



HAL
open science

A non-staggered coupling of finite element and lattice Boltzmann methods via an immersed boundary scheme for fluid-structure interaction

Zhe Li, Julien Favier

► **To cite this version:**

Zhe Li, Julien Favier. A non-staggered coupling of finite element and lattice Boltzmann methods via an immersed boundary scheme for fluid-structure interaction. *Computers and Fluids*, 2017, 143, pp.90 - 102. 10.1016/j.compfluid.2016.11.008 . hal-01403915

HAL Id: hal-01403915

<https://hal.science/hal-01403915>

Submitted on 28 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A non-staggered coupling of finite element and lattice Boltzmann methods via an immersed boundary scheme for fluid-structure interaction

Zhe Li^{a,*}, Julien Favier^b

^a*LHEEA laboratory, École Centrale de Nantes, UMR CNRS, Nantes, France*

^b*Aix-Marseille Université, CNRS, Centrale Marseille, M2P2 UMR 7340, Marseille, France*

Abstract

The paper presents a numerical framework for the coupling of finite element and lattice Boltzmann methods for transient problems involving fluid-structure interaction. The solid structure is discretized with the finite element method and integrated in time with the explicit Newmark scheme. The lattice Boltzmann method is used for the simulation of single-component weakly-compressible fluid flows. The two numerical methods are coupled via a direct-forcing immersed boundary method in a non-staggered way. Without subiteration within each time-step, the proposed method can ensure the synchronization of the time integrations, and thus the strong coupling of both subdomains by resolving a linear system of coupling equations at each time-step. Hence the energy transfer at the fluid-solid interface is correct, i.e. neither energy dissipation nor energy injection will occur at the interface, which can retain the numerical stability. A well-known fluid-structure interaction test case is adopted to validate the proposed coupling method. It is shown that the stability of the used numerical schemes can be preserved and a good agreement is found with the reference results.

Keywords: finite element method, lattice Boltzmann method, immersed boundary method, fluid-structure interaction, interface-energy-conserving

*Corresponding author

Email address: Zhe.Li@ec-nantes.fr (Zhe Li)

1. Introduction

Fluid-Structure Interaction (FSI) refers to interactions between two kinds of media, namely the fluid flow and the solid structure. Well-known examples include wind-induced oscillations of tall buildings, collapse of bridges caused by wind gusts, flow-induced vibrations of airfoils, and dynamics of blood flow through valves in veins etc. Thus, the robustness and accuracy of the numerical methods used for their simulations are of primary importance. Due to the different material and dynamical properties of fluid and solid media, one feature of FSI simulation is that the fluid and solid subdomains are generally modelled using different numerical methods, e.g. Eulerian fashion for fluid and Lagrangian fashion for solid, and different discretizations, both in space and time. In this paper, we propose a numerical framework to simulate two-way FSI problems by means of the coupling of Finite Element (FE) method for deformable solid structure and Lattice Boltzmann (LB) method for weakly-compressible fluid flow.

To simulate the solid structure, we adopt the FE method [1, 18], which is widely used in solving solid dynamics. Based on the total Lagrangian formulation [1], the adopted FE solver can be used to handle geometrical nonlinearities, such as moderate deformation-large rotation cases. For time integration, we choose to use the Newmark scheme [28] which enjoys great popularity due to its “single-step, single-solve” feature in linear as well as in nonlinear dynamic analysis. Besides, the Newmark time integrator also features controllable energy dissipation with two coefficients of the scheme [20]. More details of the chosen Newmark scheme will be discussed subsequently.

The fluid subdomain is simulated with the LB method, which is gaining much attention over the past decade. As an alternative to conventional numerical solvers based on the resolution of Navier-Stokes (NS) equations, the LB method consists in solving the discrete Boltzmann equation with collision and streaming processes, which can be proven to fully recover the macroscopic NS equations through the Chapman-Enskog analysis [3]. Recently, Shan et al. [34]

have shown that the LB method can be systematically derived from the kinetic theory for hydrodynamics. In the present work, we apply the single-component isothermal LB model for the simulation of a weakly compressible laminar fluid flow interacted with a deformable solid structure via an Immersed Boundary (IB) method.

Initially proposed by Peskin [29] for simulating blood flows in the hearts, IB method is particularly useful for introducing moving solid objects in the fluid domain modelled with a fixed Eulerian mesh, which is the case for LB method. Since its appearance, IB method has many variants [27]. Recently, several IB-LB coupled schemes have been presented for simulating FSI problems: Wu and Shu [38, 39] proposed an implicit velocity correction-based IB-LB method for one-way (stationary solid boundary) FSI simulations; Favier et al. [12] proposed an IB-LB coupling scheme based on a prediction-correction substage within each time-step, which was applied to simulate the fluid interaction with moving and slender flexible objects; de Rosis [7] proposed an iterative strong IB-LB coupling method for simulating two-way FSI problems in the presence of slender deformable solid. In the present paper, we adopt the direct-forcing IB method proposed in [22], which can enforce the no-slip condition at the immersed boundaries by using an appropriate local width for each Lagrangian point.

In the previous publications [4, 19, 21], different FE-LB coupling strategies have been proposed for the simulation of FSI problems in the presence of deformable solid structure. However, all of them can be classified as staggered coupling procedures [13], in which there always exists a lag between the time integrations of fluid and solid subdomains. For example, in [4], the force vector for the FE solver needed to be guessed with a force predictor, because the fluid-structure coupling determines the effective force. In [21], one takes the interface force \mathbf{f}^n at the moment t^n to compute the fluid velocity \mathbf{u}^{n+1} at the next moment t^{n+1} , which actually depends on \mathbf{f}^{n+1} ; in order to reduce the time lag, structural predictors [31] have been used in [19, 21] to obtain a numerically stable simulation. This time-lag may sometimes degrade the stability or con-

vergence order of the numerical schemes. In spite of the staggered feature, the advantage of these coupling strategies is that one does not need to significantly modify the existing codes or softwares for the FSI implementation.

The main objective of the present paper is to present a synchronous or non-staggered FE-LB coupling framework without subiteration within each time-step via a direct-forcing IB method. The non-staggered and subiteration-free features make the proposed coupling method robust and efficient, respectively.

Indeed, coupling of heterogeneous numerical methods may be of great importance in terms of preserving numerical stability for FSI computations, especially when a partitioned coupling algorithm is adopted. Unlike monolithic coupling procedure, which treats computationally the fluid and solid subdomains as an entity, the partitioned procedure allows us to carry out separately the time integration of each subdomain [13]. This is particularly convenient when one prefers to solve the fluid and solid equations using different numerical methods or existing softwares, which are already designed and optimized for each individual subdomain [11]. However, partitioned coupling algorithm often suffers numerical instabilities, if no special synchronization techniques, such as subiterative methods [10] or structural predictors [31], are applied. With this kind of coupling procedure, e.g. the Conventional Serial Staggered (CSS) method [13, 31], the FSI computation may diverge rapidly, even though the individual schemes used in the fluid and solid subdomains are both numerically stable. This is usually due to the time lag between the time integrations of the fluid and solid subdomains [23, 24, 26, 31].

From the energy point of view [5, 6, 8, 15], this CSS coupling procedure algorithmically injects or dissipates energy at the fluid-solid interface. When too much algorithmic interface energy is injected to the system, the coupling simulation will be interrupted due to the numerical instability. In the context of FE-LB coupling method, Kollmannsberger et al. [19] have proposed a numerical framework for FSI simulations, in which they applied and evaluated the basic and improved CSS [13] coupling algorithms with the interface-energy-conservation criteria. They showed that with the first order structural predictor

one can have a numerically stable simulation for the specific test case. However, as they showed in the comparison, using an elaborate structural predictor in a staggered coupling procedure can only reduce the increasing artificial interface energy, i.e. it cannot ensure exactly a zero interface energy. Consequently, the numerical stability of the FSI simulation will highly depend on the investigated problem.

In summary, the most important feature of the proposed FE-LB coupling method is that it can rigorously ensure the zero-interface-energy condition during the whole period of numerical simulation, in such a way that the numerical stability can be preserved for the FSI computations. The key point of this method is to construct a linear system of coupling equations at each time-step. By resolving the equations with the dual Schur formulation [14], one can update the interface velocity and force to the next time-step. Since the interface velocity and force are resolved simultaneously, the interface status update is implicit. But it is worth noting that this implicit updating procedure is based on solving directly a linear system of equations, rather than based on a subiterative coupling procedure [7, 36].

The rest of the paper is organized as follows: Section 2 gives the mathematical formulations of the used numerical methods (FE, LB and IB methods); Section 3 starts with a brief presentation of the time lag issue for staggered methods in the framework of FE-LB coupling with Newmark time integrator, and then presents the proposed non-staggered coupling method; Section 4 shows the numerical results of the validation test cases; finally, the conclusions and discussions are drawn in Section 5.

2. Mathematical formulation and numerical methods

2.1. Finite element method for solid structure

2.1.1. Semi-discrete equations obtained through total Lagrangian formulation

As presented in [1], the weak total Lagrangian formulation for solid can be obtained by integrating the momentum equation multiplied by a virtual

displacement field over the initial configuration:

$$\int_{\Omega_s^0} \delta \mathbf{u}_s \cdot \left(\rho_s^0 \frac{\partial^2 \mathbf{u}_s}{\partial t^2} - \nabla_0 \cdot \mathbf{P} - \rho_s^0 \mathbf{b} \right) d\Omega_s^0 = 0 \quad (1)$$

where $\Omega_s^0(\mathbf{X})$ denotes the solid initial configuration, as shown in Figure 1, with \mathbf{X} being the time-independent Lagrangian or material coordinate. In addition, $\delta \mathbf{u}_s(\mathbf{X})$ denotes the virtual displacement field, $\rho_s^0(\mathbf{X})$ the initial solid density, $\mathbf{u}_s(\mathbf{X}, t)$ the solid displacement field defined as $\mathbf{u}_s(\mathbf{X}, t) = \mathbf{x} - \mathbf{X}$ where $\mathbf{x}(\mathbf{X}, t)$ denotes the Eulerian or spatial coordinate of the material point \mathbf{X} in the current solid configuration $\Omega_s(t)$. $\mathbf{P}(\mathbf{X}, t)$ is the nominal stress tensor and $\nabla_0 \cdot$ is the divergence operator with respect to the material coordinate \mathbf{X} . Finally, $\mathbf{b}(\mathbf{X}, t)$ is the body force vector per unit mass such as the gravity.

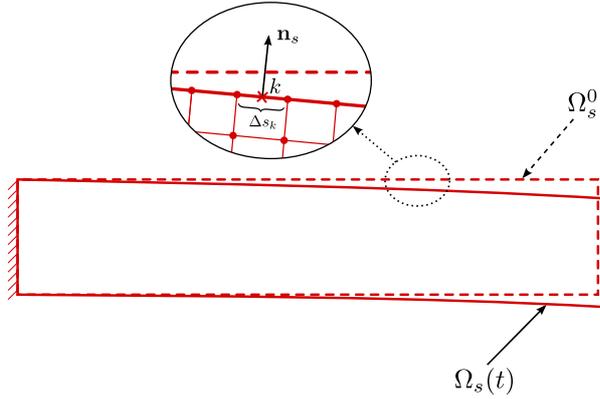


Figure 1: Initial and current solid configurations.

Eq. (1) is also referred to as the virtual work principle that can give us the semi-discrete solid equations in matrix form [1]:

$$\mathbf{M}_s \mathbf{a}_s = \mathbf{f}_{ext} - \mathbf{f}_{int} \quad (2)$$

where \mathbf{a}_s denotes the acceleration field defined as:

$$\begin{cases} \mathbf{a}_s = \frac{d\mathbf{v}_s}{dt} \\ \mathbf{v}_s = \frac{d\mathbf{u}_s}{dt} \end{cases} \quad (3)$$

with the displacement field $\mathbf{u}_s = [\mathbf{u}_s^1, \dots, \mathbf{u}_s^I, \dots, \mathbf{u}_s^N]^\top$, the velocity field $\mathbf{v}_s = [\mathbf{v}_s^1, \dots, \mathbf{v}_s^I, \dots, \mathbf{v}_s^N]^\top$ and the acceleration field $\mathbf{a}_s = [\mathbf{a}_s^1, \dots, \mathbf{a}_s^I, \dots, \mathbf{a}_s^N]^\top$, in which $\mathbf{u}_s^I = [u_s^{I,x}(t), u_s^{I,y}(t)]$, $\mathbf{v}_s^I = [v_s^{I,x}(t), v_s^{I,y}(t)]$ and $\mathbf{a}_s^I = [a_s^{I,x}(t), a_s^{I,y}(t)]$ are the I^{th} node's displacement, velocity and acceleration in x - and y -directions in 2D cases.

