Volumetric Lattice Boltzmann Method for Wall Stresses of Image-based Pulsatile Flows

Xiaoyu Zhang
Indiana University-Purdue University
Joan Gomez-Paz
The City College of New York
J. M. McDonough
University of Kentucky
Mahfuzul MD Islam
Indiana University-Purdue University
Yiannis Andreopoulos
The City College of New York
Huidan Yu (whyu@iupui.edu)
Indiana University-Purdue University

Research Article

Keywords: Image-based computational fluid dynamics, volumetric lattice Boltzmann method, wallstress, pulsatile flows

Posted Date: April 9th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-384519/v1

Version of Record: A version of this preprint was published at Scientific Reports on February 1st, 2022.
See the published version at https://doi.org/10.1038/s41598-022-05269-w.

Volumetric Lattice Boltzmann Method for Wall Stresses of Image-based Pulsatile Flows

Xiaoyu Zhang[1], Joan Gomez-Paz[2], J. M. McDonough[3], Mahfuzul MD Islam[1], Yiannis Andreopoulos[2], and Huidan (Whitney) Yu[1,4*]
[1]Department of Mechanical and Energy Engineering, Indiana University-Purdue University, Indianapolis (IUPUI), IN 46202, USA
[2]Department of Mechanical Engineering, The City College of New York, NY 10031 USA
[3]Departments of Mechanical Engineering and Mathematics, University of Kentucky, Lexington KY 40506, USA
[4]Department of Vascular Surgery, Indiana University School of Medicine, Indianapolis, IN 46202, USA
*Corresponding Author. E-mail address: whyu@iupui.edu

Abstract:
Image-based computational fluid dynamics (CFD) has become a new capability for determining wall stresses of pulsatile flows. However, a computational platform that directly connects image information to pulsatile wall stresses is lacking. Prevailing methods rely on manual crafting of a hodgepodge of multidisciplinary software packages, which is usually laborious and error prone. We present a new technique to compute wall stresses in image-based pulsatile flows using the lattice Boltzmann method (LBM). The novelty includes: (1) a unique image processing to extract flow domain and local wall normality, (2) a seamless connection between image extraction and CFD, (3) an en-route calculation of strain-rate tensor, and (4) GPU acceleration (not included here). We first generalize the streaming operation in the LBM and then conduct an application study for laminar and turbulent pulsatile flows in an image-based pipe (Reynolds number: 10 to 5000). The computed pulsatile velocity and shear stress are in good agreement with Womersley solutions for laminar flows and concurrent laboratory measurements for turbulent flows. This technique is being used to study (1) the hemodynamic wall stresses in inner choroid endothelium, (2) the drag force in sand flows, and (3) effects of waste streams on ion exchange kinetics in porous media.

Keywords: Image-based computational fluid dynamics, volumetric lattice Boltzmann method, wall-stress, pulsatile flows

1. Introduction

Pulsatile flows are omnipresent in a wide range of engineering and scientific systems. Examples include pulmonary ventilating[1] and blood circulating[2] in biological flows, sediment transport in coastal flows[3], and reciprocating flow in internal combustion engines[4]. Understanding turbulence in pulsatile flows is critically important to both knowledge advancement and technological innovation but challenges have remained due to the complexity of these flows. A typical pulsatile flow is by nature four-dimensional (4-D: 3-D in space plus 1-D in time). It is composed of a positive mean and a periodically varying time-dependent component around the mean. It is inherently unsteady with successive flow acceleration and deceleration, resulting in timewise transitions from laminar to turbulent flow (turbulentization) and then back to laminar flow (relaminarization) during pulsation[5,6]. Since the seminal work of Reynolds [7], significant efforts have been made to study the transition to turbulence in pulsatile flows[8-12]. The focus has been on the exploration of criteria for laminar, transitional, and turbulent flow in pulsatile flows from different aspects including hydrodynamic stability[13] and instability[14], pulsatility[15], and turbulence dynamics[16]. All these studies were for pulsatile flows in pipes. Pulsatile flow is commonplace in the cardiovascular system, and its flow domain geometry is far more complicated than a straight pipe. The wall stresses connect the blood flow dynamics and the vessel wall response, providing pathophysiological insights for different cardiovascular diseases that are not readily measurable in medical practice. Wall-normal stress (WNS) is associated with blood pressure, leading to the deformation of cells in the vessel wall, whereas wall-shear stress (WSS) acts on the endothelium of the vessel wall through a shearing deformation (‘shearing force’). Wall stresses play a central role, not only in aneurysm

initiation, growth, and rupture[17] but also in the development of atherosclerosis[18]. With the advances in medical imaging, scientific modeling, and computational power, image-based computational fluid dynamics (ICFD) has emerged[19-25] as a new computer-aided tool for diagnostics and therapeutics of cardiovascular diseases. Based on medical imaging data, such as computed tomography angiogram (CTA) or magnetic resonance imaging (MRI), together with Doppler ultrasound sonography (DUS), ICFD has enabled noninvasive evaluation of 4-D *in vivo* vectorial velocity and pressure in the entire arterial system with fine spatial and temporal resolution. For an arbitrarily oriented wall, both WNS and WSS are vectors, projected from the total stress tensor on local normal and tangential directions of the wall, respectively, with magnitudes and directions (derivation shown later). Full characterization of the wall stresses for cardiovascular hemodynamics includes its topological, spatiotemporal, and vectorial nature[18,26], which remains challenging in ICFD due to the fact that the real human vasculature is usually irregular in geometry and orientation with curvatures and bifurcations. When the vessel is diseased with either stenosis (lumen reduction) or aneurysm (lumen enlargement), its geometry can be exceptionally complicated.

