structures

Diederik Beckers[*1], Srikumar Balasubramanian[†2], Ching-Te Lin[‡1], Andres Goza[§2], and H. Jane Bae[¶1]
[1] *California Institute of Technology, Pasadena, CA, 91125*
[2] *University of Illinois at Urbana-Champaign, Urbana, IL, 6180*

We present a high-fidelity fluid–structure interaction framework based on a continuous-forcing immersed boundary method, integrated into a parallelized three-dimensional turbulent channel flow solver. The method is designed to handle a wide range of (sub)surface geometries, including complex metamaterial interfaces. We extend an existing strongly coupled IB-FSI formulation with new spatially discrete operators that enable information transfer between subsurface metamaterials and compliant IB patches. Several key modifications support this versatile functionality: a hybrid uniform–stretched grid to reduce the computational load, IB forcing to compute one-sided velocity gradients for accurate friction velocity, and parallelized IB operations aligned with the underlying flow solver's domain decomposition. We demonstrate the method using five test cases: laminar cylindrical Couette flow; a parallel scaling test; and three turbulent channel flow configurations at a friction Reynolds number of 186, including a minimal flow unit with rigid walls, a channel with prescribed traveling wave–like wall deformations, and a channel with a compliant wall exhibiting rigid-body–like dynamics. The last test problem demonstrates the growing capability to handle arbitrary structures engaging in coupled dynamics with the flow, towards future investigations into fluid–metamaterial interaction.

I. Nomenclature

A	=	Runge-Kutta 3 coefficient matrix
a	=	Velocity amplitude of the traveling wave wall-like deformation
C	=	Damping matrix
C	=	Runge-Kutta 3 coefficient vector
c	=	Phase speed of the traveling wave-like wall deformation
c_i	=	Damping factor of the i-th block
D	=	Discrete divergence operator
$\boldsymbol{f}$	=	Force exerted by the immersed boundary on the fluid
f	=	Discrete immersed boundary surface stresses
G	=	Discrete gradient operator
h	=	Channel half-height
$\tilde{I}$	=	Immersed boundary points transfer operator
J	=	Mapping of a subsurface block displacement to immersed boundary points
k	=	Wavenumber of the traveling wave-like wall deformation
k_i	=	Stiffness of the i-th block
K	=	Stiffness matrix
L	=	Discrete Laplacian operator
L_x, L_y, L_z	=	Domain size in streamwise, wall-normal, and spanwise directions, respectively
m_i	=	Mass of the i-th block
M	=	Mass matrix

[*] Postdoctoral Scholar Research Associate, Graduate Aerospace Laboratories, AIAA Young Professional Member
[†] Graduate research assistant, Department of Aerospace Engineering
[‡] Graduate Student, Department of Mechanical and Civil Engineering, Student Member AIAA
[§] Assistant Professor, Department of Aerospace Engineering, AIAA Member
[¶] Assistant Professor, Graduate Aerospace Laboratories, Senior AIAA Member

N	=	Discrete convection operator
$\boldsymbol{n}$	=	Unit normal vector at the immersed boundary surface
n_b	=	Number of sampled body points in the immersed boundary patch
n_{block}	=	Number of mass blocks in the subsurface spring-mass chain
p	=	Pressure
p	=	Spatially discretized pressure
p_b	=	Pressure at the immersed boundary
Q	=	Operator to sum the immersed boundary forces
R	=	Discrete immersed boundary smearing operator
r	=	Right-hand side vector in the discretized momentum equation
R^T	=	Discrete immersed boundary interpolation operator
Re_τ	=	Friction Reynolds number
s	=	Parametric variable on the compliant immersed boundary patch
$\boldsymbol{S}_b$	=	Rate-of-strain tensor evaluated at the immersed boundary
t	=	Time
$\boldsymbol{t}_1, \boldsymbol{t}_2$	=	Tangential unit vectors at the immersed boundary surface
u	=	Spatially discretized fluid velocity
$\dot{\mathsf{u}}$	=	Spatially discretized fluid acceleration
$\boldsymbol{u}$	=	Fluid velocity
U_b	=	Bulk velocity
$\boldsymbol{u}_b$	=	Fluid velocity at the immersed boundary surface
u_τ	=	Friction velocity
W	=	Weighting matrix to filter immersed boundary surface stress
x, y, z	=	Cartesian coordinates in streamwise, wall-normal, and spanwise directions
$\boldsymbol{x}$	=	Eulerian position vector
x_c, y_c, z_c	=	Coordinates of the IB patch points in the x, y and z directions
$x_{\text{block}}, y_{\text{block}}, z_{\text{block}}$	=	Coordinates of the topmost mass block in the x, y and z directions
$\tilde{\chi}_i, \tilde{\zeta}_i, \ddot{\tilde{\chi}}_i$	=	Transverse displacement, velocity and acceleration of the i^{th} mass-block
$\chi, \zeta, \ddot{\chi}$	=	Spatially discrete deformation, velocity and acceleration of the immersed boundary patch
$\boldsymbol{\chi}, \boldsymbol{\zeta}, \ddot{\boldsymbol{\chi}}$	=	Continuous deformation, velocity and acceleration of the immersed boundary patch
χ_U, χ_L	=	Upper and lower wall displacements
Δs	=	Spacing between immersed boundary points
ΔS	=	Area of a single patch element
Δx	=	Grid spacing in the fluid domain
$\Delta x^+, \Delta y^+, \Delta z^+$	=	Grid spacing in streamwise, wall-normal, and spanwise directions in wall units
Γ_c	=	Compliant immersed boundary patch
ν	=	Kinematic viscosity
Ω	=	Flow domain
τ_w	=	Wall shear stress

II. Introduction

Advances in flow control have increasingly focused on the use of deformable surfaces to passively or actively manipulate boundary layer dynamics in aerodynamic and hydrodynamic systems. Among these approaches, compliant and metamaterial-based surfaces have shown particular promise due to their passive and adaptive characteristic to alter flow behavior through tailored structural responses. Applications range from delaying laminar-to-turbulent transition to reducing turbulent skin friction and mitigating separation. For instance, recent studies have demonstrated the use of phononic metamaterials to passively delay boundary layer transition by attenuating Tollmien–Schlichting wave growth through out-of-phase surface displacements [1, 2]. In the context of turbulent drag reduction, compliant surfaces have been optimized to suppress near-wall turbulence via anisotropic material responses or in-plane wall motions [3, 4]. Beyond transition and drag, Arif et al. [5] introduced the concept of distributed surface compliance for airfoil tonal noise reduction, demonstrating that resonating panels embedded within the surface can passively mitigate flow instabilities across varying angles of attack. These examples collectively highlight the versatility of compliant and metamaterial surfaces as a platform for passive flow control.

As these control actuation increasingly rely on complex geometric and structural designs, there arises a critical need for high-fidelity simulation tools capable of resolving both fluid dynamics and the detailed geometry of the control surfaces. In this work, we employ the immersed boundary method due to its robustness and versatility in handling arbitrarily complex geometries and fluid–structure interfaces. This approach enables accurate representation of the interaction between the turbulent flow and the spatially heterogeneous, deformable surfaces characteristic of metamaterials, without the need for conformal meshing or remeshing during deformation.

To effectively model such flow-metamaterial interaction (FMI) problems, the chosen numerical framework must not only accommodate geometric complexity, but also accurately capture the bidirectional coupling between unsteady structural motion and turbulent flow dynamics. In other words, FMI problems require a robust, accurate numerical simulation framework to couple the time-varying structural motion to the complex high-dimensional flow field. Numerical formulations for general fluid-structure interaction (FSI) problems are identified by the way the numerical coupling between the flow and the structure is handled, forming a broad class of methods that are either *strongly-coupled* or *weakly-coupled* depending on how the nonlinear no-slip constraint is handled. A connected consideration is the operations count/ time-complexity associated with performing these operations. Physical considerations usually require the method to handle large structural motions, and be stable for small structure-to-fluid mass ratios. Weakly-coupled FSI methods [6–8] do not strictly enforce the nonlinear no-slip constraint and are unstable for low mass ratios and large structural displacements.

In this study, we focus on strongly-coupled methods for their stability and accuracy benefits. Some strongly-coupled methods [9, 10] evaluate the nonlinear constraint at a previous time step. They are therefore able to frame the equations to be linear, circumventing costly iterations. However, the above sub-class of methods introduce a time-lag that reduces the temporal accuracy of the method. Other strongly-coupled methods that maintain the nonlinear nature of the constraint are forced to have iterations to solve for the variables. Methods that use the block Gauss-Seidel technique for the iterations typically use heuristic relaxation parameters that incur several iterations to converge [11]. A Newton-Raphson technique could be used for the convergence of the iterations, but involves expensive Jacobian matrix-vector products per timestep [12–14].

To address the limitations of both heuristic iterative strategies and computationally expensive Newton-based solvers, we adopt an alternative strongly-coupled approach that is built upon the fluid-structure interaction (FSI) formulation of Goza and Colonius [15]. This approach extends the immersed boundary (IB) projection method [16] to coupled FSI problems, using a Newton method for the nonlinear algebraic equations that arise. A block-LU factorization is used to restrict all FSI iterations to sub-systems that scale with the number of points on the immersed surface. The method avoids the use of heuristic relaxation parameters and avoids the need to solve large-dimensional, global linear systems typically required by Newton-Raphson iterations. The algorithm is capable of handling large structural displacements and small structure-to-fluid mass ratios [15].