Additionally, \mathbf{M}_s , \mathbf{f}_{int} and \mathbf{f}_{ext} are the mass matrix, the internal and external nodal forces, respectively, which are given as:

$$\begin{cases} \mathbf{M}_s^{IJ} = \mathbf{I} \int_{\Omega_s^0} \rho_s^0 N_I N_J \, d\Omega_s^0 \\ (\mathbf{f}_{int}^I)^\top = \int_{\Omega_s^0} (\mathbf{B}_0^I)^\top \mathbf{P} \, d\Omega_s^0 \\ \mathbf{f}_{ext}^I = \int_{\Omega_s^0} N_I \rho_s^0 \mathbf{b}_0 \, d\Omega_s^0 + \int_{\Gamma_s^0} N_I \boldsymbol{\lambda}_0 \, d\Omega_s^0 \end{cases} \quad (4)$$

where \mathbf{I} is the identity matrix, $N_I(\mathbf{X})$ and $N_J(\mathbf{X})$ are the shape functions for the used 4-nodes quadrangle element at the I^{th} and J^{th} nodes, as shown in Figure 2.

The constant matrix \mathbf{B}_0^I is given as:

$$(\mathbf{B}_0^I)^\top = \left[\frac{\partial N_I}{\partial X}, \frac{\partial N_I}{\partial Y} \right] \quad (5)$$

where X and Y are the Lagrangian coordinates in x - and y -directions. Moreover, the nominal stress tensor \mathbf{P} is calculated by $\mathbf{P} = \mathbf{S} \cdot \mathbf{F}^\top$ with \mathbf{S} being the second Piola-Kirchhoff (PK2) stress and $\mathbf{F} = \partial \mathbf{x} / \partial \mathbf{X}$ denoting the deformation gradient. In the present work, we consider an elastic solid structure, hence the PK2 stress \mathbf{S} is linearly related to the Green-Lagrange strain \mathbf{E} by means of a constant fourth-order tensor \mathbf{C} , i.e. $\mathbf{S} = \mathbf{C} : \mathbf{E}$ where $\mathbf{E} = 1/2(\mathbf{F}^\top \cdot \mathbf{F} - \mathbf{I})$, which is referred to as the Saint Venant-Kirchhoff material model.

In the present paper, no external body force is considered for the structure, hence $\mathbf{b}_0 = \mathbf{0}$. In addition, $\boldsymbol{\lambda}_0$ denotes the external surficial force acted on the initial configuration Ω_s^0 .

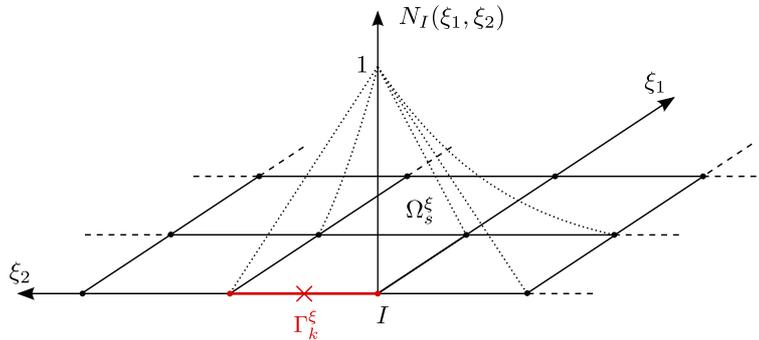


Figure 2: Shape function for the 4-nodes quadrangle element at the I^{th} node in the parent configuration Ω_s^ξ .

2.1.2. Time integration with Newmark scheme

To integrate in time this semi-discrete system of equations (2), we choose to apply the Newmark time integrator [28]:

$$\begin{cases} \mathbf{u}_s^{n+1} = \mathbf{u}_s^n + \Delta t \mathbf{v}_s^n + \frac{\Delta t^2}{2} [(1 - 2\beta)\mathbf{a}_s^n + 2\beta\mathbf{a}_s^{n+1}] \\ \mathbf{v}_s^{n+1} = \mathbf{v}_s^n + \Delta t [(1 - \gamma)\mathbf{a}_s^n + \gamma\mathbf{a}_s^{n+1}] \end{cases} \quad (6)$$

where the superscripts n and $n + 1$ indicate the time instants t^n and t^{n+1} for the displacement \mathbf{u}_s , velocity \mathbf{v}_s and the acceleration \mathbf{a}_s . In addition, Δt denotes the constant time-step, which is the same for both the solid and fluid subdomains. Finally, β and γ are the two coefficients of the Newmark scheme. In the present work, we choose $\beta = 0$ and $\gamma = 0.5$ for the solid simulations, with which the Newmark time integrator is essentially an explicit central difference scheme possessing second-order accuracy in time and conditional stability. The time-step Δt should be smaller than the critical time-step Δt_{crit} determined by [1]:

$$\Delta t < \Delta t_{crit} = \min_e \frac{l_e}{c_e} \quad (7)$$

where l_e denotes the characteristic length of the e^{th} element and c_e is the current wave speed in the e^{th} element. Because the simulations are carried out with a constant time-step Δt , one has to preset a Δt under the stability condition (7). It is here noteworthy that although the condition (7) is valid only for linear

cases, it can provide a useful guide for choosing the time-step in nonlinear cases [1]. In this work, we estimate Δt_{crit} by setting $l_e = \Delta X$ and $c_e = \sqrt{E_s/\rho_s^0}$ where ΔX denotes the initial uniform spacing of the solid mesh and E_s is the Young's modulus of the elastic solid material.

2.1.3. Calculation of external nodal force vector with a geometric operator

In the present work, the FSI happens only at the fluid-solid interface Γ_I . For the FE solver, the velocity boundary condition is tackled with the elimination method, and the time-varying force boundary condition is imposed by using a geometric operator \mathbf{L}_p that relates the external nodal force \mathbf{f}_{ext} with the flow-induced force $\mathbf{\Lambda}$ as $\mathbf{f}_{ext} = -\mathbf{L}_p\mathbf{\Lambda}$:

$$\underbrace{\begin{bmatrix} \mathbf{f}_{ext}^1 \\ \mathbf{f}_{ext}^2 \\ \vdots \\ \mathbf{f}_{ext}^{N_s} \end{bmatrix}}_{\mathbf{f}_{ext}} = - \underbrace{\begin{bmatrix} \mathbf{L}_p^{1,1} & \mathbf{L}_p^{1,2} & \dots & \mathbf{L}_p^{1,N_k} \\ \mathbf{L}_p^{2,1} & \mathbf{L}_p^{2,2} & \dots & \mathbf{L}_p^{2,N_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}_p^{N_s,1} & \mathbf{L}_p^{N_s,2} & \dots & \mathbf{L}_p^{N_s,N_k} \end{bmatrix}}_{\mathbf{L}_p} \underbrace{\begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \vdots \\ \boldsymbol{\lambda}_{N_k} \end{bmatrix}}_{\mathbf{\Lambda}} \quad (8)$$

where $\mathbf{f}_{ext}^I = [f_{ext}^{I,x}, f_{ext}^{I,y}]^\top$ is the external nodal force for the node $I \in [1, N_s]$ with N_s being the total number of the solid nodes. $\boldsymbol{\lambda}_k = [\lambda_k^x, \lambda_k^y]^\top$ denotes the force per unit length in 2D cases, which is exerted on the k^{th} ($k \in [1, N_k]$) fluid-solid interface element, as shown in Figure 1, and N_k is the total number of the interface elements.

It is worth noting that this geometric operator \mathbf{L}_p is time-varying and entirely depends on the current solid geometry. As mentioned previously, we choose to use the explicit Newmark scheme with $\beta = 0$ and $\gamma = 0.5$, which allows us to explicitly calculate the new displacement field $\mathbf{u}_s^{n+1} = \mathbf{u}_s^n + \Delta t \mathbf{v}_n + 0.5 \Delta t^2 \mathbf{a}_s^n$ so that we can explicitly update the solid geometry with this newly obtained displacement field \mathbf{u}_s^{n+1} . By doing so, we avoid the iterative procedure usually required for the implicit Newmark schemes to handle such geometrical nonlinearity.

To construct the geometric operator \mathbf{L}_p , we follow the way proposed by the author in the previous work [24] for the coupling of SPH and FE methods. As

one can observe in Equation (8), if one imposes a force boundary condition $\lambda_k^x = -1$ and $\lambda_k^y = 0$, $\lambda_m^x = 0$ and $\lambda_m^y = 0$ with $m \in [1, N_k]$ and $m \neq k$, then the $(2k-1)^{\text{th}}$ column of \mathbf{L}_p is just the external nodal force \mathbf{f}_{ext} under such force boundary condition. Similarly, the $2k^{\text{th}}$ column of \mathbf{L}_p is the external force \mathbf{f}_{ext} with the force condition $\lambda_k^x = 0$ and $\lambda_k^y = -1$, $\lambda_m^x = 0$ and $\lambda_m^y = 0$. Moreover, the formulae for calculating \mathbf{f}_{ext} is given as follows:

$$\mathbf{f}_{ext}^I = \int_{\Gamma_I^0} N_I \boldsymbol{\lambda}_0 \, d\Gamma_I^0 = \int_{\Gamma_I} N_I \boldsymbol{\lambda} \, d\Gamma_I \quad (9)$$

where $\Gamma_I(t)$ is the current interface that is constituted of N_k interface element, i.e. $\Gamma_I = \sum_k \Gamma_k$. $\boldsymbol{\lambda}(\mathbf{x}, t)$ denotes the external force acted on the current solid configuration. Here we assume that the force is piece-wise constant at the interface and is applied on the geometric center of each element Γ_k . Following the method presented previously, only one interface element is charged at each time, hence \mathbf{f}_{ext} can be calculated as:

$$\mathbf{f}_{ext}^I = \boldsymbol{\lambda}_k \int_{\Gamma_k} N_I \, d\Gamma_k = \boldsymbol{\lambda}_k \int_{\Gamma_k^\xi} N_I J_\xi \, d\Gamma_k^\xi = \begin{bmatrix} \lambda_k^x \\ \lambda_k^y \end{bmatrix} \begin{bmatrix} \Delta s_k \\ \Delta s_k^\xi \end{bmatrix} \quad (10)$$

where Γ_k^ξ represents the element k in the parent domain as shown in Figure 2, and $J_\xi = \Delta s_k / \Delta s_k^\xi$ is the Jacobian of the mapping from the current to the parent configurations, with Δs_k and Δs_k^ξ being the element length (2D) in the current and parent configurations, respectively.

2.2. Lattice Boltzmann method for single-component fluid flow

In the lattice Boltzmann method the fluid state is updated by resolving the discrete Boltzmann equation:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau} [f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)] + \Delta t G_i(\mathbf{x}, t) \quad (11)$$

where $f_i(\mathbf{x}, t)$ denotes the distribution function at the site \mathbf{x} and the time t , in the i^{th} direction of the used D2Q9 [32] lattice for 2D cases, as shown in Figure 3, and \mathbf{c}_i is the i^{th} discrete velocity vector. The set of discrete velocity vectors

of D2Q9 lattice can be obtained in the following way:

$$\begin{bmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_i \\ \vdots \\ \mathbf{c}_8 \end{bmatrix} = \frac{\Delta x}{\Delta t} \begin{bmatrix} 0 & +1 & 0 & -1 & 0 & +1 & -1 & -1 & +1 \\ 0 & 0 & +1 & 0 & -1 & +1 & +1 & -1 & -1 \end{bmatrix}^\top \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} \quad (12)$$

where Δx denotes the uniform spacing of the lattice, Δt is the constant time-step, and \mathbf{e}_x and \mathbf{e}_y are two unit vectors in x - and y -directions.