A general process of ICFD for wall stresses mainly involves four tasks: (1) *extraction* of flow domain and local normal orientation from image data, (2) *connection* between output of image processing and input of CFD, (3) *quantification* of 4-D velocity and pressure fields employing physical parameters together with initial and boundary conditions using CFD, and (4) *calculation* of wall stresses via post-processing. Either commercial software or an open-source package that directly solves ICFD from image data to wall stresses is currently unavailable. The prevailing approach for ICFD is to utilize a hodgepodge of multidisciplinary software and/or open-source packages, such as Materialise Mimics, ScanIP, BLOD3D, Avizo, and 3D VIEWNIX for image segmentation and ANSYS, COMSOL, STAR-CD, OpenFOAM, and FEFLOW for CFD. Since the packages for image processing and CFD are independent, one needs to use additional software and/or open-source packages to connect these two tasks via geometry rescaling and mesh generation. The representative software and open-source packages for such connection are T-grid, Pointwise, Mesh lab, Gmesh, and Meshfix. Although popular, such a process is usually laborious and error-prone due to the need for manual crafting of each software and/or open-source package. Meanwhile, in conventional CFD, the strain-rate tensor, also shear stress, is calculated from the 1st-order derivatives of the velocity vector field as part of the post-processing. Total stress tensor and wall stresses (WSS and WNS) are then calculated. There are two challenges in this part of the post-processing. The first is the necessity of determining the normal direction on the local wall, which is difficult when the flow domain is irregular and complicated. The second is that the velocity adjacent to the wall is small and noisy, especially when the flow is turbulent. Computing near-wall velocity requires fine resolution, which can cause a substantial computational burden. In fact, the computational cost of ICFD for pulsatile flows is inherently high and will become demanding when the flow is turbulent.

In this work, we present a unique computational platform for quantifying 4-D wall stresses in image-based pulsatile flow using volumetric lattice Boltzmann method (LBM)[27] to address the aforementioned challenges. The novelty of this computational approach consists of (1) a unique extraction of flow domain and local wall normality from image data, (2) a seamless connection between the output of image processing and the input of CFD, (3) an en-route calculation of strain-rate tensor for wall stresses, and (4) GPU parallel computing processing[28-31] (not included in this paper) to significantly mitigate the computation burden. The kinetic-based LBM has emerged for simulating a broad class of complex flows including pore-scale porous media flow[32,33], multiphase/multicomponent flows[34-38], and turbulence[36,39-43] . The main advantages of the LBM for this work are its amenability for modeling the intermolecular interactions at the two-phase interface to recover the appropriate multiphase dynamics without demanding computing cost and its suitability for scalable GPU (Graphics Processing Unit) parallelization[28,30,31,36,39-44] to achieve fast computation. Instead of solving the Navier–Stokes (NS) equations directly, the LBM

deals with a discrete mesoscopic equation coherently discretized from the Boltzmann equation[45] in phase space, which governs the time evolution of particle density distribution functions employing streaming and collision (relaxation) operations. Through the Chapman-Enskog technique[46], it can be rigorously proved[47] that the lattice Boltzmann equation (LBE) collapses to the NS equations in the incompressible limit. Being a volumetric representation of the conventional LBM[48], the volumetric LBM was specially developed for CFD in arbitrarily complicated flow domains, e.g., porous media and blood capillaries, with or without moving boundaries. We generalize the streaming operation in this work from its original formulation[48]. Aiming toward broad applications for not only biomedical flows but also parametric-designed flows in science and engineering, we do not limit the image data in DICOM (Digital Imaging and Communications in Medicine) format but deal with STL (Standard Triangle Language) data format, an industry-standard file type for 3-D domains. The DICOM is most common for medical images scanned from CTA or MRI. Techniques for DICOM image segmentation[29] and STL conversion from DICOM segmentation[49] have been developed. Whereas all modern CAD (Computer-Aided Design) software such as SolidWorks export their native file format into STL. An application study of pulsatile flows in an image-based pipe demonstrates the reliability and applicability of the computational method. We simulate Womersley (laminar) flows and compare the 4-D computed fields of the velocity, shear stress tensor, and WSS with analytical solutions. Then we simulate turbulent flows varying the base flow Reynolds number (defined later) from 535 to 4375 and compare the computed 4-D velocity and WSS with experimental data.

The remainder of the paper is organized as follows. In Section 2, we present the computational method including the volumetric LBM, the image processing, the connection of both, and the en-route calculation of WSS and WNS. An application study of pulsatile flows, both laminar and turbulent flows, in an image-based pipe is carried out in Section 3. Finally, Section 4 concludes the paper with a summary and discussion.