The current work improves and adapts this IB-FSI formulation to operate within a three-dimensional turbulent channel flow solver [17, 18]. The resulting simulation framework enables robust, accurate computations of flows over arbitrarily shaped and dynamically deforming boundaries. This capability is particularly important for studying metamaterials, whose architectures may be complex or time-varying. To integrate the IB-FSI method into our existing parallelized 3D flow solver, we introduce several key modifications. First, we adapt the discrete delta functions for use in periodic domains. Second, we implement a hybrid uniform-stretched grid to optimize computational load distribution while keeping the IB surface within a uniform grid region—necessary for accurate delta function application. Third, we incorporate "immersed layers" concepts from Eldredge [19] to estimate one-sided velocity gradients using the IB forcing. Fourth, we parallelize the IB operators in a manner consistent with the channel flow solver's domain decomposition strategy, ensuring efficient and scalable FSI simulations in turbulent regimes. Finally, we construct new spatially discrete operators to enable information transfer from (sub)surface dynamics to surface patches treated within the immersed boundary formalism. This algorithmic design allows the solver to remain agnostic to the specific structural model or geometry, facilitating easy exchange of a wide range of (sub)surface structures. We give an overview of some of the features of each of these five developments, along with results from three test cases to demonstrate the functionality of the method.

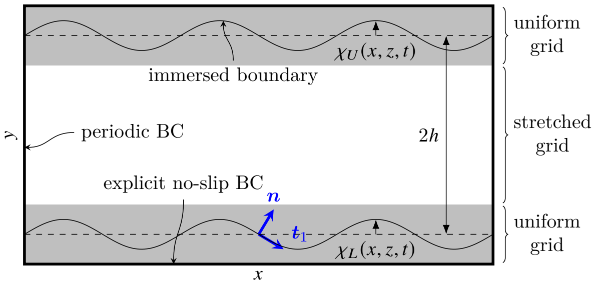


Fig. 1 Diagram of the mesh and boundary conditions setup to study fluid-metamaterial interaction with immersed boundaries. The gray areas are zones with a uniform grid. The dashed lines are the undeformed locations of the IB channel walls.

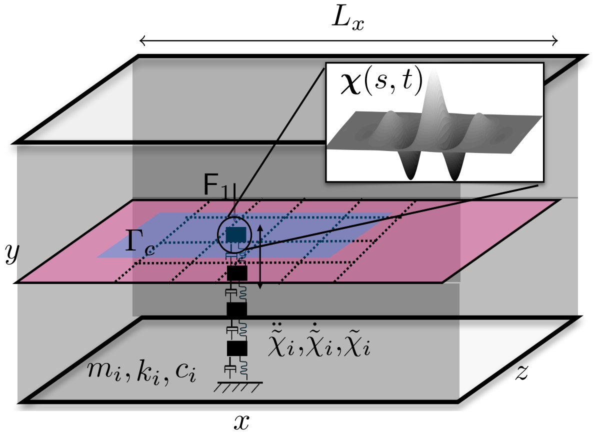

L_x
$\chi(s,t)$
F_1
Γ_c
y
$\ddot{\tilde{\chi}}_i, \dot{\tilde{\chi}}_i, \tilde{\chi}_i$
m_i, k_i, c_i
z
x

Fig. 2 Diagram of the preliminary setup to study fluid-metamaterial interaction. A spring-mass chain drives a flexible IB patch that is fully-coupled with the flow. The transverse motion of the spring-mass blocks are designed to create a gaussian deformation profile in the IB patch.

III. Methodology

A. Incorporating the immersed boundary projection method into a turbulent channel flow solver

Our FSI framework is built on the immersed boundary projection method [15, 16], which introduces a body force $\boldsymbol{f}$ into the continuous Navier-Stokes equations to enforce the no-slip condition on immersed surfaces,

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u} + \int_{\Gamma_c(s,t)} \boldsymbol{f}(\boldsymbol{\chi}(s,t)) \delta(\boldsymbol{\chi}(s,t) - \boldsymbol{x}) \mathrm{d}s \,, \tag{1}$$

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

$$\int_{\Omega} \boldsymbol{u}((\boldsymbol{x}, t)) \delta(\boldsymbol{\chi}(s,t) - \boldsymbol{x}) \mathrm{d}x = \boldsymbol{\zeta}(s,t) \,, \tag{3}$$

where $\boldsymbol{u} = (u, v, w)^T$ and p are the continuous velocity and pressure variables, and $\boldsymbol{\chi}(s,t)$ and $\boldsymbol{\zeta}(s,t)$ are the continuous time-varying displacement and velocity on the domain Γ_c comprising the surface(s) interacting with the flow, parameterized by s. When discretized, the immersed surface(s) are represented by a set of Lagrangian markers with a spacing Δs embedded within a fixed Eulerian fluid grid. Each marker represents a discrete surface element with area ΔS. The singular Dirac delta functions are approximated by smoothed, discrete versions to enable numerical interpolation and spreading. The no-slip condition on the surface (3)—involving the flow velocity interpolated to the Lagrangian points—is used to compute the IB source term $\boldsymbol{f}$, which physically corresponds to a stress along the (collection of) surface(s) Γ_c, that is spread to the surrounding fluid grid via the smoothed delta function in the momentum equation (1) to impose the no-slip boundary condition. Its role is analogous to that of pressure in enforcing continuity (2).

For completeness, the solution algorithm for the IB method, together with the FSI formulation that builds upon it, is described in detail in the following section. The derivation leverages the formulation in Goza and Colonius [15] but is reproduced in full detail here to highlight changes introduced to the algorithm to accommodate its use in the wall bounded turbulent flow setting of focus here. The IB-FSI framework is integrated into an in-house three-dimensional turbulent channel flow solver that employs second-order finite differences on a staggered grid and a fractional-step projection method [20], with time integration performed using a third-order Runge–Kutta scheme [21]. The channel flow solver has been extensively validated in previous studies of turbulent channel flows [17, 18, 22].

We incorporate the IB formalism to model moving/deforming channel boundaries, represented as immersed surfaces within a larger three-dimensional computational domain with periodic boundary conditions in the streamwise (x) and spanwise (z) directions and no-slip conditions at each wall, as illustrated in Fig. 1. A globally applied pressure gradient drives the flow. In Fig. 1, the immersion of the channel boundaries in a larger domain results in two regions—one above and one below the nominal channel of interest—that are also affected by the globally applied pressure gradient. However, their only purpose is to facilitate the use discrete delta functions (DDFs), which extends over several grid cells on both sides of the surface. The flow in these auxiliary regions is not of interest and the problem configuration is set up so it has a negligible influence on the central flow. To accommodate this aim, we mask the pressure gradient using an indicator function that equals one inside the central channel and zero elsewhere. This indicator function is computed as the solution to a Poisson equation, with the source term given by the divergence of the normals at the IB points, distributed over the grid using the DDFs [19, 23]. For simulations that enforce a constant mass flow rate, the global pressure gradient is adjusted so that the flow rate within the central channel—isolated again using the indicator function—remains constant. Fig. 2 illustrates an alternative channel configuration that we use to study fluid–metamaterial interaction. In this setup, the immersed surface is centered in the wall-normal direction and divides the domain into two channels.

To achieve a high grid resolution near the walls in turbulent flow simulations, wall-normal grid stretching is applied using a hyperbolic tangent distribution. However, most DDFs used in IB methods assume uniform grids. Furthermore, for moving walls, it is desirable to maintain consistently fine resolution wherever the wall might traverse, without needing to remesh. For these reasons, we estimate the extent of the wall motion a priori and place a zone with uniform grid spacing (hereafter referred to as a *uniform grid zone*) around the immersed boundary surfaces on each end of the stretched grid. Both uniform grid zones have a wall-normal size that is at least as big as the estimated extent of the wall motion, as illustrated in Fig. 1. This treatment ensures that the support of the DDFs at IB points that represent the deforming wall remains inside a uniform grid region at all times.