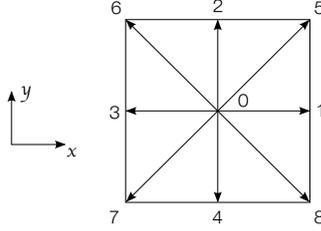


Figure 3: The D2Q9 lattice used in the present work for 2D LB simulations.

In Equation (11), the single-relaxation-time Bhatnagar-Gross-Krook (BGK) collision model is adopted and τ denotes the relaxation time. Here, $f_i^{eq}(\mathbf{x}, t)$ is referred to as the discrete equilibrium distribution function, which can be obtained by Hermite series expansion of the Maxwell-Boltzmann equilibrium distribution [34]:

$$f_i^{eq} = \rho_f \omega_i \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{v}_f}{c_{s,f}^2} + \frac{(\mathbf{c}_i \cdot \mathbf{v}_f)^2}{2c_{s,f}^4} - \frac{\mathbf{v}_f \cdot \mathbf{v}_f}{2c_{s,f}^2} \right] \quad (13)$$

where $\rho_f(\mathbf{x}, t)$ and $\mathbf{v}_f(\mathbf{x}, t)$ denote the macroscopic fluid density and velocity, $c_{s,f} = \sqrt{1/3}\Delta x/\Delta t$ the speed of sound, and ω_i the weight coefficients equaling $\omega_0 = 4/9$, $\omega_{1-4} = 1/9$ and $\omega_{5-8} = 1/36$ for D2Q9 lattice. In such isothermal single-component LB model the fluid pressure is calculate as $p_f = c_{s,f}^2 \rho_f$ and the kinematic viscosity ν is related to the relaxation time τ by $\nu = c_{s,f}^2(\tau - 0.5\Delta t)$.

Additionally, we adopt the scheme proposed by Guo et al. [16] in order to take into account the body-force effects in the fluid domain. In Equation (11),

the body-force-related term $G_i(\mathbf{x}, t)$ is given as [16]:

$$G_i = \left(1 - \frac{\Delta t}{2\tau}\right) \omega_i \left[\frac{\mathbf{c}_i - \mathbf{v}_f}{c_{s,f}^2} + \frac{(\mathbf{c}_i \cdot \mathbf{v}_f)}{c_{s,f}^4} \mathbf{c}_i \right] \cdot \mathbf{F} \quad (14)$$

where $\mathbf{F}(\mathbf{x}, t)$ denotes the body force per unit volume acted on fluid at the site \mathbf{x} and the time t .

Finally, once one calculates all the distribution functions $f_i(\mathbf{x}, t + \Delta t)$ by means of Equation (11), we can update the macroscopic fluid status by the definition [16]:

$$\begin{cases} \rho_f = \sum_i f_i \\ \rho_f \mathbf{v}_f = \sum_i \mathbf{c}_i f_i + \frac{\Delta t}{2} \mathbf{F} \end{cases} \quad (15)$$

Note that this definition of the macroscopic fluid velocity should be used at each time instant in order to make the LB scheme fully recover the Navier-Stokes equations.

In the present work, no gravity effect is considered in the fluid subdomain, hence the body force \mathbf{F} in Equation (15) is just the force related to the immersed boundary method, which is used to relate the fluid and solid subdomains for FSI simulations.

2.3. Immersed boundary method

In this paper, we apply the IB method proposed by the authors in the previous work [22], in which the IB method was used to impose moving solid boundary conditions in the fluid flow. This IB method is based on the interpolated definition of the macroscopic fluid velocity (15) from the Eulerian (fluid) nodes to the Lagrangian (solid) points:

$$\mathcal{I}[\rho_f]_k \mathbf{v}_{f,k} = \mathcal{I} \left[\sum_i \mathbf{c}_i f_i \right]_k + \frac{\Delta t}{2} \mathbf{F}_k \quad (16)$$

where $\mathbf{v}_{f,k}$ and \mathbf{F}_k denote the fluid velocity and the IB-related force at the k^{th} solid element, respectively. $\mathcal{I}[\bullet]_k$ represents the interpolation operator defined

as:

$$\begin{aligned}\phi(\mathbf{x}_k, t) &= \mathcal{I} [\phi(\mathbf{x}, t)]_k = \int \phi(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{x}_k) d\mathbf{x} \\ &\simeq \sum_{j \in D_k} \phi(\mathbf{x}_j, t) \tilde{\delta}(\mathbf{x}_j - \mathbf{x}_k) \Delta x \Delta y\end{aligned}\quad (17)$$

which provides an interpolated value of a given variable $\phi(\mathbf{x}, t)$ at the k^{th} solid point $\mathbf{x}_k = \mathbf{x}(\mathbf{X}_k, t)$ with \mathbf{X}_k being the time-independent Lagrangian coordinate. This form of the interpolation operator makes use of the sampling property of the Dirac delta function $\delta(\mathbf{x})$. In Equation (17), $\phi(\mathbf{x}_j, t)$ is the value of ϕ at the j^{th} Eulerian (fluid) node located inside the support domain D_k of the k^{th} Lagrangian (solid) point, as shown in Figure 4. Finally, $\tilde{\delta}(\mathbf{x}_j - \mathbf{x}_k)$ is a mollifier or a smooth approximation to the Dirac delta function. In the present work, we adopt the mollifier proposed by Roma et al. [33]:

$$\tilde{\delta}(\mathbf{x}_j - \mathbf{x}_k) = \frac{1}{\Delta x} \tilde{\delta}_x \left(\frac{|x_j - x_k|}{\Delta x} \right) \frac{1}{\Delta y} \tilde{\delta}_y \left(\frac{|y_j - y_k|}{\Delta y} \right) \quad (18)$$

with:

$$\tilde{\delta}_x(r) = \tilde{\delta}_y(r) = \begin{cases} \frac{1}{3} \left(1 + \sqrt{-3r^2 + 1} \right) & 0 \leq r < 0.5 \\ \frac{1}{6} \left[5 - 3r - \sqrt{-3(1-r)^2 + 1} \right] & 0.5 \leq r < 1.5 \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

where r denotes the nondimensional distance between the fluid node and the solid point in each direction, i.e. $r = |x_j - x_k|/\Delta x$ or $r = |y_j - y_k|/\Delta y$.

Supposing that at the instant t^n all variables are already known, and we will resolve the LB equation in order to update the fluid state to the instant t^{n+1} . After calculating the new distribution functions f_i^{n+1} with Equation (11), we can update the fluid density by $\rho_f^{n+1} = \sum_i f_i^{n+1}$, but we cannot update the fluid velocity, since the IB-related body force \mathbf{F} is not known yet. To clarify this point, let us rewrite Equation (16) at the instant t^{n+1} :

$$\mathcal{I} \left[\rho_f^{n+1} \right]_k \mathbf{v}_{f,k}^{n+1} = \mathcal{I} \left[\sum_i \mathbf{c}_i f_i^{n+1} \right]_k + \frac{\Delta t}{2} \mathbf{F}_k^{n+1} \quad (20)$$

In one-way FSI problems, such as the ones presented in [22], the fluid velocity at the solid point $\mathbf{v}_{f,k}^{n+1}$ is equal to the solid velocity under the no-slip condition,

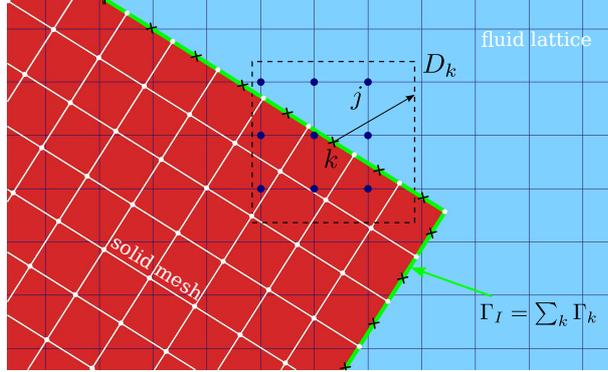


Figure 4: The fluid lattice and the support domain D_k of k^{th} interface point for IB method.

which is usually preset or imposed by the applied solid-motion law. As a result, one can directly calculate the IB-related force \mathbf{F}_k^{n+1} at solid points. However, in two-way FSI simulations, the velocity $\mathbf{v}_{f,k}^{n+1}$ can only be known after resolving the system of coupling equations, since the solid velocity at the instant t^{n+1} also depends on \mathbf{F}_k^{n+1} . That is why it is referred to as the *two-way coupled* problem.

In the proposed FE-LB coupling method, $\mathbf{v}_{f,k}^{n+1}$ and \mathbf{F}_k^{n+1} will be obtained together by resolving a system of coupling equations, then the force \mathbf{F}_k^{n+1} will be spread from the Lagrangian (solid) points onto the neighboring Eulerian (fluid) nodes by means of the following spreading operation:

$$\mathbf{F}(\mathbf{x}_j, t + \Delta t) = \sum_{k \in D_j} \mathbf{F}_k^{n+1} \tilde{\delta}_{jk}^{n+1} \epsilon_k^{n+1} \Delta s_k^{n+1} \quad (21)$$

where $\tilde{\delta}_{jk}^{n+1} = \tilde{\delta}(\mathbf{x}_j^{n+1} - \mathbf{x}_k^{n+1})$ and ϵ_k^{n+1} denotes the numerical width of the k^{th} solid segment at t^{n+1} , which is obtained by resolving a linear system of equations in order to enforce the reciprocity of the interpolation-spreading operations [30, 12, 22]. Finally, Δs_k^{n+1} is the length of the k^{th} segment at t^{n+1} . Note that the use of the explicit Newmark scheme allows us to explicitly update the structural geometry so that we can calculate the $\tilde{\delta}_{jk}^{n+1}$, ϵ_k^{n+1} and Δs_k^{n+1} independently of the coupling procedure within each time-step.

It is here noteworthy that in the present interpolation-based IB method for the FE-LB coupling, the LB equation is solved on all the lattice nodes, even for

the ones inside the solid domain. This allows us to apply the same treatment for all the lattice nodes and no special initialization technique is required for the freshly appeared real fluid nodes.

3. FE-LB coupling algorithm via an IB method

Physically, at the fluid-solid interface, there exist the continuities for the velocity and the force, respectively. However, this is not always true in the numerical simulations, due to the use of a straightforward or simple coupling algorithm. For example, when using FE method with Newmark time integrator for solid structure, the discrete system of equations for solid is given as:

$$\begin{cases} \mathbf{M}_s \mathbf{a}_s^{n+1} = \mathbf{f}_{ext}^{n+1} - \mathbf{f}_{int}^{n+1} \\ \mathbf{u}_s^{n+1} = \mathbf{u}_s^n + \Delta t \mathbf{v}_s^n + \frac{\Delta t^2}{2} [(1 - 2\beta) \mathbf{a}_s^n + 2\beta \mathbf{a}_s^{n+1}] \\ \mathbf{v}_s^{n+1} = \mathbf{v}_s^n + \Delta t [(1 - \gamma) \mathbf{a}_s^n + \gamma \mathbf{a}_s^{n+1}] \end{cases} \quad (22)$$

where the external nodal force \mathbf{f}_{ext}^{n+1} , see Equation (8), depends on the displacement field \mathbf{u}_s^{n+1} and the flow-induced force $\mathbf{\Lambda}^{n+1}$ at the interface and next time-step t^{n+1} . It is obvious that one cannot accomplish the time integration without knowing $\mathbf{\Lambda}^{n+1}$.