2. Volumetric lattice Boltzmann modeling and formulation for wall stresses in ICFD

As schematized in Fig. 1, the volumetric LBM for wall stresses in ICFD consists of three components: image acquisition and segmentation (blue), determination of volumetric parameter and wall normality

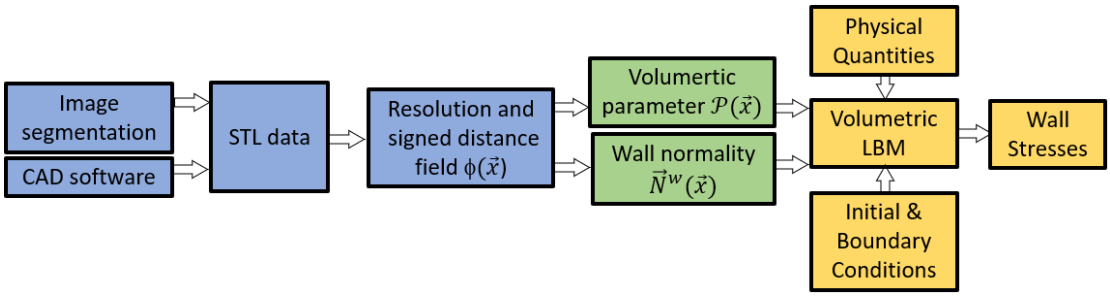


Fig. 1 Flowchart from image data to wall stresses. The volumetric parameter and wall normality (green) are the outputs of image processing (blue) and inputs of CFD using volumetric LBM (yellow), enabling seamless connection between both.

(green), and CFD using volumetric LBM for wall stresses (yellow). Image segmentation and CFD are seamlessly connected as the volumetric parameter and wall normality are the outputs of the former and the inputs of the latter. In this section, we present the modeling and formulation from three aspects: the formulation of volumetric LBEs, geometrical processing, and en-route calculation of wall stresses. It is noted that the streaming operation in the LBEs is generalized from its original formula[27].

2.1 Formulation of volumetric lattice Boltzmann method

The volumetric representation of the LBM[27] was developed specifically for treating arbitrarily oriented boundaries with or without boundary movement. In the volumetric LBM, fluid particles are uniformly distributed in lattice cells. By introducing a volumetric parameter $\mathcal{P}(\vec{x},t)$, defined as the occupation of solid volume in the cell, i.e., $\mathcal{P}(\vec{x},t) \equiv \Delta V_s(\vec{x},t)/\Delta V(\vec{x},t)$, we distinguish three types of lattice cells in the simulation domain: solid cell (pure solid occupation, $\mathcal{P}$ = 1), fluid cell (pure fluid occupation, $\mathcal{P}$ = 0), and boundary cell (partial solid and partial fluid, 0 < $\mathcal{P}$ < 1), as illustrated in Fig. 2.

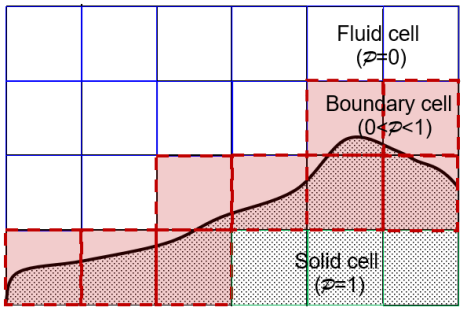


Fig. 2 Illustration of the cell-based lattice involving solid ($\mathcal{P}$ = 1, green), fluid ($\mathcal{P}$ = 0, white), and boundary (0 < $\mathcal{P}$ < 1, red) cells.

Formulation of the volumetric LBM is self-regularized to deal with the complex flow geometries through $\mathcal{P}$. It consists of three operations: (1) collision considering the momentum exchange between the moving boundary and the flow; (2) streaming accompanying a volumetric bounce-back procedure in boundary cells; and (3) migrating volumetrically moving the residual fluid particles into the flow domain when the boundary swipes over a boundary cell toward a solid cell. The volumetric LBM strictly satisfies mass conservation and can handle any irregular boundary orientation and motion with respect to the mesh. In this work, we consider fixed flow geometries only, thus $\mathcal{P}(\vec{x},t) = \mathcal{P}(\vec{x})$, which is a one-time extraction from the imaging information or CAD. Thus, the collision reverts to the conventional operation in LBM and the migrating operation is not involved.

We use a D3Q19 lattice model to present the volumetric LBEs. The 19 discrete molecular velocities are

$$\vec{e}_i = \begin{cases} (0,0,0) & & & i=0 \\ (\pm1,0,0)c, & (0,\pm1,0)c, & (0,0,\pm1)c & i=1-6 \\ (\pm1,\pm1,0)c, & (\pm1,0,\pm1)c, & (0,\pm1,\pm1)c & i=7-18 \end{cases}$$

where c ($=\frac{\delta x}{\delta t}$ with δx and δt the lattice width and time interval respectively) represents the lattice speed in lattice units. We select $c = 1$, i.e., $\delta x = \delta t = 1$, meaning that the particles stream one lattice unit per time step. The sound speed of the D3Q19 model in the lattice units is $c_s = \frac{c}{\sqrt{3}}$; and the weighting factors ω_i are $\omega_0 = \frac{1}{3}$, $\omega_{1-6} = \frac{1}{18}$, and $\omega_{7-18} = \frac{1}{36}$.