To estimate the friction velocity $u_\tau = \sqrt{\tau_w}$ at the IB points, we recognize that the IB forcing contains the jump of the pressure and viscous stresses over the interface [19]:

$$\boldsymbol{f} = -(p_b^{\text{ext}} - p_b^{\text{int}})\boldsymbol{n} + \frac{2}{\text{Re}}\left(\boldsymbol{S}_b^{\text{ext}} - \boldsymbol{S}_b^{\text{int}}\right)\cdot\boldsymbol{n}\,, \tag{4}$$

where the superscripts "ext" ("int") denote the side of the interface that the positive (negative) normal is pointing to, the subscript "b" indicates the values at the IB points, and where $\boldsymbol{S}_b = \frac{1}{2}\left(\nabla\boldsymbol{u}_b + \nabla^\top\boldsymbol{u}_b\right)$ with $\nabla\boldsymbol{u}_b$ the velocity gradient at the interface. Furthermore, using the immersed boundary techniques from Eldredge [19], we can show that the interpolation operation applied to the gradient of the velocity field gives the average of the velocity gradients at each side of the interface, or,

$$\int_\Omega \nabla\boldsymbol{u}(x)\delta(\boldsymbol{x}-\boldsymbol{\xi})\mathrm{d}\boldsymbol{x} = \frac{1}{2}\left(\nabla\boldsymbol{u}_b^{\text{ext}} + \nabla\boldsymbol{u}_b^{\text{int}}\right)\,. \tag{5}$$

Because we impose tangential velocity gradients on both sides of the interface, we can combine Eqs. (4) and (5) to retrieve the wall-normal derivative of the tangential velocity on one side of the interface,

$$\nabla_{\boldsymbol{n}} u_{b,\boldsymbol{t}_1}^{\text{ext}} = \left[\left(\int_\Omega \nabla\boldsymbol{u}\delta(\boldsymbol{x}-\boldsymbol{\xi})\mathrm{d}\boldsymbol{x}\right)\cdot\boldsymbol{n} + \frac{\text{Re}}{2}\boldsymbol{f}\right]\cdot\boldsymbol{t}_1\,. \tag{6}$$

which we can then use to compute the wall shear stress on one side of the interface $\tau_w^{\text{ext}} = \frac{1}{\text{Re}}\nabla_{\boldsymbol{n}} u_{b,\boldsymbol{t}_1}^{\text{ext}}$ and subsequently the friction velocity for that side.

B. Discretization and a strongly-coupled fluid-structure interaction formulation

We now describe the IB-endowed spatially discrete governing equations and the algorithm used to solve them. As previously mentioned, second-order finite differences are used for the discretization of the governing equations for the turbulent wall-bounded flow, Eqs. (1)–(3). The discrete divergence, gradient, viscous-Laplacian, and convection operators are indicated as $\mathsf{D}, \mathsf{G}, \mathsf{L}$ and N, respectively. The discrete versions of the IB smearing and interpolation operators are indicated by R and R^T respectively, where the transpose relation between the operators hold true when appropriate scaling factors are absorbed into the surface stress and velocity variables (see [16] for more details on the construction of these operators). The spatially discrete form of the coupled governing equations is given by

$$\dot{\mathsf{u}} = \mathsf{N}(\mathsf{u},\mathsf{u}) - \mathsf{G}\mathsf{p} + \frac{1}{\text{Re}}\mathsf{L}\mathsf{u} + \mathsf{R}(\chi)\,\mathsf{f}\,, \tag{7}$$

$$\mathsf{D}\mathsf{u} = 0\,, \tag{8}$$

$$\mathsf{M}\ddot{\tilde{\chi}} + \mathsf{K}\tilde{\chi} + \mathsf{C}\dot{\tilde{\chi}} = \mathsf{F} = \left[\mathsf{Q}\tilde{\mathsf{I}}^T\mathsf{W}(\chi)\mathsf{f}\right], \tag{9}$$

$$\dot{\tilde{\chi}} = \tilde{\zeta}, \tag{10}$$

$$\mathsf{R}^T(\chi)\mathsf{u} - \tilde{\mathsf{I}}\mathsf{J}\tilde{\zeta} = 0. \tag{11}$$

Equations (7, 8, 11) are the spatially discrete variants of Eqs. (1)–(3), and Eqs. (9)–(10) are the governing equations for the structure, written generically in terms of typical mass, stiffness and damping matrices which arise from typical finite element discretizations. For consistency with the flow equations, the structural equations (9)–(10) are written in first order form (though this is non-essential). The flow and structure equations are coupled by the discrete force term f, which simultaneously imposes no-slip on the fluid via Eq. (7), drives structural motion via Eq. (9), and is a constraint variable that arises to solve Eq. (11).

The discrete variables χ, ζ correspond to the three-dimensional displacement and velocity of the immersed surface, Γ_c. These quantities are generally distinct from the displacements and velocities of the structure $\tilde{\chi}, \tilde{\zeta}$. The operator W is a weighting matrix that acts as a filter to obtain smooth surface stresses (see [15] for its definition). The operators $\mathsf{Q}\tilde{\mathsf{I}}^T$ and $\tilde{\mathsf{I}}\mathsf{J}$ arise to relate these distinct structure versus surface variables. For illustration purposes, and to facilitate a description of the FSI test case explored below, we detail their construction below for the specific problem configuration of interest, detailed in Fig. 2. We emphasize that the computational methodology is agnostic to the specific structural configuration and surface/subsurface architecture, and can be adapted to a variety of flow-structure configurations by appropriately altering these operators.

The problem configuration in Fig. 2 involves a tiling of the bottom wall into constituent rectangular elements along the x and z directions. At coordinates $x_{\text{block}}, y_{\text{block}}, z_{\text{block}}$, a subsurface spring mass chain is attached to the IB surface. The subsurface chain consists of individual blocks of mass m_i connected to each other through springs of stiffness k_i and dampers with damping factor c_i. The motion of the blocks is constrained to be along the transverse direction, with $\tilde{\chi}_i, \dot{\tilde{\chi}}_i, \ddot{\tilde{\chi}}_i$ representing the transverse displacement, velocity, and acceleration of each block. While the structure for the current study uses a uniform value for the mass and stiffness across the chain (m_i, k_i), we emphasize that the model allows for easy integration of defects, di-atomicity into the setup by tuning the local mass and stiffness values of the individual blocks. The entire sub-surface structure is not visible to the flow, and its modeling is designed to transfer its dynamic motion to the compliant IB surface. The transverse displacement of the topmost mass $(\tilde{\chi}_1)$ is transferred to the IB surface Γ_c through a Gaussian function whose continuous expression is given below (the discrete form is given by the operator J in Eq. (11)):

$$\chi(s,t) = \tilde{\chi}_1 \cos\left(\frac{3\pi\,(x_c - x_{\text{block}})}{L_x}\right) e^{-\,(x_c - x_{\text{block}})^2/(2\sigma^2)}. \tag{12}$$

In the above expression, $\tilde{\chi}_1$ represents the displacement of the first mass in the sub-surface chain and $\sigma = 1$. $\tilde{\mathsf{I}}$ and $\tilde{\mathsf{I}}^T$ are transfer operators that are sparse and consist of unit values to help extract components in the y direction from the full vector f consisting of components in x, y, z directions. The exact form depends on the implementation, but is given by the expression below if f is implemented as a pointwise-stacked array of vectors, where $\tilde{\mathsf{I}}^T \in \mathbb{R}^{n_b \times 3n_b}$.

$$\left(\tilde{\mathsf{I}}^T\right)_{i,j} = \begin{cases} 1, & j = 3(i-1)+2, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1,\ldots,n_b, \tag{13}$$

The sparse operator Q is suitably constructed such that the topmost subsurface block receives the cumulative surface force acting on the IB surface. The mathematical definition for $\mathsf{Q} \in \mathbb{R}^{n_{\text{nblock}} \times n_b}$ is

$$\left(\mathsf{Q}\right)_{i,j} = \begin{cases} \Delta S, & i = 1,\ j = 1,\ldots,n_b, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1,\ldots,n_{\text{block}}. \tag{14}$$

Equations (7)–(11) are time-discretized using the three-step Runge-Kutta scheme. Within each RK stage, a second-order accurate, implicit, average-acceleration Newmark time stepping scheme [24] (same variant as the one used within [15]) is used to time-march Eqs. (9, 10). If u^n represents the solution variable at t^n, we seek u^{n+1} that represents the variable at the next time step. We define

$$\mathsf{F}(\mathsf{u}^n, \mathsf{t}^n) = \mathsf{N}(\mathsf{u}^n, \mathsf{u}^n) + \frac{1}{\text{Re}}\mathsf{L}\mathsf{u}^n, \tag{15}$$

and let $\mathsf{u}^{n+1(j)}$ represent the solution at the end of the j^{th} RK stage. The fully-discrete equations for the j^{th} RK stage is written as:

$$\mathsf{u}^{n+1(j)} = \mathsf{u}^{n} + \Delta t {\mathsf{r}_{\mathsf{n}}}^{j} - \Delta t \mathsf{G} \mathsf{p}^{\,n+1(j)} + \Delta t \mathsf{R}(\chi^{\,n+1(j)})\, \mathsf{f}^{\,n+1(j)}, \tag{16}$$

$$\mathsf{D}\mathsf{u}^{n+1(j)} = 0\,, \tag{17}$$

$$\frac{4}{\Delta t^2}\mathsf{M}\tilde{\chi}^{n+1(j)} + \mathsf{K}\tilde{\chi}^{n+1(j)} - \left[\mathsf{Q}\tilde{\mathsf{I}}^{T}\mathsf{W}^{n+1(j)}\mathsf{f}^{n+1(j))}\right] = \mathsf{M}\left(\frac{4}{\Delta t^2}\tilde{\chi}^{n} + \frac{4}{\Delta t}\tilde{\zeta}^{n} + \ddot{\tilde{\chi}}^{n}\right)\,, \tag{18}$$

$$\frac{2}{\Delta t}\tilde{\chi}^{n+1(j)} - \tilde{\zeta}^{n+1(j)} = \tilde{\zeta}^{n} + \frac{2}{\Delta t}\tilde{\chi}^{n}\,, \tag{19}$$

$$\mathsf{R}^{T}(\chi^{n+1(j)})\mathsf{u}^{n+1(j)} - \tilde{\mathsf{I}}\mathsf{J}\tilde{\zeta}^{n+1(j)} = 0\,. \tag{20}$$

where $j = 1, 2, 3$ represents the index associated with the RK3 stage. ${\mathsf{r}_{\mathsf{n}}}^{j}$ can be computed using the coefficients derived from the butcher tableau and is given by:

$${\mathsf{r}_{\mathsf{n}}}^{j} = \sum_{m=1}^{j} A_{jm}\, \mathsf{F}\left(\mathsf{u}^{n+1(m)}, \mathsf{t}_{n} + C_{m}\Delta t\right), \qquad j = 1, 2, 3. \tag{21}$$

$$C = \begin{pmatrix} 0 \\ \frac{8}{15} \\ \frac{2}{3} \end{pmatrix}, \qquad A = \begin{pmatrix} \frac{8}{15} & 0 & 0 \\ \frac{1}{4} & \frac{5}{12} & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} \end{pmatrix}. \tag{22}$$