In order to simplify the coupling procedure, one may assume that the flow-induced force $\mathbf{\Lambda}$ is constant during each time-integration step for the solid sub-domain and this force has the value at the previous instant t^n , i.e. $\mathbf{\Lambda} = \mathbf{\Lambda}^n$ for $t \in (t^n, t^{n+1}]$. With this assumption, \mathbf{f}_{ext}^{n+1} can be calculated, so that the solid state can be updated to the next time-step. Then the fluid solver receives the updated solid geometry and velocity in order to finish the time integration. This coupling procedure is referred to as the Conventional Serial Staggered (CSS) coupling algorithm [13].

As presented previously in Section 1, the CSS coupling algorithm might inject artificial or algorithmic energy into the coupled system, which sometimes causes numerical instability for the FSI simulation. The *incremental interface energy* ΔE_I can be used to measure the energy injection or dissipation at the

fluid-solid interface due to the staggered coupling algorithm. The definition of ΔE_I is given as:

$$\Delta E_I = \sum_{k \in \Gamma_I} \int_{t^n}^{t^{n+1}} \mathcal{F}_{s,k} \cdot \mathbf{v}_{s,k} \, dt + \sum_{k \in \Gamma_I} \int_{t^n}^{t^{n+1}} \mathcal{F}_{f,k} \cdot \mathbf{v}_{f,k} \, dt \quad (23)$$

where $\mathcal{F}_{s,k}$ (or $\mathcal{F}_{f,k}$) is the interface force exerted on solid (or fluid), which can be calculated as $\mathcal{F}_{s,k} = \lambda_k \Delta s_k$ and $\mathcal{F}_{f,k} = \mathbf{F}_{k \in k} \Delta s_k$. $\mathbf{v}_{s,k}$ (or $\mathbf{v}_{f,k}$) is the solid (or fluid) velocity at the k^{th} interface element, as shown in Figure 4. Using the trapezoidal rule, one can approximately calculate ΔE_I as:

$$\Delta E_I \simeq \underbrace{\sum_{k \in \Gamma_I} \Delta t \bar{\mathcal{F}}_{s,k} \cdot \bar{\mathbf{v}}_{s,k}}_{W_s} + \underbrace{\sum_{k \in \Gamma_I} \Delta t \bar{\mathcal{F}}_{f,k} \cdot \bar{\mathbf{v}}_{f,k}}_{W_f} \quad (24)$$

where $\bar{\mathcal{F}}_{s,k} = (\mathcal{F}_{s,k}^{n+1} + \mathcal{F}_{s,k}^n)/2$ and $\bar{\mathcal{F}}_{f,k} = (\mathcal{F}_{f,k}^{n+1} + \mathcal{F}_{f,k}^n)/2$ denote the mean values of the interface forces at the k^{th} interface element, which act on the solid and the fluid, respectively. $\bar{\mathbf{v}}_{s,k} = (\mathbf{v}_{s,k}^{n+1} + \mathbf{v}_{s,k}^n)/2$ and $\bar{\mathbf{v}}_{f,k} = (\mathbf{v}_{f,k}^{n+1} + \mathbf{v}_{f,k}^n)/2$ are the mean velocities of solid and fluid at the interface. W_s and W_f represent the external work transferred to the solid and fluid subdomains at the interface. Clearly, when the force and velocity continuity conditions are ensured, i.e. $\mathcal{F}_{s,k}(t) = -\mathcal{F}_{f,k}(t)$ and $\mathbf{v}_{s,k}(t) = \mathbf{v}_{f,k}(t) \, \forall t \in [\dots, t^{n-1}, t^n, t^{n+1}, \dots]$, the quantity of the incremental interface energy $\Delta E_I = W_s + W_f$ is zero. Since the CSS algorithm cannot ensure the continuities at the interface, this quantity ΔE_I is generally not zero with the CSS algorithm.

3.1. The CSS coupling algorithm

In order to show the importance of the interface-energy-conserving feature of the proposed coupling method, we provide a comparison with the CSS coupling algorithm in the context of FE-LB coupling simulation. The procedure of this staggered algorithm is illustrated in Figure 5 and can be briefly summarized as:

- ① The interface force field $\mathbf{\Lambda}^n$ is sent to the solid solver, considered as the flow-induced force acted on the structure at t^{n+1}

- ② Using this force boundary condition, the solid solver makes use of the Newmark time integrator (6) to update the solid status to the next instant t^{n+1} : \mathbf{u}_s^{n+1} , \mathbf{v}_s^{n+1} , \mathbf{a}_s^{n+1}
- ③ Meanwhile, the fluid solver calculates the new distribution function f_i^{n+1} by Equation (11) with the fluid status f_i^n , ρ_f^n , \mathbf{v}_f^n and \mathbf{F}^n known from the previous time-step
- ④ The newly obtained structural geometry and velocity condition are sent to the fluid solver as a preset solid boundary condition, with which one will determine the new IB-related body force \mathbf{F}^{n+1} so as to fully update the macroscopic fluid status to t^{n+1} with Equation (15): ρ_f^{n+1} and \mathbf{v}_f^{n+1}
- ⑤ Go to ① for the next time-step

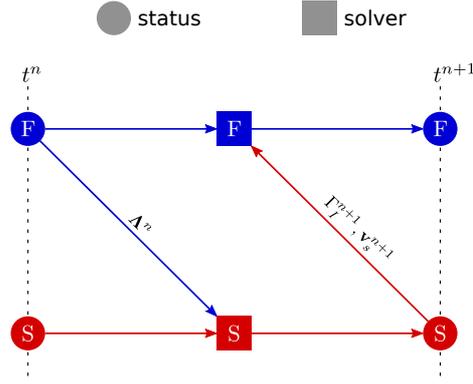


Figure 5: The Conventional Serial Staggered (CSS) coupling algorithm.

As one can observe, there is a time lag in this CSS coupling procedure, because the solid solver takes the interface force $\mathbf{\Lambda}^n$ as the force boundary condition at the instant t^{n+1} . Such assumption ruins the force continuity at the fluid-solid interface, which is the main reason why such staggered coupling method suffers sometimes from numerical instabilities [23, 24, 26].

Given the definition of the incremental interface energy Equation (23) and Equation (24), one can evaluate ΔE_I for the CSS coupling algorithm over one

time-step as follows:

$$\Delta E_I^{CSS} = \Delta t \sum_{k \in \Gamma_I} \bar{\mathbf{v}}_{s,k} \cdot \left(\frac{\mathcal{F}_{s,k}^{n+1} + \mathcal{F}_{s,k}^n}{2} + \frac{\mathcal{F}_{f,k}^{n+1} + \mathcal{F}_{f,k}^n}{2} \right) \quad (25)$$

where we applied the velocity continuity condition $\mathbf{v}_{s,k}(t) = \mathbf{v}_{f,k}(t)$, because in this CSS algorithm (Figure 5) the fluid and solid solvers share the same interface velocity. However, the interface force is not the same for the two solvers, because:

$$\begin{cases} \mathcal{F}_{s,k}^{n+1} = -\mathbf{F}_k^n \epsilon_k^{n+1} \Delta s_k^{n+1} \\ \mathcal{F}_{f,k}^{n+1} = \mathbf{F}_k^{n+1} \epsilon_k^{n+1} \Delta s_k^{n+1} \end{cases} \quad (26)$$

As a result, the value of ΔE_I^{CSS} might have positive or negative value, corresponding to an algorithmic energy injection or dissipation, respectively, at the interface for the whole coupled system. As mentioned in the paper of Combescure and Gravouil [5], this interface energy condition is crucial in terms of numerical stability for the time integration of the coupled system. Indeed, even if the individual schemes for solid and fluid subdomains are both numerically stable, the FSI simulation will still encounter instability issues when too much algorithmic interface energy is injected into the system.

Farhat and Lesoinne [9] has proposed an Improved Serial Staggered (ISS) coupling method which does not introduce errors of energy exchange at the fluid-solid interface. However, this method is not easy to be applied in the FE-LB coupling context, because neither the Newmark integrator nor the LB solver relies on the variable's state at the mid-time-step $t^{n+1/2}$.

3.2. The proposed non-staggered FE-LB coupling method

We shall provide here the details of the proposed non-staggered FE-LB coupling method via an IB scheme. As an implicit coupling procedure, it consists in resolving a system of coupling equations involving the solid and fluid equations, and the continuity conditions at the interface as well.

Starting with the solid coupling equations, let us write Equation (4) at the instant t^{n+1} :

$$\mathbf{M}_s \mathbf{a}_s^{n+1} = \mathbf{f}_{ext}^{n+1} - \mathbf{f}_{int}^{n+1} \quad (27)$$

Combining Equation (27) with the Newmark scheme (6) yields the solid equations that will be used in the coupling system:

$$\mathbf{K}_s^c \mathbf{v}_s^{n+1} + \mathbf{L}_p^{n+1} \boldsymbol{\Lambda}^{n+1} = \mathbf{g}_s^{n+1} \quad (28)$$

where \mathbf{L}_p^{n+1} is the geometric operator constructed as previously described in Section 2.1., which is based on the explicitly updated solid geometry with the chosen explicit Newmark scheme ($\beta = 0$ and $\gamma = 0.5$), and $\boldsymbol{\Lambda}^{n+1}$ is the interface force field at t^{n+1} . In addition, \mathbf{K}_s^c and \mathbf{g}_s^{n+1} are calculated as:

$$\begin{cases} \mathbf{K}_s^c = \frac{2}{\Delta t} \mathbf{M}_s \\ \mathbf{g}_s^{n+1} = \mathbf{M}_s \left(\frac{2}{\Delta t} \mathbf{v}_s^n + \mathbf{a}_s^n \right) - \mathbf{f}_{int}^{n+1} \end{cases} \quad (29)$$

Since the new displacement field \mathbf{u}_s^{n+1} has been calculated with the explicit Newmark scheme, one can then update the internal nodal force field with Equation (4) and the Saint Venant-Kirchhoff material model. As a result, there are only two unknowns in Equation (28), which are \mathbf{v}_s^{n+1} and $\boldsymbol{\Lambda}^{n+1}$. Now, let us omit the $n + 1$ superscript for the already calculated operator or matrix, i.e. \mathbf{L}_p^{n+1} and \mathbf{g}_s^{n+1} , so that we can rewrite Equation (28) as:

$$\mathbf{K}_s^c \mathbf{v}_s^{n+1} + \mathbf{L}_p \boldsymbol{\Lambda}^{n+1} = \mathbf{g}_s \quad (30)$$

As for the fluid coupling equations, by multiplying Equation (20) with the numerical local width ϵ_s^{n+1} of the k^{th} IB-segment, we can obtain:

$$\frac{2\epsilon_k^{n+1}}{\Delta t} \mathcal{I} \left[\rho_f^{n+1} \right]_k \mathbf{v}_{f,k}^{n+1} - \epsilon_k^{n+1} \mathbf{F}_k^{n+1} = \frac{2\epsilon_k^{n+1}}{\Delta t} \mathcal{I} \left[\sum_i \mathbf{c}_i f_i^{n+1} \right]_s \quad (31)$$

where $-\epsilon_k^{n+1} \mathbf{F}_k^{n+1} = \boldsymbol{\lambda}_k^{n+1}$ represents the interface force per unit length exerted on the solid at the k^{th} interface element or segment. Once again, let us omit the $n + 1$ superscript for the already known variables and rewrite Equation (31) in matrix form as:

$$\mathbf{K}_f^c \mathbf{v}_f^{n+1} + \boldsymbol{\Lambda}^{n+1} = \mathbf{g}_f \quad (32)$$

with:

$$\mathbf{K}_f^c = \begin{bmatrix} \mathbf{K}_{f,1}^c & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{f,2}^c & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{K}_{f,N_k}^c \end{bmatrix}, \mathbf{v}_f^{n+1} = \begin{bmatrix} \mathbf{v}_{f,1}^{n+1} \\ \mathbf{v}_{f,2}^{n+1} \\ \vdots \\ \mathbf{v}_{f,N_k}^{n+1} \end{bmatrix} \text{ and } \mathbf{g}_f = \begin{bmatrix} \mathbf{g}_{f,1} \\ \mathbf{g}_{f,2} \\ \vdots \\ \mathbf{g}_{f,N_k} \end{bmatrix} \quad (33)$$

where, for the k^{th} segment, we have:

$$\begin{cases} \mathbf{K}_{f,k}^c = \frac{2\epsilon_k^{n+1}}{\Delta t} \mathcal{I} \left[\rho_f^{n+1} \right]_k \mathbf{I} \\ \mathbf{g}_{f,k} = \frac{2\epsilon_k^{n+1}}{\Delta t} \mathcal{I} \left[\sum_i c_i J_i^{f,n+1} \right]_k \end{cases} \quad (34)$$

Now we have the solid and fluid coupling equations (30) and (32) with the unknown \mathbf{v}_s^{n+1} , \mathbf{v}_f^{n+1} and Λ^{n+1} . Obviously, one more equation is needed to solve the coupling system, which will come from the velocity continuity condition under the no-slip condition at the fluid-solid interface:

$$\mathbf{L}_s \mathbf{v}_s^{n+1} + \mathbf{v}_f^{n+1} = \mathbf{0} \quad (35)$$

where \mathbf{L}_s is another geometric operator that gives the solid velocity at the interface from the velocity field \mathbf{v}_s .