The volumetric LBM deals with the time evolution of the particle population in each lattice cell, and the volumetric LBE for the collision operation is

$$n_i(\vec{x}+\vec{e}_i\delta t,t) - n_i(\vec{x},t) = \Omega_i(\vec{x},t) \quad i = 0\sim18 \qquad (1)$$

where $n_i(\vec{x},t)$ and $\Omega_i(\vec{x},t)$ are the particle population and particle collision operator with discrete velocity $\vec{e}_i$ along the *i*-th direction, respectively, in cell $\vec{x}$ at time *t*. The simplest and most popular choice for the collision operation is the BGK (Bhatnagar, Gross, and Krook) model[50]

$$\Omega_i(\vec{x},t) = -\frac{1}{\tau}\left[n_i(\vec{x},t) - n_i^{eq}(\vec{x},t)\right] \qquad (2)$$

with $n_i^{eq}(\vec{x},t)$ the equilibrium particle population and τ the relaxation time due to fluid-particle collisions. The corresponding equilibria for incompressible flows are

$$n_i^{eq}(\vec{x},t) = N\omega_i\left\{1 + \frac{3\vec{e}_i\cdot\vec{u}}{c^2} + \frac{9(\vec{e}_i\cdot\vec{u})^2}{2c^4} - \frac{3\vec{u}\cdot\vec{u}}{2c^2}\right\} \qquad (3)$$

where $N(\vec{x},t) \equiv \sum_{i=0}^{18} n_i(\vec{x},t) \equiv \sum_{i=0}^{18} n_i^{eq}(\vec{x},t)$ and $N(\vec{x},t)\vec{u}(\vec{x},t) \equiv \sum_{i=0}^{18} \vec{e}_i n_i(\vec{x},t) \equiv \sum_{i=0}^{18} \vec{e}_i n_i^{eq}(\vec{x},t)$.

The relation between the particle density distribution function $f_i(\vec{x},t)$ in a node-based LBM and the particle population $n_i(\vec{x},t)$ in cell-based volumetric LBM is $n_i(\vec{x},t) = [1-\mathcal{P}(\vec{x})]f_i(\vec{x},t)$. As $\Delta V_f(\vec{x},t) = [1-\mathcal{P}(\vec{x})]\Delta V$, $n_i(\vec{x},t)$ collapses to $f_i(\vec{x},t)$ in fluid the cells where $\mathcal{P}=0$.

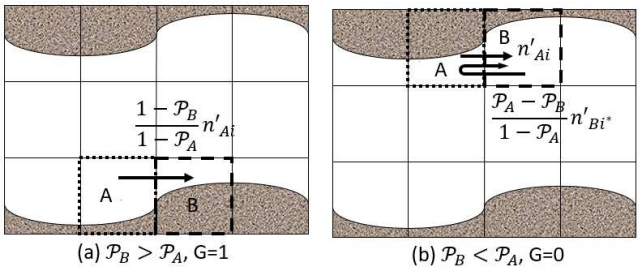
The implementation of Eq. (1) includes two operations. The first involves the collision operator, $\Omega_i(\vec{x},t)$, resulting in $n'_i(\vec{x},t)$ that is referred to as 'post-collision' particle population.

$$n'_i(\vec{x},t) = n_i(\vec{x},t) + \Omega_i(\vec{x},t) \tag{4}$$

The second operation is for streaming, which reflects the particles moving from the current cell to the neighboring cells. Since a boundary cell is partially occupied by fluid, only a partial volume fraction of fluid particles can stream to a neighboring cell in the direction of the molecular velocity. In the original paper[27], the streaming equation is formulated as follows.

$$n_i(\vec{x},t+\delta t) = [1-\mathcal{P}(\vec{x})]\cdot n'_i(\vec{x}-\vec{e}_i\delta t,t) + \mathcal{P}(\vec{x}+\vec{e}_{i*}\delta t)\cdot n'_{i*}(\vec{x},t) \tag{5}$$

where i^* is the particle velocity direction opposite to the i-th direction, i.e., $\vec{e}_{i*} = -\vec{e}_i$. In Eq. (5), the receiving of the streaming particles from the upwind neighboring cell at $\vec{x}+\vec{e}_{i*}\delta t$ depends on the fluid fraction of the current cell at $\vec{x}$, i.e., $[1-\mathcal{P}(\vec{x})]n_i(\vec{x}+\vec{e}_{i*}\delta t,t)$, plus $\mathcal{P}(\vec{x}+\vec{e}_{i*}\delta t)n'_{i*}(\vec{x},t)$ particles bounced back from the upwind neighboring cell in the opposite direction. This equation does not simultaneously consider the fluid fractions of both current and upwind cells. This is problematic when the current and upwind cells are boundary cells. Also, Eq. (5) does not apply for the scenario that both current and upwind cells are boundary cells with an equal fluid fraction. To address these deficiencies, we consider two scenarios in the streaming operation as illustrated in Fig. 3. Assume cell B, located at $\vec{x}$, is receiving particles during the streaming, and cell A is cell B's upwind neighboring cell located at $\vec{x}+\vec{e}_{i*}\delta t$. If $\mathcal{P}_B > \mathcal{P}_A$, see Fig. 3 (a), the receiving particles are purely streamed from cell A. Because the fluid fraction of cell B is smaller than that of cell A, only $\frac{1-\mathcal{P}_B}{1-\mathcal{P}_A}n'_{Ai}$ particles among the total particles in cell A in the i-th direction can stream into cell B and the remaining particles $\frac{\mathcal{P}_B-\mathcal{P}_A}{1-\mathcal{P}_A}n'_{Ai}$ will bounce back to cell A in the direction of $\vec{e}_{i*}$. Figure 3 (b) illustrates another scenario, $\mathcal{P}_B < \mathcal{P}_A$, in which the receiving particles are from two sources: n'_{Ai} (streamed from cell A) and $\frac{\mathcal{P}_A-\mathcal{P}_B}{1-\mathcal{P}_A}n'_{Bi^*}$ (bounced back from n'_{Bi^*}). We now introduce parameter G as



(a) $\mathcal{P}_B > \mathcal{P}_A$, G=1 (b) $\mathcal{P}_B < \mathcal{P}_A$, G=0