Equations (16)–(20) represent a system of nonlinear algebraic equations. To solve for the variables, we use an iterative approach to linearize the solution at iteration $k+1$ about the variables evaluated at iteration k.

$$\tilde{\chi}^{n+1(j)}_{(k+1)} = \tilde{\chi}^{n+1(j)}_{(k)} + \Delta\tilde{\chi}\,, \tag{23}$$

$$\tilde{\zeta}^{n+1(j)}_{(k+1)} = \tilde{\zeta}^{n+1(j)}_{(k)} + \Delta\tilde{\zeta}\,. \tag{24}$$

In the above linearization, the variables $\Delta\tilde{\chi}$, $\tilde{\zeta}$ are assumed to be small in magnitude. Substituting the above linearization into Eq. (16)–(20), performing a Taylor series expansion, and dropping the third-rank tensor terms (see [15] for more details), we arrive at the equations that are linearized about the k-th FSI iteration (indicated by subscripts in relevant operators).

$$\begin{pmatrix} \mathsf{I} & \mathsf{G} & 0 & 0 & \mathsf{R}^{n+1(j)T}_{(k)} \\ \mathsf{D} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \left(\mathsf{K} + \frac{4}{\Delta t^2}\mathsf{M}\right) & -\mathsf{Q}\tilde{\mathsf{I}}^{\mathsf{T}}\mathsf{W}^{n+1(j)}_{(k)} \\ 0 & 0 & -\mathsf{I} & \frac{2}{\Delta t} & 0 \\ \mathsf{R}^{n+1(j)}_{(k)} & 0 & -\tilde{\mathsf{I}}\mathsf{J} & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathsf{u}^{n+1(j)} \\ \mathsf{p}^{n+1(j)} \\ \Delta\tilde{\zeta} \\ \Delta\tilde{\chi} \\ \mathsf{f}^{n+1(j)}_{(k+1)} \end{pmatrix} = \begin{pmatrix} {\mathsf{r}_{\mathsf{n}}}^{f} \\ {\mathsf{r}_{\mathsf{n}}}^{p} \\ {\mathsf{r}_{\zeta}}^{k} \\ {\mathsf{r}_{\chi}}^{k} \\ {\mathsf{r}_{\mathsf{c}}}^{k} \end{pmatrix}, \tag{25}$$

where

$${\mathsf{r}_{\mathsf{n}}}^{f} = \mathsf{u}^{n} + \Delta t {\mathsf{r}_{\mathsf{n}}}^{j}\,,$$

$${\mathsf{r}_{\mathsf{n}}}^{p} = 0\,,$$

$${\mathsf{r}_{\zeta}}^{k} = \mathsf{M}\left(\frac{4}{\Delta t^2}\tilde{\chi}^{n} + \frac{4}{\Delta t}\tilde{\zeta}^{n} + \dot{\tilde{\zeta}}^{n}\right) - \frac{4}{\Delta t^2}\mathsf{M}\tilde{\chi}^{n+1(j)}_{(k)} - \mathsf{K}\tilde{\chi}^{n+1(j)}_{(k)}\,, \tag{26}$$

$${\mathsf{r}_{\chi}}^{k} = \tilde{\zeta}^{n} + \frac{2}{\Delta t}\tilde{\chi}^{n} - \frac{2}{\Delta t}\tilde{\chi}^{n+1(j)}_{(k)} + \tilde{\zeta}^{n+1(j)}_{(k)}\,, \tag{27}$$

$${\mathsf{r}_{\mathsf{c}}}^{k} = \tilde{\mathsf{I}}\mathsf{J}\tilde{\zeta}^{n+1(j)}_{(k)}\,, \tag{28}$$

$$\hat{\mathsf{K}} = \mathsf{K} + \frac{4}{\Delta t^2}\mathsf{M}\,. \tag{29}$$

The block system in Eq. (25) can be LU factorized to give:

$$\mathsf{u}^{n+1(j)*} = \mathsf{u}^{n} + \Delta t \mathsf{r_n}^{j} \tag{30}$$

$$[\mathsf{B}]\mathsf{f}^{n+1(j)}_{(k+1)} = \left(\mathsf{R}^{n+1(j)T}_{(k)}\mathsf{G}(\mathsf{DG})^{-1}\mathsf{D}\mathsf{r_n}^{f}\right) - \left(\mathsf{R}^{n+1(j)T}_{(k)}\mathsf{G}(\mathsf{DG})^{-1} r^{p}_{n}\right) - \left(\mathsf{R}^{n+1(j)T}_{(k)}\mathsf{r_n}^{f}\right) - \left(\tilde{\mathsf{I}}\mathsf{J}\mathsf{r}_{\chi}{}^{k}\right) + \left(\frac{2}{\Delta t}\tilde{\mathsf{I}}\mathsf{J}\hat{\mathsf{K}}^{-1}\mathsf{r}_{\zeta}{}^{k} + \mathsf{r_c}{}^{k}\right). \tag{31}$$

$$\Delta\tilde{\chi} = \hat{\mathsf{K}}^{-1}\left[\mathsf{r}_{\zeta}{}^{k} + \mathsf{Q}\tilde{\mathsf{I}}^{T}\mathsf{W}\mathsf{f}^{n+1(j)}_{(k+1)}\right] \tag{32}$$

$$\Delta\tilde{\zeta} = \frac{2}{\Delta t}\Delta\tilde{\chi} - \mathsf{r}_{\chi}{}^{k} \tag{33}$$

$$-\left[\mathsf{DG}\right]\mathsf{p}^{n+1(j)} = -\mathsf{D}\mathsf{r_n}^{f} + \mathsf{r_n}^{p} + \mathsf{D}\mathsf{R}^{n+1(j)}_{(k)}\mathsf{f}^{n+1(j)}_{(k+1)} \tag{34}$$

$$\mathsf{u}^{n+1(j)} = \mathsf{u}^{n+1(j)*} - \mathsf{G}\mathsf{p}^{n+1(j)} - \mathsf{R}^{n+1(j)}_{(k)}\mathsf{f}^{n+1(j)}_{(k+1)} \tag{35}$$

$$\mathsf{B} := -\left(\mathsf{R}^{n+1(j)T}_{(k)}\mathsf{R}^{n+1(j)}_{(k)}\right) + \left(\mathsf{R}^{n+1(j)T}_{(k)}\mathsf{G}(\mathsf{DG})^{-1}\mathsf{D}\mathsf{R}^{n+1(j)}_{(k)}\right) - \left(\frac{2}{\Delta t}\tilde{\mathsf{I}}\mathsf{J}\hat{\mathsf{K}}^{-1}\mathsf{Q}\tilde{\mathsf{I}}^{T}\mathsf{W}\right). \tag{36}$$

In the procedure containing Eqs. (30)–(35), the FSI iterations are formally restricted to Eqs. (31)–(33). This outcome arises directly from the LU-factorization, and results in iterations that are restricted to linear(ized) systems that scale as the number of surface points on the IB surface. That is, the iterations avoid solving systems that scale with the number of points in the flow and structure domains. Within each FSI iteration, we highlight that Eq. 31 is the most computationally intensive step due to a linear solve for the scaled surface stress f. We use a stabilized bi-conjugate gradient (BiCGStab(l)) algorithm to iteratively obtain the surface stress. The solution at the next time instance u^{n+1} is obtained by advancing the entire system through stages $j = 2, 3$ through a similar procedure as outlined above.

C. Parallel framework

The solver is parallelized using the Message Passing Interface (MPI) standard. The computational domain is partitioned in the z-direction into slices that span the full x- and y-extents, as illustrated in Fig. 3. During the simulation, each subdomain exchanges boundary data with its neighbors on the z-normal faces to maintain continuity across slice interfaces, in addition to applying global boundary conditions.

Operations involving immersed boundary (IB) points are parallelized by assigning each point to the MPI rank corresponding to the partition in which it resides. A key challenge arises when interpolating flow values or regularizing IB forces for points located near partition boundaries. For example, in Fig. 3, the IB point (red diamond) lies close to the interface between ranks 1 and 2, and the stencil of fluid grid points influenced by this IB point spans both partitions. To perform these IB operations entirely on a local processor, the relevant flow data must be transferred between neighboring ranks before and after the interpolation and regularization steps.k This issue is addressed by extending inter-partition communication to include additional flow grid values used exclusively for IB operations. These exchange regions, termed "support cells" in Fig. 3, provide each rank with the necessary flow data to evaluate the IB operators locally. The spanwise width of the support-cell region, $2N_\delta + 1$, is determined by the discrete delta function used for regularization; in the present implementation, $N_\delta = 2$.