Regrouping the equations (30), (32) and (35) gives the coupling system of equations:

$$\begin{bmatrix} \mathbf{K}_s^c & \mathbf{0} & \mathbf{L}_p \\ \mathbf{0} & \mathbf{K}_f^c & \mathbf{I} \\ \mathbf{L}_s & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_s^{n+1} \\ \mathbf{v}_f^{n+1} \\ \Lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_s \\ \mathbf{g}_f \\ \mathbf{0} \end{bmatrix} \quad (36)$$

To solve such system of equations, we follow the procedure presented in [5, 24]:

- (1) Calculate the *free* velocities by:

$$\mathbf{v}_s^{free} = [\mathbf{K}_s^c]^{-1} \mathbf{g}_s \text{ and } \mathbf{v}_f^{free} = [\mathbf{K}_f^c]^{-1} \mathbf{g}_f \quad (37)$$

- (2) Calculate the *condensed* matrix \mathbf{H} , which is essentially the Poincaré-Steklov operator:

$$\mathbf{H} = \mathbf{L}_s [\mathbf{K}_s^c]^{-1} \mathbf{L}_p + [\mathbf{K}_f^c]^{-1} \quad (38)$$

(3) Calculate the interface force field Λ^{n+1} at the instant t^{n+1} :

$$\Lambda^{n+1} = \mathbf{H}^{-1} \left\{ \mathbf{L}_s [\mathbf{K}_s^c]^{-1} \mathbf{g}_s + [\mathbf{K}_f^c]^{-1} \mathbf{g}_f \right\} \quad (39)$$

(4) Calculate the *link* velocities:

$$\mathbf{v}_s^{link} = -[\mathbf{K}_s^c]^{-1} \mathbf{L}_p \Lambda^{n+1} \text{ and } \mathbf{v}_f^{link} = -[\mathbf{K}_f^c]^{-1} \Lambda^{n+1} \quad (40)$$

(5) Finally, regroup the *free* and *link* velocities in order to get the *total* velocities:

$$\mathbf{v}_s^{n+1} = \mathbf{v}_s^{free} + \mathbf{v}_s^{link} \text{ and } \mathbf{v}_f^{n+1} = \mathbf{v}_f^{free} + \mathbf{v}_f^{link} \quad (41)$$

Once Λ^{n+1} , \mathbf{v}_s^{n+1} and \mathbf{v}_f^{n+1} are updated, the solid solver will use \mathbf{v}_s^{n+1} to calculate the new acceleration field through Equation (6) with $\gamma = 0.5$. Meanwhile, the fluid solver will spread this newly obtained interface force field \mathbf{F}^{n+1} from the solid points onto the neighboring fluid nodes in order to accomplish the calculation of the macroscopic fluid velocity by Equation (15).

It is worth noting that in Equation (36) the matrix \mathbf{K}_s^c , \mathbf{K}_f^c , \mathbf{L}_p and \mathbf{L}_s are all independent of the unknowns \mathbf{v}_s^{n+1} , \mathbf{v}_f^{n+1} and Λ^{n+1} , due to the use of explicit Newmark time integrator. It can then be easily demonstrated that the present resolution procedure is mathematically equivalent with a direct one. This linearity allows us to split the final velocity solution field into two parts: the free velocity and the link velocity, so that the fluid and solid solvers can calculate first their free velocities individually and simultaneously. In addition, instead of directly solving the system of equations (36), we solve the condensed equation (38) which has a much smaller dimension than Equation (36). All these make the present resolution procedure very efficient.

Now, the proposed coupling algorithm is illustrated in Figure 6, and can be briefly reviewed as follows:

- ① The solid and fluid solvers carry out independently and simultaneously the first-stage calculation: the solid solver calculates the new displacement field \mathbf{u}_s^{n+1} using the explicit Newmark scheme and then update the

structural geometry; the fluid solver carries out the collision and streaming steps of LB method so that one can update the fluid density ρ_f^{n+1}

- ② The coupler interpolates the fluid information shown in Equation (20) from the fluid nodes to the newly updated solid points so as to prepare \mathbf{K}_f^c and \mathbf{g}_f , and then receives \mathbf{K}_s^c and \mathbf{g}_s from the solid solver in order to obtain the $\mathbf{\Lambda}^{n+1}$, \mathbf{v}_s^{n+1} and \mathbf{v}_f^{n+1} by resolving the coupling system of equations (36)
- ③ The solid solver receives \mathbf{v}_s^{n+1} and the fluid solver receives $\mathbf{\Lambda}^{n+1}$ from the coupler and then accomplish the individual integrations of one entire time-step

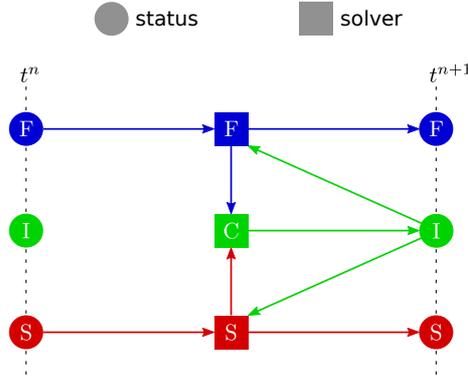


Figure 6: The proposed implicit strong coupling algorithm (F, S, I and C correspond to ‘Fluid’, ‘Solid’, ‘Interface’ and ‘Coupler’, respectively).

4. Numerical validations and discussions

In this part, we have adopted a widely used numerical test case [2, 19, 35, 36, 37] to validate the present FE-LB coupling method. Before showing the FSI results, we provide two numerical validations for the solid and fluid solvers alone. Finally, the FSI results are shown and the stability of the proposed algorithm is discussed.

4.1. Configuration of the numerical test case

This well-known test case was firstly proposed by Turek and Hron [37] for the numerical benchmarking of laminar flow-elastic structure FSI simulations. As shown in Figure 7, we consider an elastic solid bar immersed in a rectangular fluid flow channel. The deformable solid bar is attached to a fixed rigid cylinder, of which the geometric center is located near the inlet and slightly deviated from the horizontal center line of the channel. The geometric and material parameters are provided in Table 1 and Table 2, respectively.

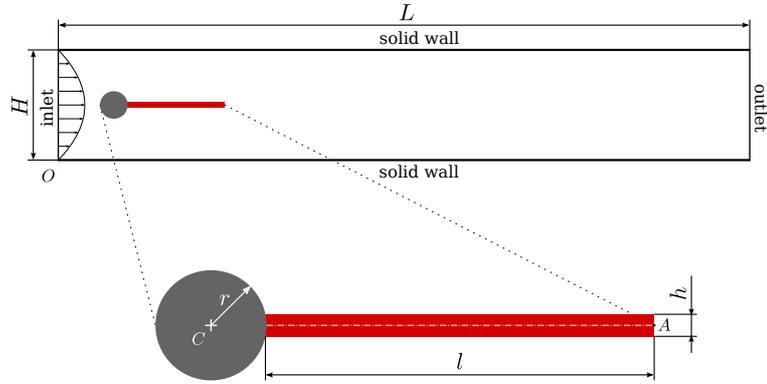


Figure 7: Configuration of the validation test case.

Table 1: Geometric parameters of the FSI test case.

Parameters	Values
Length of the flow channel L	2.5 m
Height of the flow channel H	0.41 m
Radius of the cylinder r	0.05 m
Length of the solid bar l	0.35 m
Height of the solid bar h	0.02 m
Coordinate of the point O	(0.0m, 0.0 m)
Coordinate of the cylinder center C	(0.2m, 0.2 m)
Coordinate of the point A	(0.6m, 0.2 m)

At the inlet of the flow channel, a parabolic velocity profile is imposed by:

$$\begin{cases} v_f^x(0, y) = 1.5\bar{U}\frac{y(H-y)}{(0.5H)^2} \\ v_f^y(0, y) = 0 \end{cases} \quad (42)$$

where $\bar{U} = 1$ m/s denotes the spatially mean inlet velocity, which will be used to determine the Reynolds number $Re = \bar{U}D/\nu_f = 100$ with D and ν_f being the diameter of the fixed cylinder and the kinematic fluid viscosity, respectively. Additionally, no-slip boundary conditions are applied at the upper and lower solid wall. All these three velocity boundary conditions are ensured by means of the scheme of Zou and He [40] for the LB simulation.

As for the outflow boundary condition, we apply a linear extrapolation scheme [25] for the distribution functions f_i for $i \in [1, 8]$, while tuning the f_0 in order to impose a prescribed exit density or pressure value. As a consequence, we can linearly extrapolate the fluid velocity and fix the fluid density to a certain value at the outlet.

Initially, the whole field is at rest and a smooth increase of the inlet velocity is imposed as:

$$\begin{cases} v_f^x(0, y, t) = 1.5\bar{U}\frac{y(H-y)}{(0.5H)^2}\frac{1 - \cos(0.5\pi t)}{2} & \text{if } t < 2.0 \text{ s} \\ v_f^x(0, y, t) = 1.5\bar{U}\frac{y(H-y)}{(0.5H)^2} & \text{otherwise} \end{cases} \quad (43)$$

Table 2: Material parameters of the FSI test case.

Parameters	Values
Initial fluid density ρ_f^0	10^3 kg/m ³
Fluid viscosity ν_f	10^{-3} m ² /s
Initial solid density ρ_s^0	10×10^3 kg/m ³
Solid Poisson ratio ν_s	0.4
Young's modulus E_s	1.4×10^6 Pa

The fluid subdomain is discretized with a uniform lattice of size $1250\Delta x \times 205\Delta y$ with $\Delta x = \Delta y$. As the LB equations are usually resolved with dimension-

less variables, we choose the *length* rescaling factor $\mathcal{C}_x = \Delta x^p / \Delta x^l = 0.002$ m and the *time* rescaling factor $\mathcal{C}_t = \Delta t^p / \Delta t^l = 10^{-4}$ s so that the lattice spacing $\Delta x^l = 1$ and the lattice time-step $\Delta t^l = 1$, where the superscripts p and l are devoted to *physical* and *lattice*, respectively. Hence, the velocity rescaling factor is determined as $\mathcal{C}_v = \Delta x^p / \Delta t^p = 20$ m/s. As a result, the lattice mean inlet velocity, the speed of sound in fluid, the viscosity and the relaxation time can be calculated by $\bar{U}^l = \bar{U}^p / \mathcal{C}_v = 0.05$, $c_{s,f}^l = \sqrt{1/3}$, $\nu_f^l = \bar{U}^l D^l / Re = 0.025$ and $\tau^l = 3\nu_f^l + 0.5 = 0.575$, respectively. Note that in such case we have a small Mach number $Ma = \bar{U}^l / c_{s,f}^l \simeq 0.0288 \ll 1$, which is consistent with the incompressible limit for the adopted LB model.