Fig. 3 Illustration of two streaming scenarios when the current cell B at $\vec{x}$ (dashed) and upwind cell A (doted) at $\vec{x}+\vec{e}_{i*}\delta t$ are both boundary cells. A parameter G is introduced to distinguish these two cases. (a) $\mathcal{P}_B > \mathcal{P}_A$, G=1. The receiving particles are streamed from the upwind cell. Because the fluid fraction of B is smaller than that of cell A, only $\frac{1-\mathcal{P}_B}{1-\mathcal{P}_A}n'_{Ai}$ particles of cell A can stream into cell B and no bounced-back particles. (b) $\mathcal{P}_B < \mathcal{P}_A$, G=0. The receiving particles consists of two parts: streamed from the upwind cell A (n'_{Ai}) and the bounced-back particles from the current cell $\frac{\mathcal{P}_A-\mathcal{P}_B}{1-\mathcal{P}_B}n'_{Bi^*}$ in the opposite direction.

$$G_i(\vec{x}) = \begin{cases} 1, & if\ \mathcal{P}(\vec{x}+\vec{e}_{i*}\delta t) < \mathcal{P}(\vec{x}), \\ 0, & if\ \mathcal{P}(\vec{x}+\vec{e}_{i*}\delta t) \geq \mathcal{P}(\vec{x}) \end{cases} \tag{6}$$

and generalize the streaming operation as

$$n_i(\vec{x}, t+\delta t) = G_i(\vec{x})\frac{1-\mathcal{P}(\vec{x})}{1-\mathcal{P}(\vec{x}+\vec{e}_{i*}\delta t)} n_i(\vec{x}+\vec{e}_{i*}\delta t, t) + [1-G_i(\vec{x})]\left[n_i(\vec{x}+\vec{e}_{i*}\delta t, t) + \frac{\mathcal{P}(\vec{x}+\vec{e}_{i*}\delta t)-\mathcal{P}(\vec{x})}{1-\mathcal{P}(\vec{x})} n_{i*}(\vec{x}, t)\right] \quad (7)$$

In the case of $\mathcal{P}_B = \mathcal{P}_A$, Eq. (7) becomes $n_i(\vec{x}, t+\delta t) = n_i(\vec{x}+\vec{e}_{i*}\delta t, t)$, which Eq. (5) cannot achieve.

The resulting density, velocity, and pressure of the flow system are obtained as follows:

$$\rho(\vec{x}, t) = \frac{\sum_{\alpha=0}^{18} n_i(\vec{x},t)}{1-\mathcal{P}(\vec{x})}, \quad (8)$$

$$\vec{u}(\vec{x}, t) = \frac{\sum_{i=0}^{18} \vec{e}_i n_i(\vec{x},t)}{\sum_{i=0}^{18} n_i(\vec{x},t)}, \quad (9)$$

and

$$p(\vec{x}, t) - p_0 = c_s^2[\rho(\vec{x}, t) - \rho_0] \quad (10)$$

where p_0 and ρ_0 (=1) are reference pressure and density in lattice units, respectively.

The volumetric LBM can handle arbitrary oriented boundary with or without motion, and it satisfies mass conservation strictly. The formulation taking into consideration of bounce-back boundary conditions is self-regulated by the volumetric parameter $\mathcal{P}(\vec{x})$ that distinguishes fluid, boundary, and solid cells. The implementation of the volumetric LBEs only occurs in fluid and boundary cells. Thus, the key to the volumetric LBM is the determination of $\mathcal{P}(\vec{x})$ for a flow domain.

2.2 Geometrical processing for volumetric parameter $\mathcal{P}(\vec{x})$ and local wall normality $\vec{N}^w(\vec{x})$

Geometrical processing is an important step in ICFD needed to connect image segmentation to CFD. Usually, it requires extra effort for geometry rescaling and mesh generation. When using the volumetric LBM as the CFD solver, seamless connection between the image information and CFD can be achieved by quantifying the volumetric parameter $\mathcal{P}(\vec{x})$ for the entire simulation domain. When CT or MRI images (usually in DICOM data format) are used, image segmentation needs to be done first to extract the 3-D flow domain, followed by the calculation of $\mathcal{P}(\vec{x})$[29,51]. In scientific and engineering applications, the 3-D flow domains are usually generated by CAD software, and STL is the standard output format for enabling 3-D printing. Since DICOM image data can be exported in STL format after the image segmentation, we present the method to calculate the $\mathcal{P}(\vec{x})$ starting from image data in STL format for a broad range of applications. The normal direction of each boundary cell, $\vec{N}^w(\vec{x})$, which is required for calculating WSS and WNS, is calculated concurrently. The geometrical process from STL data to the volumetric parameter $\mathcal{P}(\vec{x})$ and wall normality $\vec{N}^w(\vec{x})$ is seen in Fig. 1. The green part is the output of the image processing (blue part) and the input of volumetric LBM (yellow part), enabling a seamless connection between image information and CFD. The geometrical process consists of three main components described in the following subsections.