IV. Results

To assess the accuracy and applicability of our simulation tool, we perform a sequence of benchmark tests and validation studies. We first consider two tests aimed at verifying the numerical accuracy of the underlying IB implementation: a spatial convergence analysis of laminar cylindrical Couette flow, and a comparison of turbulent flow statistics for a channel flow in a minimal flow unit at $Re_\tau \approx 186$ against reference data from the literature. We then demonstrate parallel efficiency of the IB operators. Next, we show a turbulent channel flow simulation with prescribed, traveling wave-like wall deformations, replicating a numerical experiment from the literature to evaluate physical fidelity.

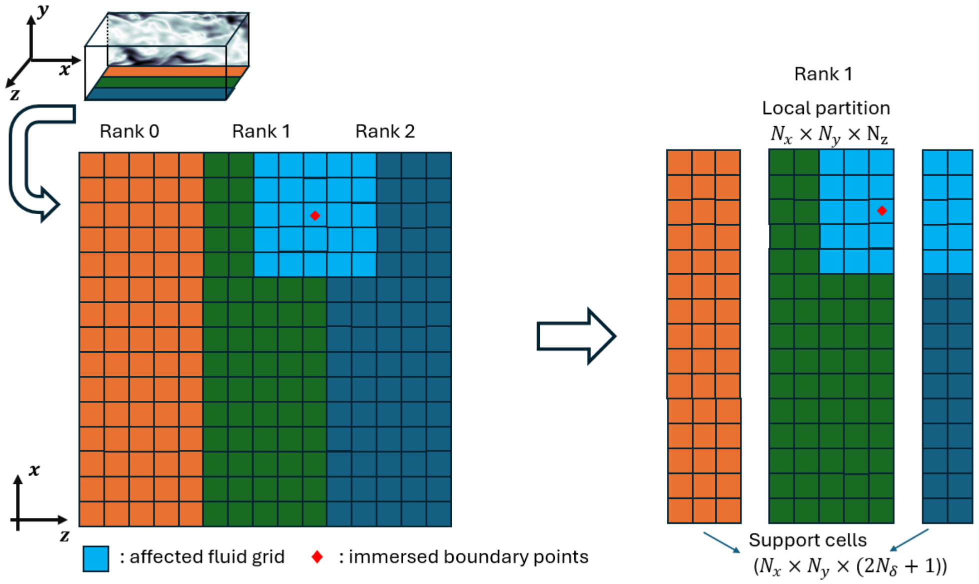

y
x
z
Rank 0
Rank 1
Rank 2
x
z
🟦 : affected fluid grid
♦ : immersed boundary points
Rank 1
Local partition
$N_x \times N_y \times N_z$
Support cells
$(N_x \times N_y \times (2N_\delta + 1))$

Fig. 3 Schematics of parallelization. Domain decomposition for parallel computation. The fluid grid is partitioned into ranks (Rank 0-2). The fluid grids affected by the immersed boundary points (red) are marked in light blue.

In all the preceding test cases, we do not model FSI, and the steps (30)–(35) reduce to the immersed boundary projection method (IBPM) of Taira and Colonius [16] adapted here for RK3 time integration. In the first two examples, the IB points remain stationary, and we can precompute and store the inverse of the operator acting on the surface stress in (31) to efficiently simulate large number of time steps while enforcing the no-slip constraint on the IB to machine precision. In the third and fourth cases, we solve for the forcing iteratively using BiCGStab, similar to the FSI algorithm described in Sec. III.B.

In the last example, we demonstrate the method's capability in a coupled FSI scenario using a simplified test case with high mass and stiffness structural parameters, so that the dynamics are effectively rigid even though the simulation is performed on the fully coupled FSI algorithm (30)–(35). These tests provide a foundation for future investigations involving more complex FMI behavior.

A. Laminar flow validation problem

To demonstrate the expected spatial accuracy of the proposed immersed boundary (IB) method, we simulate the steady-state cylindrical Couette flow between two concentric cylinders with radii $r_1 = 0.5$ and $r_2 = 1$. The inner cylinder rotates with a constant angular velocity Ω, while the outer cylinder remains stationary. The Reynolds number based on the velocity and diameter of the inner cylinder is $Re = 2r_1^2\Omega/\nu = 0.225$. The cylinder axes are aligned with the z-direction. Although the simulation is conducted in a three-dimensional domain, the flow remains laminar and varies only in the radial and azimuthal directions within the x-y plane.

This configuration admits an analytical solution for the steady-state azimuthal velocity profile, which we use to evaluate the accuracy of our numerical results. We perform simulations on successively refined grids and compute the L_2 and L_∞ norms of the velocity error relative to the exact solution. Body points are spaced with Δs such that $\Delta s/\Delta x = 2$, ensuring adequate resolution of the immersed interface. As shown in Fig. 4, the error norms exhibit first-order convergence, consistent with expectations for this class of IB methods.

B. Turbulent channel flow in a minimal flow unit at $\mathrm{Re}_\tau \approx 186$

In this example, we perform a series of physically resolved turbulent channel-flow simulations at a Reynolds number $\mathrm{Re}_\tau = u_\tau h/\nu = 186$, where u_τ is the friction velocity and h is the channel half-height. We use the immersed boundary configuration of Fig. 2 with domain size $(L_x, L_y, L_z) = (1.718, 4, 0.859)$ and a static, undeformed planar IB wall located at $y = 2$, where x, y, and z denote the streamwise, wall-normal, and spanwise coordinates, respectively. This IB

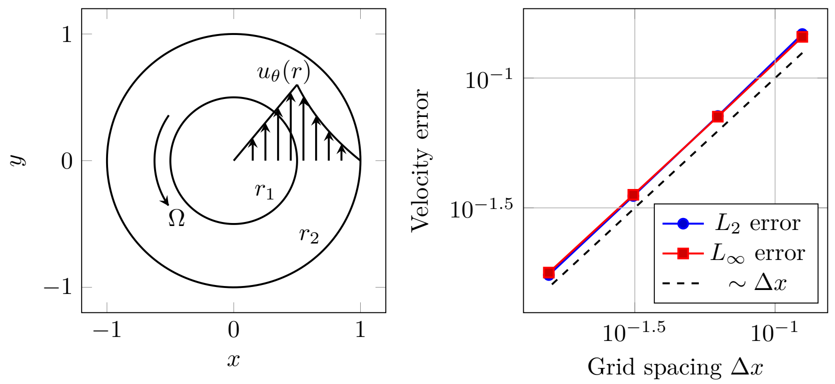


Fig. 4 Left: Schematic of the azimuthal velocity profile $u_\theta(r)$ in a cylindrical Couette flow between two concentric cylinders of radii r_1 and r_2, with the inner cylinder rotating at angular velocity Ω. Right: Log-log plot of the error convergence for approximating the velocity field with the current method.

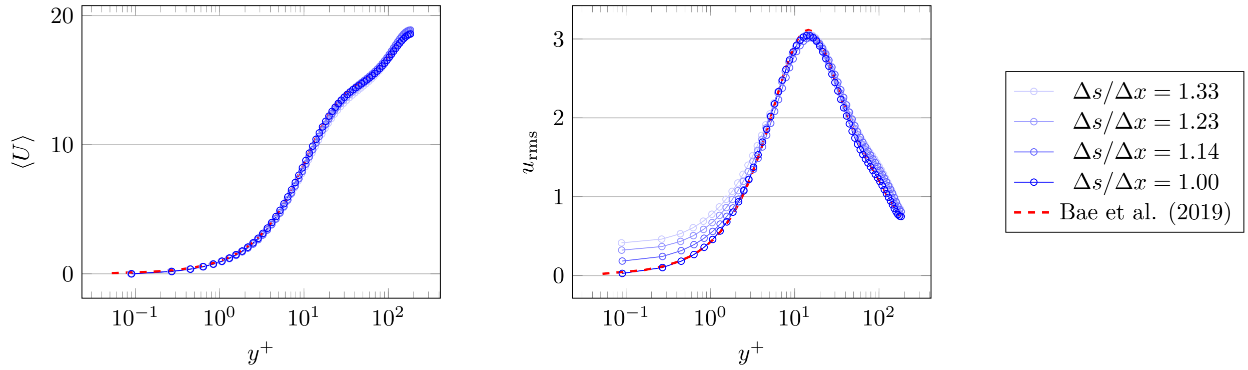


Fig. 5 Effect of IB point spacing on turbulent channel-flow statistics at $\text{Re}_\tau = 186$. The left panel shows the mean streamwise velocity, and the right panel shows the streamwise velocity fluctuations. Results from simulations with varying body-to-grid spacing ratios, $\Delta s/\Delta x \in \{1.33,\ 1.23,\ 1.14,\ 1.00\}$, are shown using solid blue lines with circle markers, ranging from light to dark blue as $\Delta s/\Delta x$ decreases. For reference, the profile of Bae et al. (2019) is overlaid with red dashed lines.