Here it is worth noticing that in the present work the solid mesh and the IB frontier have the same spacing as the fluid lattice, i.e. $\Delta X = \Delta s = \Delta x$. Additionally, the fluid and solid solvers use the same time-step. Hence, given the material parameters in Table 2, one can estimate the critical time-step for the solid solver with Equation (7) as $\Delta t_{crit} = \Delta x^p / \sqrt{E_s / \rho_s^0} \simeq 1.69 \times 10^{-4}$ s, which is larger than the used time-step $\Delta t^p = 10^{-4}$ s.

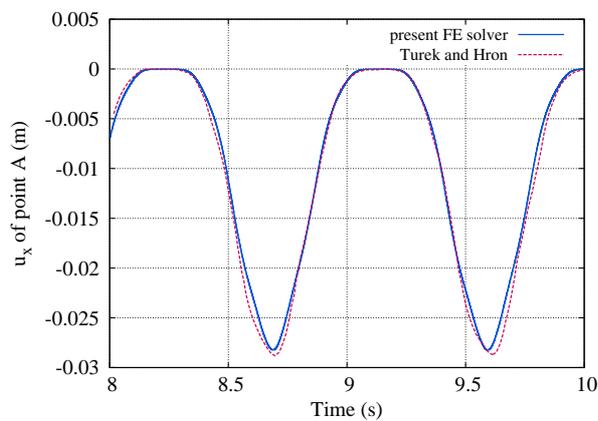
One may observe that the time-step is quite restricted. This is mainly due to two reasons. First, we use here the explicit Newmark time integrator for the solid structure. While this choice allows us to explicitly calculate the solid displacement at t^{n+1} without subiterations, it does limit the time-step because of its explicit feature as time integrator. Second, in the present work, we use the identical time-step for the solid and fluid solvers. In addition to this, the time-step Δt and the space-step Δx are tied up for LB solver. Consequently, one needs to choose the same time-step which should be appropriate for the solid as well as for the fluid solvers.

4.2. Validations of the solid and fluid solvers

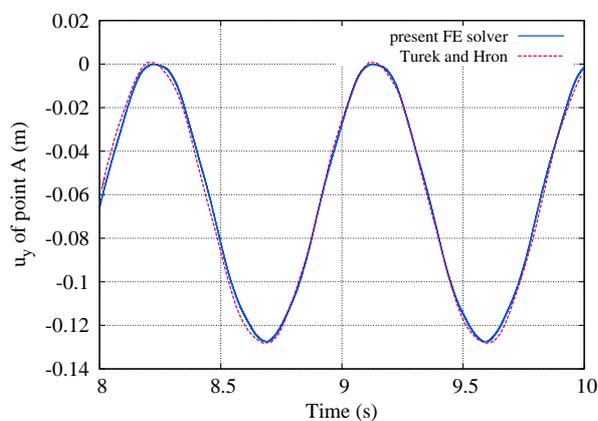
Before coupling the fluid and solid subdomains, it is necessary to validate individually the solid and fluid solvers using some pure solid and pure fluid test cases.

To do so, we have carried out firstly a pure solid test case, in which the solid

bar is only charged by the gravity body force. It corresponds to the “CSM3” test case proposed by Turek and Hron [37], where $\mathbf{g} = -g\mathbf{e}_z$ with $g = 2 \text{ m/s}^2$ and $\rho_s^0 = 10^3 \text{ kg/m}^3$. The other material parameters are the same as in Table 2. Figure 8 shows the time evolution of the displacement of the point A (see Figure 7) in x - and y -directions. A good agreement with the results of Turek and Hron [37] can be found in Figure 8, which allows us to validate the present FE solver in such kind of configuration for the solid simulation.



(a)



(b)

Figure 8: Time history of the point A 's displacement in x - and y -directions.

Secondly, we also validate the present LB solver with a steady test case

(“CFD2” in [37]) by measuring the drag and lift on the fixed and rigid cylinder-bar entity in the flow channel under the previously presented fluid boundary conditions. In Table 3, one can observe that the steady drag and lift obtained by the present LB solver are in good agreement with the ones in [37]. A difference less than 3% is found between the two results, which validates the present LB solver for fluid simulation. More validations for the LB solver can also be found in the previous work [22] of the authors.

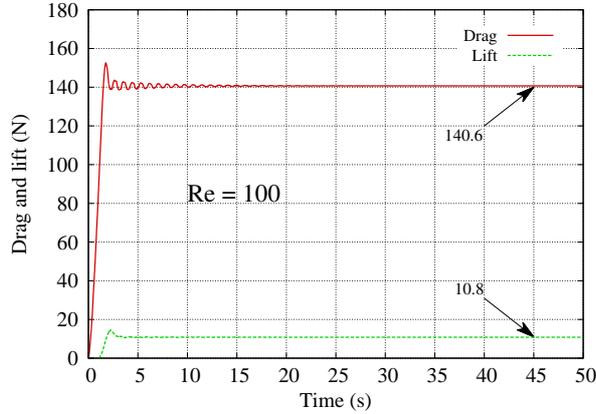


Figure 9: Drag and lift in the steady pure fluid test case.

Table 3: Drag and lift in the steady fluid solver validation test case.

	Drag (N)	Lift (N)
Present method	140.6	10.8
Turek and Hron [37]	136.7	10.5
Difference	2.85%	2.86%

4.3. Numerical results of the FSI simulations

4.3.1. Results with the proposed synchronous FE-LB coupling method

Now, we shall present the FSI results obtained with the proposed synchronous FE-LB coupling method. Figure 10 shows the vorticity field ω_f in the fluid flow and the stress component σ_s^{xx} in the solid bar at different instants.

Figure 11 gives the time history of the vertical (y -direction) displacement of the point A . From Figure 11-(a) one can observe that such flow-induced vibration begins to get stabilized after 10 seconds, and it can be considered as periodic after 12 seconds. Figure 11-(b) shows good agreement with the results of Turek and Hron [37], Kollmannsberger et al. [19] and Bhardwaj and Mittal [2].

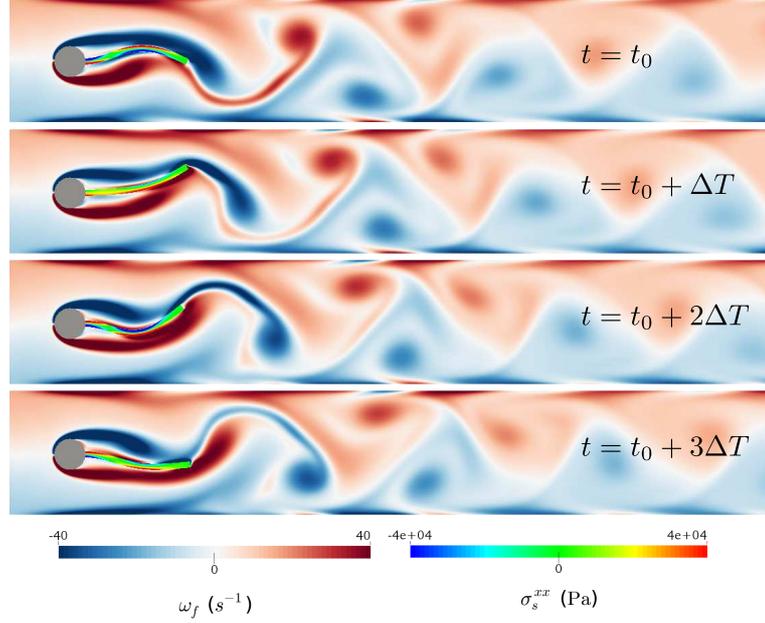
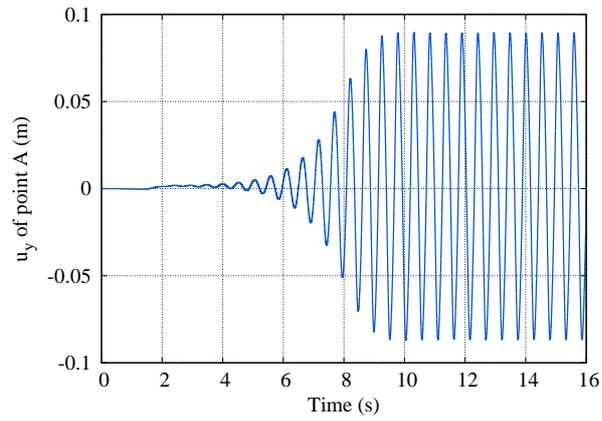
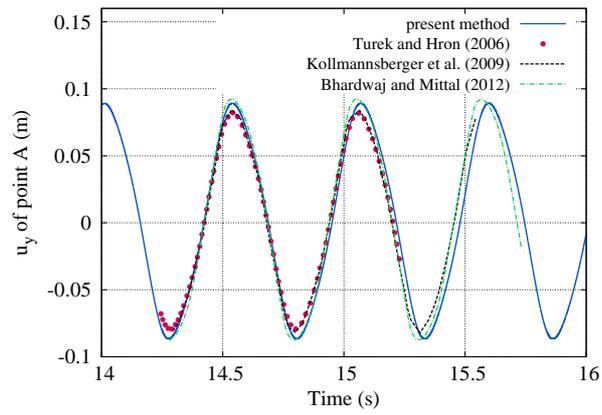


Figure 10: The vorticity field ω_f in the fluid flow and the stress component σ_s^{xx} in the solid bar at four different instants with $\Delta T = 0.1$ s.

A quantitative comparison is shown in Table 4. The vertical displacement of the point ‘A’ is expressed as: $u_y^A = u_{mean} \pm u_{amplitude}$ where u_{mean} and $u_{amplitude}$ denote the mean value and the amplitude of $u_y(A)$ during one period, respectively. To calculate the values of u_{mean} and $u_{amplitude}$, we adopt the same method presented in [37]: $u_{mean} = (u_{max} + u_{min})/2$ and $u_{amplitude} = (u_{max} - u_{min})/2$ with u_{max} and u_{min} being the maximal and minimal values of the vertical displacement of the point ‘A’, respectively. As shown in Table 4, the present method gives a mean displacement similar to those in [37, 19]. But there is a relatively obvious difference for the amplitude. The reason could



(a)



(b)

Figure 11: Time evolution of the displacement in y-direction of the point A: (a) the results obtained with the present method ($\Delta x = 0.002$ m); (b) the comparison with the reference results.

be the low accuracy of the used IB method which is only 1st-order accurate in space. Nevertheless, from Table 4 and Figure 11-(b), one can observe that the amplitude obtained by the present method is similar to the one in [2] which is also based on the use of an IB method. The frequency f of the flow-induced vibration in Table 4 is calculated as: $f = 1/T$ where T is the period time of the stabilized vibration. Here, one can find a good agreement with reference values.

Table 4: The vertical displacement of the point A and the frequency of the flow-induced vibration.

	u_y^A ($\times 10^{-3}$ m)	frequency (Hz)
Present method	1.32 ± 88.6	1.888
Turek and Hron [37]	1.23 ± 80.6	2.0
Kollmannsberger et al. [19]	1.20 ± 83.4	1.9
Bhardwaj and Mittal [2]	2.32 ± 91.2	1.9

The approximate delta function $\tilde{\delta}(r)$ plays an important role in the proposed framework, since it relates the Eulerian lattice nodes and the Lagrangian points via the adopted IB method. Figure 12 shows the result obtained with another approximate delta function of radius of 2 [29]:

$$\tilde{\delta}(r) = \begin{cases} \frac{1}{8} \left(3 - 2r + \sqrt{1 + 4r - 4r^2} \right) & 0 \leq r < 1 \\ \frac{1}{8} \left(5 - 2r - \sqrt{-7 + 12r - 4r^2} \right) & 1 \leq r < 2 \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

In Figure 12, the FSI result is compared with the previous one obtained using Equation (19). In this comparison, there is only one difference between the two FSI simulations: the approximate delta function. One can observe that there is little influence of the radius of the delta function.