2.2.1 Level set method for signed-distance field $f(\vec{x})$ from STL data

The level set method is a conceptual framework for using level sets as a tool for numerical analysis of surfaces and shapes. One can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects[52]. A signed-distance function $\phi(\vec{x})$ determines the distance of a given point $\vec{x}$ from an interface, with the sign determined by whether $\vec{x}$ is inside or outside the interface. It is a special level set, that can handle curves and surfaces with topological changes easily. When a flow domain is closed and bounded by an interface Γ with inward normal $\vec{N}^w$, as shown in Fig. 4, the signed distance field $\phi(\vec{x})$ represents the geometric space using a level set of

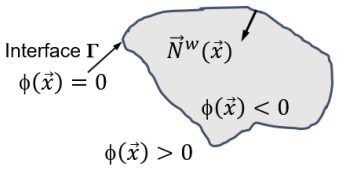


Fig. 4 Signed distance field $\phi(\vec{x})$: $\phi(\vec{x}) = 0$ locates on the interface Γ with inward normal $\vec{N}^w(\vec{x})$. $\phi(\vec{x}) < 0$ and $\phi(\vec{x}) > 0$ mean inside and outside of surface Γ.

$$\phi(\vec{x}) = \begin{cases} > 0, & \vec{x} \in \Gamma \\ = 0, & \vec{x}\ on\ \Gamma \\ < 0, & \vec{x} \notin \Gamma \end{cases} \quad (9)$$

with

$$|\nabla\phi| = 1. \quad (10)$$

The STL data describes a raw, unstructured triangulated surface by the unit normal and vertices (ordered by the right-hand rule) of the triangles using a 3-D Cartesian coordinate system. This file format is supported by many other software packages, e.g., MATLAB. We use existing open-source packages of *READ_stl.m*, *VOXELISE.m*, and *ac-reinit.m* to read the STL file, generate a 3-D uniform mesh with the specified resolution, and calculate the signed distance field $\phi(\vec{x})$ in MATLAB.

2.2.2 Volumetric parameter $\mathcal{P}(\vec{x})$ and wall-normal direction $\vec{N}^w(\vec{x})$ from signed distance field $\phi(\vec{x})$

For illustration, we use a 2-D lattice, in Fig. 5, to describe how to determine the $\mathcal{P}$ value for a cell from the signed distance field $\phi(\vec{x})$. Given the interface, which is the boundary of the flow domain, the sign ϕ at each node of the lattice indicates the area, either fluid (white) or solid (orange), and the node location based on the sign convention ϕ. If the ϕs at the four vertices of a cell are all negative, the cell is inside the fluid domain, and $\mathcal{P} = 0$; whereas, if all four ϕs are positive, the cell is in the solid domain and $\mathcal{P} = 1$, see green and blue cells in Fig. 5, respectively. Otherwise, the cell is a boundary cell (red) occupied by partial solid and partial fluid.

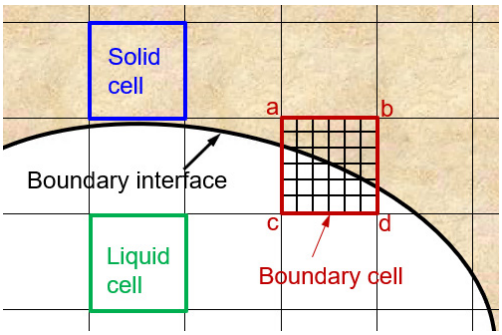


Fig. 5: Identification of fluid cell, solid cell, and boundary cell based on the ϕ signs at four vertexes of each cell. All negative ϕs mean a fluid cell (green, $\mathcal{P} = 0$), all positive ϕs means a solid cell (blue, $\mathcal{P} = 1$), and partial negative (vertexes c and d) partial positive (vertexes a and b) indicates a boundary cell (red, $0 < \mathcal{P} < 1$).

For a boundary cell, we use the following steps to calculate the $\mathcal{P}$ value.

- Uniformly divide the boundary cell into q^2 sub-cells. Each cell is represented by the square center $\vec{x}_s^j$ with a solid volume V_s^j, $j = 1, \dots, q^2$.
- Calculate the ϕ value at its $\vec{x}_s^j$ using the known ϕ values at 4 vertices of the cell through interpolation for each sub-cell. If $f(\vec{x}_s^j) > 0$, set $V_s^j = 1$; otherwise, set $V_s^j = 0$.
- Obtain the total solid volume through a summation of the solid volume of all the q^2 sub-cells: $V_s = \sum_{j=1}^{q^2} V_s^j(\vec{x}_s^j)$.
- Obtain the solid volume fraction, i.e., the $\mathcal{P}$ value, as $\mathcal{P} = V_s/q^2$.

Note that more sub-cells, equivalent to larger q, result in a more accurate $\mathcal{P}$ value, but more computation will be needed to calculate the volume fraction. Thus, q should be appropriately chosen to balance between accuracy and computational cost.

The inward normal direction at the volumetric center of a boundary cell can be easily determined. The normal direction to the boundary wall is needed to calculate the WSS and WNS, see Eqs. (17) and (18) below. For a 3-D flow domain, a boundary cell is divided into q^3 sub-cells, each of which is identified by its geometrical center $\vec{x}_s^j$ and its volume $V_s^j(\vec{x}_s^j)$. Notice that $V_s^j(\vec{x}_s^j)$ is either 1 if positive $\phi(\vec{x}_s^j)$ or 0 if negative $\phi(\vec{x}_s^j)$. The volumetric center of the fluid in a boundary cell can be determined by

$$\vec{x}_b = \frac{\sum_{j=1}^{q^3}[1-V_s^j(\vec{x}_s^j)]\vec{x}_s^j}{\sum_{j=1}^{q^3}[1-V_s^j(\vec{x}_s^j)]}. \quad (11)$$