wall partitions the domain into a bottom ($0 < y < 2$) and top ($2 < y < 4$) channel, each corresponding to a minimal flow unit. The entire domain is discretized using $32 \times 257 \times 16$ computational cells. Two uniformly sized grid cells are placed on either side of the IB, while we apply wall-normal grid stretching everywhere else (in both the top and bottom channels) using a hyperbolic tangent function with a stretching factor of 2.6. This results in $\Delta x^+ \approx 10$, $\Delta z^+ \approx 10$ and a wall-normal minimum grid spacing of $\Delta y^+_{\text{min}} \approx 0.18$, where the superscript + indicates inner-scale normalization using u_τ and ν. This setup enables a realistic assessment of near-wall turbulence and allow us to examine the effect of the body-to-grid spacing ratio ($\Delta s/\Delta x$) on the accuracy of the IB representation. Four values of this ratio are considered: $\Delta s/\Delta x \in \{1.33,\ 1.23,\ 1.14,\ 1.00\}$. Fig. 5 shows the resulting mean streamwise velocity profile $\langle U \rangle$ and streamwise velocity fluctuations u_{rms} in the top channel versus the wall-normal coordinate measured relative to the IB wall. The mean velocity is largely insensitive to the IB-point spacing and agrees well with the results of Bae et al. [18]. In contrast, the streamwise fluctuations exhibit noticeable sensitivity in the near-wall region. In particular, physically consistent decay to zero at the boundary is observed only when $\Delta s/\Delta x = 1$, indicating that sufficiently fine IB-point spacing is required to accurately capture near-wall turbulent fluctuations.

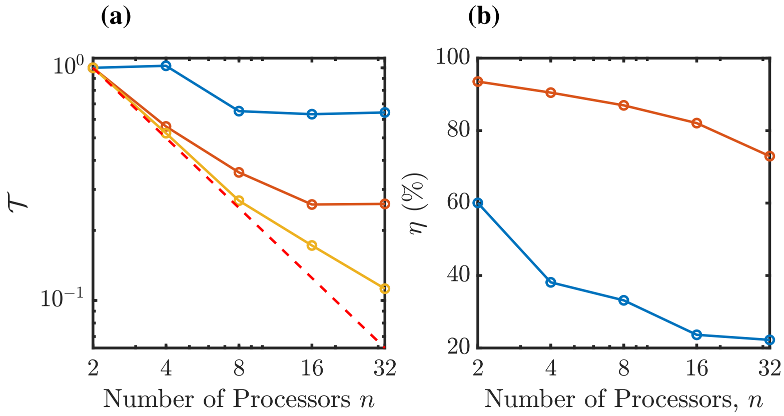


Fig. 6 Performance metrics of the parallel Immersed Boundary Projection Method (IBPM). (a) Relative computational time ($\mathcal{T}$) as a function of the number of processors (n). The dashed red line represents the ideal speedup, 2^{-n}. (b) Parallelization efficiency ($\eta(\%)$) as a function of the number of processors (n). This compares the time spent on non-IB calculations relative to the total computational time. The solid orange line represents the *parallel IBPM*, the blue solid line represents the *serial IBPM*, and the yellow solid line represents the *non-IB*.

C. Parallel scaling

To evaluate the computational scaling of the parallelized IB operators, we perform non-FSI simulations of the configuration shown in Fig. 1. The computational grid consists of 32 points in the streamwise direction, 200 points in the wall-normal direction, and 128 points in the spanwise direction. The two IB walls are planar and stationary, each discretized with 20 points in the streamwise direction and 40 points in the spanwise direction. This grid resolution is selected to allow simulations on up to 32 processors for the scaling study. Parallel performance is evaluated using five processor counts, $n \in \{2, 4, 8, 16, 32\}$.

To assess parallel performance, three cases are evaluated in Fig. 6. All three configurations include the parallelized channel-flow solver for non-IB operations. The first case, denoted *non-IB*, contains no immersed boundary implementation and serves as the benchmark. The second case, referred to as *serial IBPM*, uses the parallel channel-flow solver but retains a serial IB force operator; in this configuration, each MPI rank gathers its local data to the global domain before executing the IB operations. The third case, denoted *parallel IBPM*, corresponds to the fully parallel IBPM–channel-flow solver in which both interpolation and regularization operators are parallelized using support cells to exchange IB-related data across partitions. Comparing these three cases highlights the efficacy of the proposed parallelization strategy.

To compare the simulations, we evaluate the relative computational time $\mathcal{T}$, defined as

$$\mathcal{T} = \frac{\Delta T_i}{\Delta T_2}, \tag{37}$$

where ΔT_i is the computational time per time step using i processors and ΔT_2 is the corresponding time using two processors. Ideally, $\mathcal{T}$ should scale proportionally to $\frac{1}{2^n}$, represented by the red dashed line in Fig. 6(a). In practice, data-transfer overhead leads to deviations from this ideal line. Fig. 6(a) shows that, compared with the *non-IB channel flow* case, the *parallel IBPM* configuration exhibits reasonable scaling for processor counts up to $n = 16$. The saturation at $n = 32$ arises from nonuniform distributions of immersed boundary points in the spanwise direction: at this resolution, some ranks still process a comparable number of IB points to those at $n = 16$. Because the IBPM operations require synchronization among all ranks, the overall computational time does not improve further. It is expected that simulations with larger domains and more extensive immersed boundaries would exhibit better scaling characteristics.

To further quantify the effectiveness of the parallelization, we evaluate the parallelization efficiency

$$\eta(\%) = \frac{\Delta t_{\text{non-IB}}}{\Delta t_{\text{total IB}}} \times 100, \tag{38}$$

where $\Delta t_{\text{non-IB}}$ is the time spent on operations associated with the original channel-flow solver that are reused in the IB surface stress computation and $\Delta t_{\text{total IB}}$ is the total time required to compute the IB surface stresses. An efficiency of $\eta(\%) = 100$ corresponds to an ideal scheme where no additional time is consumed by IB-related operations. Conversely, values approaching zero indicate that the IB computations (interpolation, spreading, communication) dominate the total cost.

As shown in Fig. 6(b), without parallelization of the IB operators, the *serial IBPM* configuration spends up to 80% of the total time on IB-related operations. In contrast, the *parallel IBPM* configuration reduces this fraction to approximately 30% under the same processor count. This demonstrates that the support-cell implementation substantially reduces the communication and data-gathering overhead, resulting in significantly improved performance for multi-processor IBPM simulations.

D. Prescribed traveling-wave wall deformation at $\mathrm{Re}_\tau \approx 186$

To test the IB method's ability to capture wall-bounded turbulence with moving walls, we simulate a channel flow with prescribed traveling wave-like deformations applied to both the upper and lower walls of the setup shown in Fig. 1. This case is inspired by the study of Nakanishi et al. [25] that used boundary-fitted grids to investigate the effects of streamwise traveling waves on turbulent drag.

In our implementation, the wall motions are imposed via the IB method as described in the methodology section (see Fig. 1). The body-to-grid spacing ratio is reduced to $\Delta s/\Delta x = 1$ to prevent turbulent flow fluctuations at the IB walls. For this low ratio, the solution of the IB force in the IBPM becomes ill-conditioned [26], requiring many iterations to converge to a forcing solution that contains high frequency noise due to amplified discretization errors. Nevertheless, due to the smoothing effect of the IB operators, the flow solution itself will be unaffected by this noise. However, we will limit the iteration count per timestep stage to one iteration, which consistently gives us a no-slip error of around 10^{-3} times the bulk velocity.

The Reynolds number based on friction velocity and channel half-height in the undeformed channel is $\mathrm{Re}_\tau \approx 186$. Our computational domain has dimensions $L_x \approx 4\pi$, $L_y = 2$, and $L_z \approx 2\pi$, and uses $384 \times 380 \times 192$ computational cells with 132 cells in the y-direction inside each uniform grid zone and a hyperbolic tangent stretching function with a stretching factor of 2.6 for the central stretched grid patch. This results in grid spacings $\Delta x^+ \approx 6.09$, $0.18 \lessapprox \Delta y^+ \lessapprox 7.82$, and $\Delta z^+ \approx 6.09$ in wall units. Each IB surface is discretized using 384×192 points. The reference domain is similar with dimensions $L_x = 4\pi$, $L_y = 2$, and $L_z = 3.5$, using $256 \times 96 \times 128$ computational cells. Our higher number of cells in the y-direction is largely due to the extent of the uniform grid regions around each IB wall, which is one of the disadvantages of IB methods for turbulent flows.

As in the reference study, we use the channel half-height, h, and twice the bulk velocity, $2U_b$, as the length and velocity scales. The vertical deformations are prescribed as

$$\chi_U(x,t) = \frac{a}{kc}\sin\left[k(x-ct)\right], \quad \chi_L(x,t) = -\frac{a}{kc}\sin\left[k(x-ct)\right], \tag{39}$$

where a, c, and k denote the amplitude, phase speed, and wave number of the traveling wave, respectively. We consider two cases from the reference study while keeping the flow rate identical to the undeformed case: their case 6, which uses the parameter values $a = 0.1$, $c = 1$, and $k = 4$, and their case 20, which uses $a = 0.1$, $c = -1$, and $k = 2$,. These cases result in a drag decrease and drag increase, respectively, in the reference study.