In the present simulation, there are 257706 lattice nodes for LB method, 1936 mesh nodes for FE method, and 507 Lagrangian points for IB method. Two Xeon E5520 (2.27 GHz) cores were used: one for the LB solver, one for the FE solver. 16 seconds of physical time (160 000 time-steps) took 52 hours of

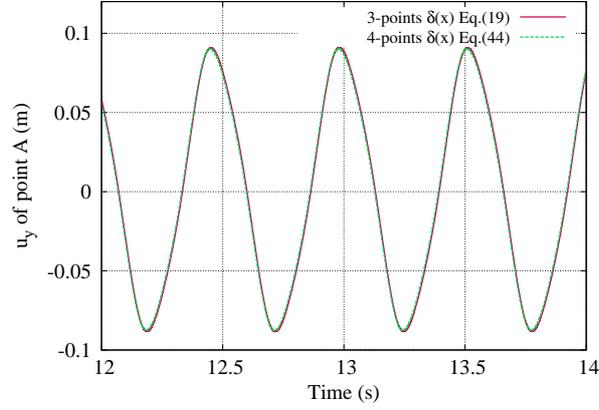


Figure 12: Influence of the approximate delta function.

CPU time. The time needed to compute each time-step of the FSI simulation is approximately 1.2 seconds.

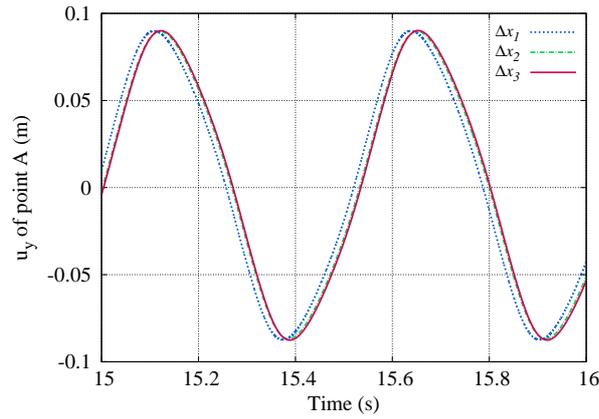


Figure 13: Numerical simulations with three different levels of mesh resolution: the coarse mesh ($\Delta x_1 = 0.002$ m), the intermediate mesh ($\Delta x_2 = 2/3\Delta x_1$) and the fine mesh ($\Delta x_3 = 1/2\Delta x_1$).

A mesh convergence study has been carried out using three levels of mesh resolution: the coarse (initial) mesh of $\Delta x_1 = 0.002$ m, the intermediate of $\Delta x_2 = 2/3\Delta x_1$ and the fine mesh of $\Delta x_3 = 1/2\Delta x_1$. As shown in Figure 13, one can observe that the numerical simulations have satisfactorily converged,

while decreasing the spacing size. Although a slight phase shift can be observed for the result using the coarse mesh with respect to the intermediate and fine meshes, little difference has been found among these three mesh resolutions for the amplitude and frequency of the flow-induced vibration.

Table 5: The vertical displacement of the point A and the frequency of the flow-induced vibration for three different levels of mesh resolution.

Mesh spacing Δx	u_y^A ($\times 10^{-3}\text{m}$)	frequency (Hz)
$\Delta x_1 = 0.002$ m	1.32 ± 88.6	1.888
$\Delta x_2 = 2/3\Delta x_1$	1.43 ± 88.3	1.886
$\Delta x_3 = 1/2\Delta x_1$	1.28 ± 88.8	1.886

Since the incremental interface energy is an important indicator for evaluating the applied coupling algorithm, we provide in Figure 14 the time history of the accumulated or total interface energy $E_I = 1/E_{ref} \sum_n \Delta E_I^{n \rightarrow n+1}$ where $E_{ref} = 0.5\rho_f^0 V_f \bar{U}^2$ is the reference kinetic energy in the fluid subdomain and $\Delta E_I^{n \rightarrow n+1}$ is calculated by Equation (24) over each time interval:

$$\begin{aligned} \Delta E_I^{n \rightarrow n+1} &\simeq \Delta t \sum_{k \in \Gamma_I} \bar{\mathcal{F}}_{s,k} \cdot \bar{\mathbf{v}}_{s,k} + \bar{\mathcal{F}}_{f,k} \cdot \bar{\mathbf{v}}_{f,k} \\ &= \Delta t \sum_{k \in \Gamma_I} \frac{\mathcal{F}_{s,k}^{n+1} + \mathcal{F}_{s,k}^n}{2} \cdot \frac{\mathbf{v}_{s,k}^{n+1} + \mathbf{v}_{s,k}^n}{2} + \frac{\mathcal{F}_{f,k}^{n+1} + \mathcal{F}_{f,k}^n}{2} \cdot \frac{\mathbf{v}_{f,k}^{n+1} + \mathbf{v}_{f,k}^n}{2} \end{aligned} \quad (45)$$

where $\mathcal{F}_{s,k}$ and $\mathcal{F}_{f,k}$ depend on the interface force field $\mathbf{\Lambda}$, and $\mathbf{v}_{s,k}$ and $\mathbf{v}_{f,k}$ are the solid and fluid velocity at the k^{th} interface element. Because of the implicit treatment of interface status in the proposed coupling method, the solid and fluid share the same $\mathbf{\Lambda}$ at each time-step. Hence $\mathcal{F}_{s,k}$ and $\mathcal{F}_{f,k}$ are always equal and opposite. In addition, the velocity continuity condition is assured by Equation (35). Consequently, the incremental interface energy is ensured to be zero *by construction* with the proposed coupling method. As one can observe that the absolute value of the accumulated interface energy is rigorously zero ($\sim 10^{-17}$).

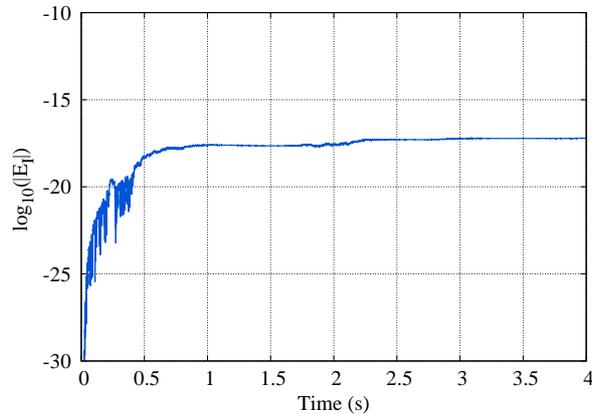


Figure 14: The time history of the absolute accumulated or total algorithmic interface energy introduced with the present strong coupling method.

4.3.2. Results with the CSS coupling algorithm

The CSS coupling algorithm is very straightforward and simple to be implemented for FSI simulations, especially when using two existing computation softwares. The drawback of CSS algorithm is that it cannot ensure the zero-interface-energy condition, and then sometimes encounters numerical instability. In order to show the importance of the interface-energy-conserving feature of the proposed method, we carried out a numerical simulation using the same configuration parameters as previously, except that the CSS coupling method (Section 3.1) is adopted for the coupling of FE and LB methods. We find that this CSS coupling algorithm cannot provide a numerically stable simulation for the chosen test case. The calculation diverged quickly after several hundred time-steps, as shown in Figure 15 by the curve entitled by “staggered: $\gamma = 0.50$ ”. The quickly increasing interface energy E_I shows that this CSS coupling algorithm cannot ensure a zero interface energy, i.e. too much algorithmic energy has been injected into the coupled system at the interface, which finally interrupted the simulation.

In this circumstance, one of the usual solutions is to choose a larger Newmark coefficient $\gamma > 0.50$, which introduces numerical dissipation in the solid

simulation, and might probably stabilize the FSI simulation. However, such solution does not work all the time and it decreases the order of accuracy of the Newmark time integrator, and more inconveniently one can never know beforehand how big γ needs to be in order to get a stable simulation. As shown in Figure 15, choosing a larger γ cannot stabilize the FSI simulation, even though it can slightly defer the divergence moment. In fact, the coefficient γ cannot be too much big either, for the sake of stability for the Newmark scheme [20]. We have here verified the superiority of the proposed non-staggered or synchronous algorithm in terms of numerical stability for FE-LB coupling.

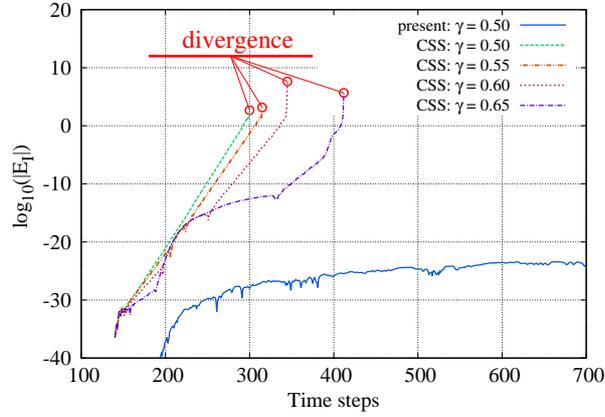


Figure 15: Comparison of the interface energy between the CSS and the proposed non-staggered coupling algorithms.

Another solution is to adopt subiterations within each time-step of the FSI simulation. From t^n to t^{n+1} , the subiteration procedure is given as follows:

- ① The solid solver sends an interface state $\Gamma_I(\mathbf{u}_s^{n+1}, \tilde{\mathbf{v}}_s)$ to the fluid solver, where $\tilde{\mathbf{v}}_s$ denotes an estimated solid velocity field. For the first subiteration step, we set $\tilde{\mathbf{v}}_s = \mathbf{v}_s^n$ in the present work
- ② With this interface state, the fluid solver updates the fluid state to the next instant t^{n+1} and then sends back to the solid solver the interface force field $\mathbf{\Lambda}^{n+1}$

- ③ Using this interface force field Λ^{n+1} , the solid solver calculates a new solid velocity field \mathbf{v}_s^{n+1} and the relative difference between \mathbf{v}_s^{n+1} and $\tilde{\mathbf{v}}_s$ by:

$$\mathcal{R} = \frac{\sqrt{(\mathbf{v}_s^{n+1} - \tilde{\mathbf{v}}_s)^\top (\mathbf{v}_s^{n+1} - \tilde{\mathbf{v}}_s) / N_s}}{\bar{U}} \quad (46)$$

where \bar{U} is the characteristic fluid velocity in Equation (42) and N_s is the total number of solid nodes

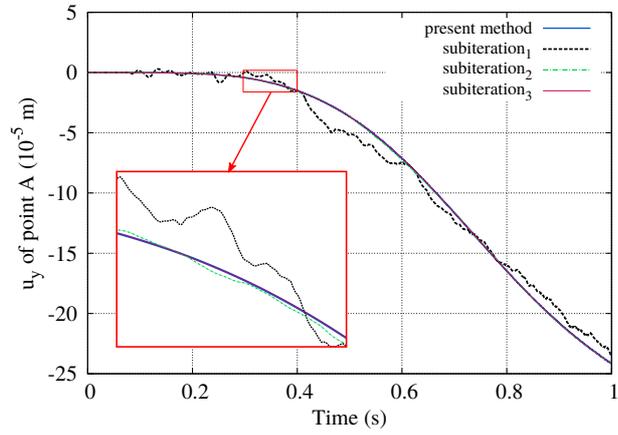
- ④ Compare \mathcal{R} with the prescribed critical value \mathcal{R}_c :

(4.1) If $\mathcal{R} < \mathcal{R}_c$: break the subiteration loop and proceed the next physical time-step of the FSI simulation

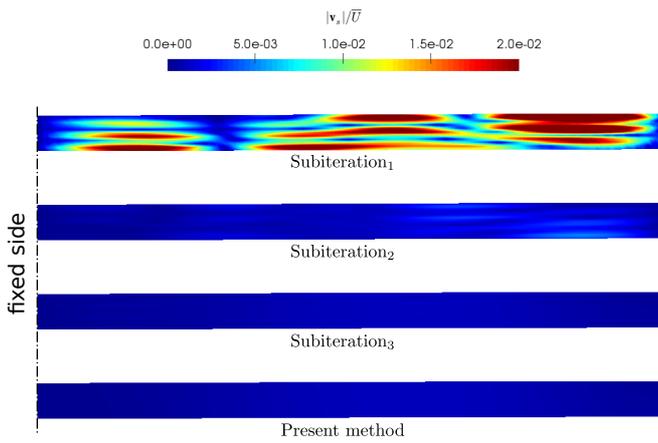
(4.2) Else: set $\tilde{\mathbf{v}}_s = \mathbf{v}_s^{n+1}$ and goto the step ①

Subiteration-based coupling algorithms are widely applied because of the simplicity of implementation. In addition, the subiteration steps help to reduce the interface energy so as to stabilize the FSI simulation and improve the accuracy. However, the subiterations are often costly and sometimes might diverge in the presence of strong fluid-structure coupling [17].