Then based on the volumetric center $\vec{x}_b$ and the distance functions ϕs at the eight vertices of the cubic cell, linear interpolation results in the direct distance of the volumetric center to the boundary surface. Meanwhile, the normal direction of the local interface in each boundary cell can be obtained from the gradient of the distance function at the eight vertices, through interpolation, as

$$\vec{N}^w(\vec{x}) = \frac{\nabla\phi(\vec{x}_b)}{|\nabla\phi(\vec{x}_b)|} \quad . \tag{12}$$

2.3 En-route calculation of wall stresses

In the LBM, the strain-rate tensor, which is defined as $S_{\alpha\beta} \equiv \frac{1}{2}\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right)$, can be calculated from the non-equilibrium distribution function as [53]

$$S_{\alpha\beta} = -\frac{1}{2\rho\tau c_s^2}\sum_i e_{i\alpha}e_{i\beta}\left(f_i - f_i^{eq}\right) \quad . \tag{13}$$

It has been demonstrated that Eq. (13) is more robust than the conventional finite difference method (FDM) to calculate the strain rate from the velocity field[40].

In volumetric LBM, Eq. (13) becomes

$$S_{\alpha\beta} = -\frac{1}{2N\tau c_s^2}\sum_i e_{i\alpha}e_{i\beta}\left(n_i - n_i^{eq}\right) \quad . \tag{14}$$

Observing that shear stress tensor $\sigma_{\alpha\beta} = 2\mu S_{\alpha\beta}$, pressure $p = c_s^2\frac{N}{1-\mathcal{P}}$, and the viscosity $\mu = c_s^2\left(\tau - \frac{1}{2}\right)$, we can formulate the total stress tensor as

$$T_{\alpha\beta} = -\frac{Nc_s^2}{(1-\mathcal{P})}\delta_{\alpha\beta} + (2\tau - 1)c_s^2 S_{\alpha\beta} \quad . \tag{15}$$

where $\delta_{\alpha\beta}$ is the Kronecker unit tensor.

The Cauchy formula gives the overall stress on the wall with a normal vector $\vec{N}^w$:

$$T_\alpha^{(W)} = T_{\alpha\beta}N_\beta^w \quad . \tag{16}$$

in which Einstein summation convention with index notation has been used. Then its projection onto the normal direction yields the WNS

$$T_\alpha^{(WNS)} = \left(N_\beta^w T_{\gamma\beta} N_\gamma^w\right) N_\alpha^w. \tag{17}$$

The WSS is then computed as the difference between the overall stress and its projection onto the normal:

$$T_\alpha^{(WSS)} = T_{\alpha\beta}N_\beta^w - (n_\beta T_{\gamma\beta} n_\gamma)n_\alpha \quad . \tag{18}$$

With the determination of the local wall normality $\vec{N}^w$ in Eq. (12), both WSS and WNS (when needed) can be obtained en-route during the volumetric LBM implementation.

3. Application study of oscillating flows in a 3-D pipe

We now apply the volumetric LBM to simulate pulsatile flows in an image-based 3-D pipe, focusing on demonstrating the reliability and applicability of the computational platform by comparing the computational results with analytical and experimental results. Several applications in more realistic flow systems are being carried, which will be mentioned later.

3.1 Computation set-up and convergence checks

A 3-D rigid, right-circular straight pipe is generated in Solidworks in STL format with a length L=19.05 mm and a radius R=9.525 mm. The density and kinematic viscosity of the fluid is 1.0e3 kg/m^3 and 3.004e-6 m/s^2, respectively. The pulsatile flow is driven by a pressure gradient, $P = \frac{p_{in} - p_{out}}{L}$. A no-slip boundary condition is applied on the pipe wall, realized by the bounce-back boundary condition in the LBM. It is noted that in volumetric LBMs, the bounce-back boundary condition is included in the streaming operation, see Eqs. (6) and (7), regulated by the volumetric parameter $\mathcal{P}(\vec{x})$. The pressure gradient is introduced as a body force in the volumetric LBM implementation with periodic boundary conditions at the inlet and outlet of the pipe. The initial conditions are constant pressure and zero velocity in the entire flow domain. The volumetric parameter $\mathcal{P}(\vec{x})$ of each lattice cell and the normal direction of each boundary cell $\vec{N}^w(\vec{x})$ are calculated based on the STL file by our in-house MATLAB code.

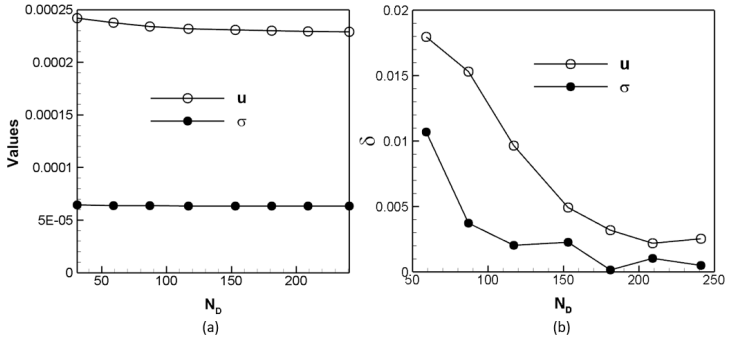


Fig. 6 Convergence checks for an appropriate spatial resolution: (a) Spatial mean velocity u (empty circle) and shear stress σ (solid circle) and (b) relative differences of u and σ between two successive resolutions (δ) vs. the cell number along the diameter (N_D). The spatial resolution is represented by N_D.