Fig. 7 shows the time series of the mean pressure gradient required to maintain a constant flow rate for both deforming-wall cases, along with the corresponding pressure gradient for the undeformed channel that yields the same bulk flow rate. The figure also includes results from the reference study for the same configurations. Our method does not yet reproduce those results quantitatively, for several reasons. First, our simulation runtimes are much shorter than those in the reference study, so transient effects still affect our results. Second, the initial conditions differ: the reference study maps the undeformed flow onto a body-fitted grid with deformed walls, whereas our simulations begin from the undeformed case and increase the wall-deformation amplitude linearly from zero to its full value over one convective time unit. Consequently, we plot only the time-averaged pressure gradient from the reference study after its initial transient phase. Finally, our simulations do not satisfy the no-slip condition to machine precision, unlike the reference study, because we relax the tolerance of the iterative solver for computational efficiency. Given these differences in setup and simulation duration, a direct quantitative comparison with the reference study is not straightforward. Future work will focus on improving the conditioning of the method to enforce the no-slip condition more accurately on the IB surface and on running longer simulations to minimize transient effects.

E. Turbulent channel flow with compliant IB surface

In this section, we validate the FSI implementation by using it to replicate a standard turbulent channel, similar to the test case in Sec . IV.B, but now using a compliant wall setup. The Reynolds number based on the friction velocity and the channel half height is set to $\mathrm{Re}_\tau \approx 186$. Fig. 2 shows the configuration of the test case. The domain is chosen to

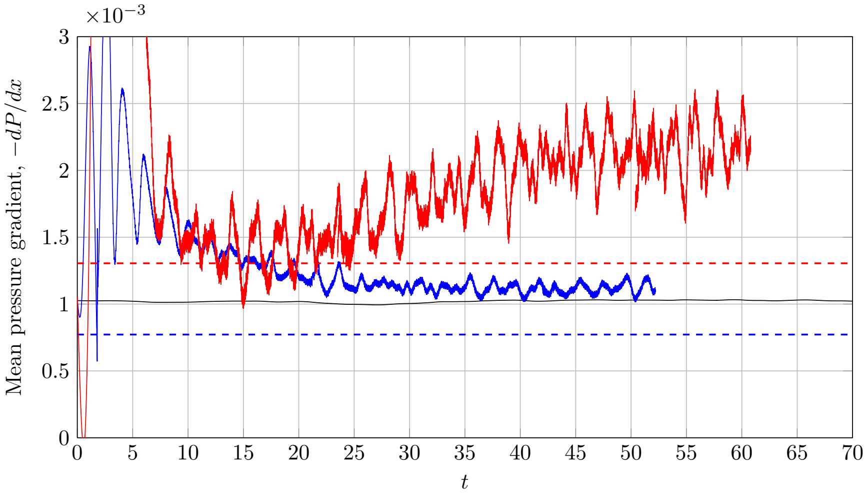


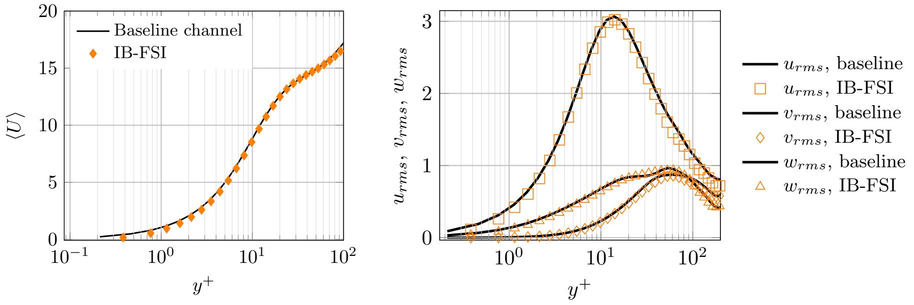
Fig. 7 Time series of the non-dimensional mean streamwise pressure gradient required to maintain a constant flow rate. Our present simulation results are shown with solid lines for three cases: an undeformed channel (black); a case using parameters $a = 0.1$, $c = 1$, and $k = 4$ (blue); and case using parameters $a = 0.1$, $c = -1$, and $k = 2$ (red). For comparison, the dashed lines represent the average value of the results published by Nakanishi et al. [25], calculated after the initial transient phase (specifically, after 50 time units for the first case and 250 time units for the second case).

represent a minimal flow unit configuration ($L_x = 1.718$, $L_y = 4$, $L_z = 0.859$). The number of computational cells in each Cartesian direction is now $32 \times 131 \times 16$. The grid in the y-direction is again stretched by a factor of 2.6, and a uniform grid region of three grid cells in the wall-normal direction is used on each side of the IB interface. Turbulent flow is initiated in both the bottom and the top channels. The planar IB patch is comprised of 32×16 points. This geometry maintains a body-to-grid spacing ratio $\Delta s/\Delta x \approx 1.0$ for the IB method. The tolerance for the convergence of the FSI iterations is set to 1×10^{-5}. For these parameters, the code takes approximately 1 FSI iteration per RK stage. The iterative solve for the surface stress is performed using the stabilized bi-conjugate gradient algorithm and approximately consumes 25 iterations to reach convergence. The spring-mass system drives the compliant patch deformations. The topmost mass corresponding to the spring-mass block system is centered at $x_{\text{block}} = 0.9$, $y_{\text{block}} = 2.0$, $z_{\text{block}} = 0.45$. A total of 8 subsurface blocks ($n_{\text{blocks}} = 8$) is used for the simulation. For our test simulation, we set a high value of mass and stiffness to the blocks that constitute the subsurface structure ($m_i = k_i = 10000$, $c_i = 0$). This ensures that the compliant patch behaves as a rigid, stationary wall and undergoes minimal deformations. This enables a comparison of the mean and rms statistics in the top channel with statistics from a non-IB rigid wall.

In Fig. 8, we show results from the test simulations, comparing the statistics obtained from the IB-FSI code with the non-IB rigid wall. The mean velocity statistics for the top channel, obtained with the IB-FSI code, are indicated as orange diamond markers in Fig. 8(a). We can see that the markers agree well with the solid thick profile that represents the mean velocity from the baseline channel [18]. The root mean squared components of the velocity fluctuations are shown in Fig. 8(b). We can notice that the markers representing the results from the IB-FSI code are shown to agree well with the thick lines representing the baseline non-IB channel. Overall, the profiles obtained from the IB-FSI code agrees well with the study of Bae et al. [18], indicating a good first pass implementation of the fully-coupled FSI simulation framework. As a part of future work, we continue to test and validate the setup for parameters that lead to significant deformations of the IB surface. After the validation stage, we will incorporate metamaterial characteristics such as material defects and di-atomicity into the subsurface structural model.

V. Conclusions

We presented a high-fidelity FSI framework based on an immersed boundary projection method, incorporated into a parallelized three-dimensional turbulent channel flow solver. The framework is compatible with a wide variety of (sub)surface structures, making it particularly suited for studying complex metamaterial interfaces. The current work extends an existing strongly coupled IB-FSI formulation for operation in high-Reynolds-number turbulent channel flows.



(a)

(b)

Fig. 8 (a): Statistics corresponding to the mean velocity profile in the top-channel (orange diamond markers) is compared to the baseline channel flow solver of Bae et al. [18]. (b): The root-mean squared velocity components of the turbulent flow field obtained from the IB-FSI code (square, diamond and triangle markers) compared to the baseline non-IB case (black solid lines).

To create this simulation framework, we introduced several novel modifications to existing FSI immersed boundary projection methods. First, we adapted the interpolation and regularization operations, inherent to the immersed boundary formalism, to periodic spatial domains. Second, we implemented a hybrid uniform–stretched grid to optimize computational load distribution while preserving delta function accuracy near IB surfaces. Third, we incorporated the surface stresses provided as a byproduct of the immersed boundary formulation, to compute one-sided velocity gradients for the computation of the friction velocity. Fourth, we developed a parallel computing architecture of IB operations consistent with the underlying flow solver's domain decomposition. Finally, we constructed new spatially discrete operators to facilitate information transfer across a wide range of surface and sub-surface configurations, with specific focus for this manuscript placed on a spring-mass chain model for a metamaterial subsurface.

We demonstrated a preliminary validation of the framework using a set of benchmark cases. First, we performed a laminar and turbulent validation test and conducted a parallel scaling analysis. We then applied the method to a turbulent channel flow with prescribed traveling wave-like wall deformations, inspired by previous literature, but the comparison with the literature is limited due to differences in our setup. Lastly. we performed a preliminary test setup for validating the FSI implementation.

In our future work, we plan to further validate our IB method for deforming walls and develop fully coupled fluid–structure interaction (FSI) simulations in which the wall dynamics are governed by the behavior of resonant phononic materials. The modeling framework will incorporate the detailed dynamics of phononic crystals with embedded defects, as demonstrated in Ramakrishnan and Matlack's recent work [27]. These materials exhibit rich frequency-domain properties, including pass bands and localized defect modes, which can be tailored to interact with specific turbulent flow features. By exploiting these spectral characteristics, we aim to design surfaces that passively engage with and modulate near-wall coherent structures. Ultimately, we hope this novel approach will enable effective drag reduction by harnessing the inherent dynamics of engineered metamaterials.