Figure 16 shows the numerical results obtained with the subiteration-based coupling algorithms using three different critical relative differences $\mathcal{R}_c^1 = 10^{-3}$, $\mathcal{R}_c^2 = 10^{-4}$ and $\mathcal{R}_c^3 = 10^{-5}$. In this comparison, one can observe that the staggered FSI simulations have been stabilized with the help of the subiterations at every time-step. In addition, as shown in Figure 16-(a), when the convergence criteria of subiteration is less strict, numerical oscillation occurs around the result obtained with the proposed non-staggered coupling method. This numerical oscillation can also be observed in the velocity magnitude field of the deformable solid bar, as shown in Figure 16-(b). Too much energy has been injected to the solid subdomain. Finally, the accumulated interface energy is given in Figure 17, where one can observe that the level of interface energy is higher with a less strict convergence criteria for the subiterations, and the interface energy is strictly zero with the proposed non-staggered algorithm.



(a)



(b)

Figure 16: The simulation results of the subiteration-based coupling algorithms: (a) the time evolution of the vertical displacement of the point A; (b) the velocity field of the deformable solid bar at the time $t = 0.4$ s.

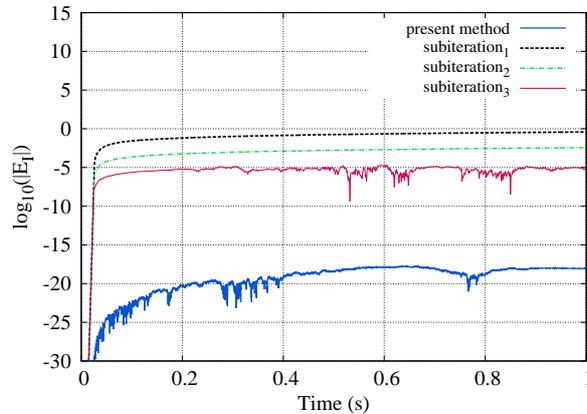


Figure 17: The time history of the absolute accumulated interface energy introduced with the subiteration-based coupling method.

In the present comparison, although the subiteration can stabilize the fluid-structure coupling and can give a similar result compared with the one obtained by the proposed non-staggered coupling method, the subiteration-based CSS method is theoretically much more costly, since it requires several subiterations at every physical time-step. Figure 18 shows the relative time consumption for the three difference convergence criteria \mathcal{R}_c : T is the mean CPU time used by the subiteration-based CSS method for one physical time-step and $T_{ref} = 1.2$ s is the CPU time with the proposed subiteration-free non-staggered coupling method.

5. Conclusions

In the present article, a subiteration-free numerical framework is proposed for simulating fluid-structure interaction problems, which is based on the coupling of finite element method for solid and lattice Boltzmann method for fluid via an immersed boundary scheme. The most important feature of the proposed coupling method is that the fluid and solid subdomains are coupled in a synchronous way, which means that there is no time lag between the two time integrations, so that the proposed coupling method can rigorously ensure

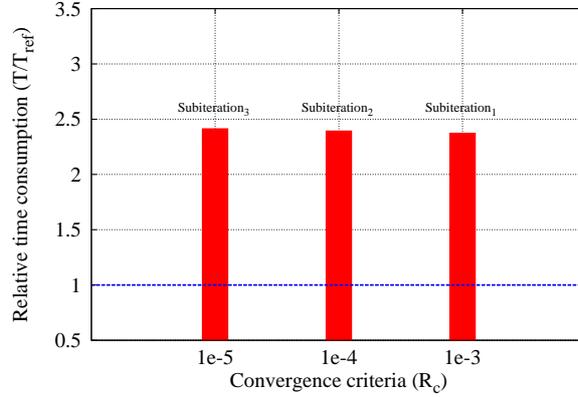


Figure 18: The mean relative CPU time for the subiteration-based CSS coupling method.

a zero algorithmic energy at the fluid-solid interface. In the adopted validation test case, the present result is found to be in good agreement with the reference ones. Additionally, a comparison with the conventional serial staggered coupling procedure has also been carried out. The result shows that, in this specific numerical framework, the staggered methods cannot give a stable simulation, even if one uses a relatively large numerical dissipation coefficient of Newmark time integrator. On the contrary, the proposed subiteration-free non-staggered coupling method can retain the numerical stability during the whole period of numerical simulation.

The advantage of proposed method is that it provides a stable subiteration-free numerical framework that couples two very different numerical methods for simulating fluid-structure interaction problems. The continuity conditions at fluid-solid interface are satisfied so that the incremental interface energy can be ensured to be zero by means of an implicit treatment of interface status. Nevertheless, some possible improvements could be (1) use of incompatible time-steps: in the present framework, the time-step must be the same for the fluid and solid solver, which might limit the efficiency of the coupling method; (2) use of implicit Newmark time integrator for solid: the present coupling is based on the use of explicit Newmark scheme that allows us to handle the geometri-

cal nonlinearity without subiteration within each time-step; (3) coupling via a higher-order immersed boundary method: the 1st-order accuracy of the adopted immersed boundary method might limit the accuracy of the FSI simulations. In the future, the next step of the present work is to use incompatible time-steps for fluid and solid, while still preserving the interface energy.

References

- [1] T. Belytschko, W.K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John Wiley & Sons, 2013.
- [2] R. Bhardwaj and R. Mittal. Benchmarking a Coupled Immersed-Boundary-Finite-Element Solver for Large-Scale Flow-Induced Deformation. *AIAA Journal*, 50:1638 – 1642, 2012.
- [3] S. Chapman and T.G. Cowling. *The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. Cambridge Mathematical Library. Cambridge University Press, 1970.
- [4] Jonathan R. Clausen, Daniel A. Reasor Jr., and Cyrus K. Aidun. Parallel performance of a lattice-Boltzmann/finite element cellular blood flow solver on the IBM Blue Gene/P architecture. *Computer Physics Communications*, 181:1013 – 1020, 2010.
- [5] A. Combescure and A. Gravouil. A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 191:1129 – 1157, 2002.
- [6] A. Combescure, A. Gravouil, and B. Herry. An algorithm to solve transient structural non-linear problems for non-matching time-space domains. *Computers & Structures*, 81(12):1211 – 1222, 2003.
- [7] A. De Rosis, S. Ubertini, and F. Ubertini. A partitioned approach for two-dimensional fluidstructure interaction problems by a coupled lattice boltzmann-finite element method with immersed boundary. *Journal of Fluids and Structures*, 45:202 – 215, 2014.
- [8] C. Farhat, L. Crivelli, and M. G eradin. On the spectral stability of time integration algorithms for a class of constrained dynamics problems. *AIAA 34th Structural Dynamics Meeting*, 1993.

- [9] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182:499 – 515, 2000.
- [10] C. Farhat, A. Rallu, K. Wang, and T. Belytschko. Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly non-linear compressible fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 84:73 – 107, 2010.
- [11] C. Farhat, K.G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. 195:1973 – 2001, 2006.
- [12] J. Favier, A. Revell, and A. Pinelli. A Lattice Boltzmann - Immersed Boundary method to simulate the fluid interaction with moving and slender flexible objects. *Journal of Computational Physics*, 261:145 – 161, 2014.
- [13] C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24):3247 – 3270, 2001.
- [14] A. Gravouil and A. Combescure. Multi-time-step and two-scale domain decomposition method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 58(10):1545–1569, 2003.
- [15] A. Gravouil, A. Combescure, and M. Brun. Heterogeneous asynchronous time integrators for computational structural dynamics. *International Journal for Numerical Methods in Engineering*, 102(3-4):202 – 232, 2015.
- [16] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E*, 65:046308, 2002.

- [17] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193:1 – 23, 2004.
- [18] T.J.R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.
- [19] S. Kollmannsberger, S. Geller, A. Düster, J. Tölke, C. Sorger, M. Krafczyk, and E. Rank. Fixed-grid fluidstructure interaction in two dimensions based on a partitioned lattice Boltzmann and p-FEM approach. *International Journal for Numerical Methods in Engineering*, 79(7):817 – 845, 2009.
- [20] S. Krenk. Energy conservation in newmark based time integration algorithms. 195(44):6110 – 6124.
- [21] T. Krüger, F. Varnik, and D. Raabe. Efficient and accurate simulations of deformable particles immersed in a fluid using a combined immersed boundary lattice Boltzmann finite element method. *Computers & Mathematics with Applications*, 61:3485 – 3505, 2011.
- [22] Z. Li, J. Favier, U. D’Ortona, and S. Poncet. An immersed boundary-lattice boltzmann method for single- and multi-component fluid flows. *Journal of Computational Physics*, 304:424 – 440.
- [23] Z. Li, J. Leduc, A. Combescure, and F. Leboeuf. Coupling of SPH-ALE method and finite element method for transient fluid-structure interaction. *Computers & Fluids*, 103:6 – 17, 2014.
- [24] Z. Li, J. Leduc, J. Nunez-Ramirez, A. Combescure, and J.C. Marongiu. A non-intrusive partitioned approach to couple smoothed particle hydrodynamics and finite element methods for transient fluid-structure interaction problems with large interface motion. *Computational Mechanics*, 55(4):697 – 718, 2015.

- [25] Q. Lou, Z. Guo, and B. Shi. Evaluation of outflow boundary conditions for two-phase lattice boltzmann equation. *Physical Review E*, 87(6):063301, 2013.
- [26] C. Michler, S.J. Hulshoff, E.H. van Brummelen, and R. de Borst. A monolithic approach to fluid-structure interaction. *Computers & Fluids*, 33(5 – 6):839 – 848, 2004.
- [27] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239 – 261.
- [28] N.M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85, 1959.
- [29] C.S. Peskin. The immersed boundary method. *Acta Numerica*, 11:1 – 39, 2002.
- [30] A. Pinelli, I.Z. Naqavi, U. Piomelli, and J. Favier. Immersed-boundary methods for general finite-difference and finite-volume Navier-Stokes solvers. *Journal of Computational Physics*, 229:9073 – 9091, 2010.
- [31] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: Energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3147 – 3170, 2001.
- [32] Y.H. Qian, D. D’Humières, and P. Lallemand. Lattice BGK Models for Navier-Stokes Equation. *Europhysics Letters*, 17(6):479 – 484, 1992.
- [33] A.M. Roma, C.S. Peskin, and M.J. Berger. An adaptive version of the immersed boundary method. *Journal of Computational Physics*, 153:509 – 534, 1999.
- [34] X. Shan, X.F. Yuan, and H. Chen. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *Journal of Fluid Mechanics*, 550:413 – 441, 2006.

- [35] A.K. Soti, R. Bhardwaj, and J. Sheridan. Flow-induced deformation of a flexible thin structure as manifestation of heat transfer enhancement. *International Journal of Heat and Mass Transfer*, 84:1070 – 1081, 2015.
- [36] F.B. Tian, H. Dai, H. Luo, J.F. Doyle, and B. Rousseau. Fluidstructure interaction involving large deformations: 3d simulations and applications to biological systems. *Journal of Computational Physics*, 258:451 – 469, 2014.
- [37] S. Turek and J. Hron. *Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow*, volume 53 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2006.
- [38] J. Wu and C. Shu. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *Journal of Computational Physics*, 228:1963 – 1979, 2009.
- [39] J. Wu and C. Shu. An improved immersed boundary-lattice boltzmann method for simulating three-dimensional incompressible flows. *Journal of Computational Physics*, 229(13):5022 – 5042, 2010.
- [40] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice boltzmann BGK model. *Physics of Fluids*, 9:1591 – 1598, 1997.