Acknowledgments

This work is supported by AFOSR MURI under grant number FA9550-23-1-0299.

References

[1] Barnes, C. J., Willey, C. L., Rosenberg, K., Medina, A., and Juhl, A. T., "Initial Computational Investigation Toward Passive Transition Delay Using a Phononic Subsurface," *AIAA Scitech 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-1454, URL https://arc.aiaa.org/doi/abs/10.2514/6.2021-1454.

[2] Willey, C. L., Barnes, C. J., Chen, V. W., Rosenberg, K., Medina, A., and Juhl, A. T., "Multi-input multi-output phononic subsurfaces for passive boundary layer transition delay," *Journal of Fluids and Structures*, Vol. 121, 2023, p. 103936. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2023.103936, URL https://www.sciencedirect.com/science/article/pii/S0889974623001044.

[3] Fukagata, K., Kern, S., Chatelain, P., Koumoutsakos, P., and Kasagi, N., "Evolutionary optimization of an anisotropic compliant surface for turbulent friction drag reduction," *Journal of Turbulence*, Vol. 9, 2008, p. N35. https://doi.org/10.1080/14685240802441126, URL https://doi.org/10.1080/14685240802441126.

[4] Józsa, T. I., Balaras, E., Kashtalyan, M., Borthwick, A. G. L., and Viola, I. M., "Active and passive in-plane wall fluctuations in turbulent channel flows," *Journal of Fluid Mechanics*, Vol. 866, 2019, p. 689–720. https://doi.org/10.1017/jfm.2019.145.

[5] Arif, I., Lam, G. C. Y., Leung, R. C. K., and Naseer, M. R., "Distributed surface compliance for airfoil tonal noise reduction at various loading conditions," *Physics of Fluids*, Vol. 34, No. 4, 2022, p. 046113. https://doi.org/10.1063/5.0087350, URL https://doi.org/10.1063/5.0087350.

[6] Borazjani, I., Ge, L., and Sotiropoulos, F., "Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies," *Journal of Computational Physics*, Vol. 227, No. 16, 2008, pp. 7587–7620. https://doi.org/https://doi.org/10.1016/j.jcp.2008.04.028, URL https://www.sciencedirect.com/science/article/pii/S0021999108002490.

[7] Causin, P., Gerbeau, J., and Nobile, F., "Added-mass effect in the design of partitioned algorithms for fluid–structure problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 42, 2005, pp. 4506–4527. https://doi.org/https://doi.org/10.1016/j.cma.2004.12.005, URL https://www.sciencedirect.com/science/article/pii/S0045782504005328.

[8] Förster, C., Wall, W. A., and Ramm, E., "Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, No. 7, 2007, pp. 1278–1293. https://doi.org/https://doi.org/10.1016/j.cma.2006.09.002, URL https://www.sciencedirect.com/science/article/pii/S0045782506002544.

[9] Yang, J., and Stern, F., "Sharp interface immersed-boundary/level-set method for wave–body interactions," *Journal of Computational Physics*, Vol. 228, No. 17, 2009, pp. 6590–6616. https://doi.org/https://doi.org/10.1016/j.jcp.2009.05.047, URL https://www.sciencedirect.com/science/article/pii/S0021999109003209.

[10] Lācis, U., Taira, K., and Bagheri, S., "A stable fluid–structure-interaction solver for low-density rigid bodies using the immersed boundary projection method," *Journal of Computational Physics*, Vol. 305, 2016, pp. 300–318. https://doi.org/https://doi.org/10.1016/j.jcp.2015.10.041, URL https://www.sciencedirect.com/science/article/pii/S0021999115007184.

[11] Tian, F.-B., Dai, H., Luo, H., Doyle, J. F., and Rousseau, B., "Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems," *Journal of Computational Physics*, Vol. 258, 2014, pp. 451–469. https://doi.org/https://doi.org/10.1016/j.jcp.2013.10.047, URL https://www.sciencedirect.com/science/article/pii/S0021999113007237.

[12] Degroote, J., Bathe, K.-J., and Vierendeels, J., "Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction," *Computers & Structures*, Vol. 87, No. 11, 2009, pp. 793–801. https://doi.org/https://doi.org/10.1016/j.compstruc.2008.11.013, URL https://www.sciencedirect.com/science/article/pii/S0045794908002605, fifth MIT Conference on Computational Fluid and Solid Mechanics.

[13] Mori, Y., and Peskin, C. S., "Implicit second-order immersed boundary methods with boundary mass," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 25, 2008, pp. 2049–2067. https://doi.org/https://doi.org/10.1016/j.cma.2007.05.028, URL https://www.sciencedirect.com/science/article/pii/S0045782507002952, immersed Boundary Method and Its Extensions.

[14] Hou, G., Wang, J., and Layton, A., "Numerical Methods for Fluid-Structure Interaction — A Review," *Communications in Computational Physics*, Vol. 12, No. 2, 2012, p. 337–377. https://doi.org/10.4208/cicp.291210.290411s.

[15] Goza, A., and Colonius, T., "A strongly-coupled immersed-boundary formulation for thin elastic structures," *Journal of Computational Physics*, Vol. 336, 2017, pp. 401–411. https://doi.org/https://doi.org/10.1016/j.jcp.2017.02.027, URL https://www.sciencedirect.com/science/article/pii/S0021999117301201.

[16] Taira, K., and Colonius, T., "The immersed boundary method: A projection approach," *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 2118–2137. https://doi.org/https://doi.org/10.1016/j.jcp.2007.03.005, URL https://www.sciencedirect.com/science/article/pii/S0021999107001234.

[17] Bae, H. J., Lozano-Durán, A., Bose, S. T., and Moin, P., "Turbulence intensities in large-eddy simulation of wall-bounded flows," *Phys. Rev. Fluids*, Vol. 3, 2018, p. 014610. https://doi.org/10.1103/PhysRevFluids.3.014610.

[18] Bae, H. J., Lozano-Durán, A., Bose, S. T., and Moin, P., "Dynamic slip wall model for large-eddy simulation," *Journal of Fluid Mechanics*, Vol. 859, 2019, p. 400–432. https://doi.org/10.1017/jfm.2018.838.

[19] Eldredge, J. D., “A method of immersed layers on Cartesian grids, with application to incompressible flows,” *Journal of Computational Physics*, Vol. 448, 2022, p. 110716. https://doi.org/https://doi.org/10.1016/j.jcp.2021.110716, URL https://www.sciencedirect.com/science/article/pii/S0021999121006112.

[20] Kim, J., and Moin, P., “Application of a fractional-step method to incompressible Navier-Stokes equations,” *Journal of Computational Physics*, Vol. 59, No. 2, 1985, pp. 308–323. https://doi.org/https://doi.org/10.1016/0021-9991(85)90148-2.

[21] Wray, A. A., “Minimal storage time advancement schemes for spectral methods,” *NASA Ames Research Center, California, Report No. MS*, Vol. 202, 1990.

[22] Lozano-Duran, A., and Bae, H. J., “Turbulent channel with slip boundaries as a benchmark for subgrid-scale models in LES,” *Annual research briefs. Center for Turbulence Research*, Vol. 2016, 2016, pp. 97–103.

[23] Unverdi, S. O., and Tryggvason, G., “A front-tracking method for viscous, incompressible, multi-fluid flows,” *Journal of Computational Physics*, Vol. 100, No. 1, 1992, pp. 25–37. https://doi.org/https://doi.org/10.1016/0021-9991(92)90307-K, URL https://www.sciencedirect.com/science/article/pii/002199919290307K.

[24] Newmark, N. M., “A Method of Computation for Structural Dynamics,” *Journal of the Engineering Mechanics Division*, Vol. 85, No. 3, 1959, pp. 67–94. https://doi.org/10.1061/JMCEA3.0000098, URL https://ascelibrary.org/doi/abs/10.1061/JMCEA3.0000098.

[25] Nakanishi, R., Mamori, H., and Fukagata, K., “Relaminarization of turbulent channel flow using traveling wave-like wall deformation,” *International Journal of Heat and Fluid Flow*, Vol. 35, 2012, pp. 152–159. https://doi.org/https://doi.org/10.1016/j.ijheatfluidflow.2012.01.007, URL https://www.sciencedirect.com/science/article/pii/S0142727X1200015X, 7th Symposium on Turbulence & Shear Flow Phenomena (TSFP7).

[26] Goza, A., Liska, S., Morley, B., and Colonius, T., “Accurate computation of surface stresses and forces with immersed boundary methods,” *Journal of Computational Physics*, Vol. 321, 2016, pp. 860–873. https://doi.org/https://doi.org/10.1016/j.jcp.2016.06.014, URL https://www.sciencedirect.com/science/article/pii/S002199911630239X.

[27] Ramakrishnan, V., and Matlack, K. H., “A quantitative study of energy localization characteristics in defect-embedded monoatomic phononic crystals,” *Journal of Sound and Vibration*, Vol. 614, 2025, p. 119164. https://doi.org/https://doi.org/10.1016/j.jsv.2025.119164, URL https://www.sciencedirect.com/science/article/pii/S0022460X2500238X.