We first perform a grid-function convergence check to determine the appropriate spatial resolution. A uniform mesh in the volumetric LBM is used with the cell number, N_D, across the pipe diameter to represent the spatial resolution. We used 7 resolutions from 31 to 241 and check the dependence of interested quantities on N_D. Figure 6 (a) shows the convergence of spatial mean downstream velocity u and shear stress σ on a cross-section. It is seen that each of them becomes flat as N_D increases. The quantitative measurement of the convergence is found in Fig. 6(b), in which the convergence rate δ is calculated as the relative difference of the quantity of interest—velocity or shear stress—between two successive spatial resolutions. With the goal of $\delta < 0.5\%$ for both velocity and shear stress, we select the spatial resolution in the range of 150 to 241, depending on the Reynolds number of the flow, in what follows.

In LBM, temporal resolution Δt is determined by conversion between a physical unit and a corresponding lattice unit for a characteristic quantity such as the speed of sound. Thus, there is no need to check the convergence of temporal resolution.

3.2 Validation and verification

To address the validity of the computational platform, we simulate both Womersley flow (with the analytical solution for the downstream velocity) and turbulent pulsatile flows (with experimental measurements for the downstream velocity). The differences between the computational results and the corresponding analytical solutions and experimental data are quantitatively assessed.

3.2.1 Computational results vs. analytical solutions of Womersley flow

We use the analytical solution of Womersley flow[54] to verify the computational results. A Womersley flow, driven by a pulsatile pressure gradient, $P = P_s + P_o e^{i\omega t}$, has the following analytical solutions:

$$u(\mathrm{r},t) = \frac{P_s R^2}{4\mu}\left(1-\frac{r^2}{R^2}\right) + \mathrm{Re}\left\{\frac{P_o R^2}{i\mu\alpha^2}\left[1-\frac{J_0(\frac{\alpha r}{R} i^{\frac{3}{2}})}{J_0(\alpha i^{\frac{3}{2}})}\right] e^{i\omega t}\right\} \quad (19)$$

where u is the downstream velocity, $J_0(\cdot)$ is the Bessel function of the first kind and order zero, $Re(\cdot)$ is the real part of a complex function, $\alpha(=R\sqrt{\frac{\omega}{n}})$ is the Womersley number, and $r(=\sqrt{x^2+y^2})$ is the distance to the center of a cross-section. The strain-rate, $S=\frac{\partial u}{\partial r}$, shear stress $\sigma = 2\mu S$, and WSS $\sigma_w = \sigma|_{wall}$ can be calculated from Eq. (19). Besides the physical values introduced in Section 3.1, we use $P_s = 0.3\ Pa$, $P_o = 1.8\ Pa$, and $\omega = 1.57\ s^{-1}$ to simulate the Womersley flow. The Womersley number α and the base flow Reynolds number $Re_0 = \frac{2Ru_0}{n}$, with u_0 the spatial mean velocity and $\nu(=\frac{\mu}{\rho})$ the kinematic viscosity, of the flow, are 6.89 and 9.4, respectively. Figure 7 compares the simulation results (symbols) with the analytical solutions (lines) at 8 representative time instants in a pulsation for the downstream velocity u (a) and shear stress σ (b) and demonstrates good agreement between them. More quantitative comparisons are presented in Table 1 with the relative difference between the numerical

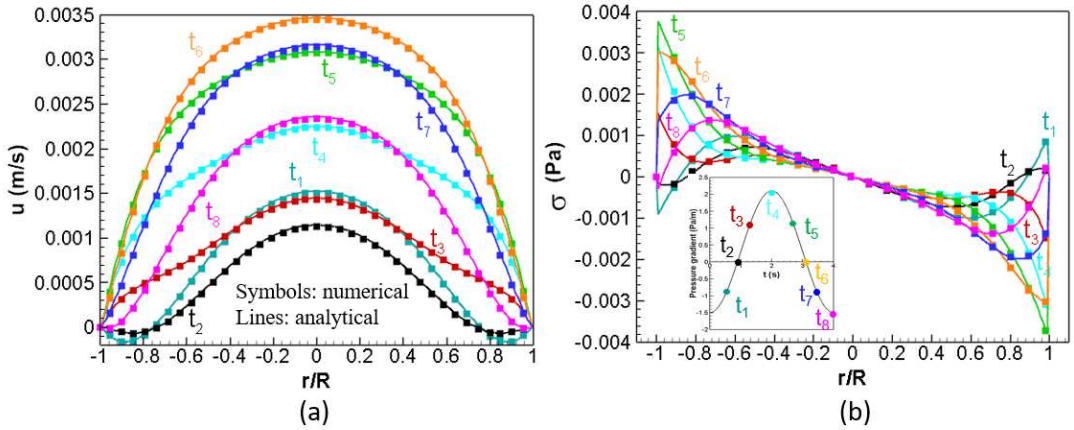


Fig. 7 Comparisons of numerical results (symbols) and analytical solutions (lines) at 8 representative time instants, shown in the embedded plot, in a pulsation for velocity u (a) and shear stress σ (b), respectively. The Womersley number α and the base flow Reynolds number Re_0 of the flow are 6.89 and 9.4, respectively.

Table 1. Relative errors calculated by $\sqrt{\frac{\sum_{i,j}(f_{num}-f_{ana})^2}{\sum_{i,j} f_{ana}^2}}$ where f is for u and σ_w on a cross-section of the pipe at the same representative time instants as embedded in Fig. 7(b). The average relative errors at the 8 time points are 1.53% for velocity and 3.98% for WSS. The analytical solution of σ_w at t_8 tends to zero thus the relative error is not available.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
u	2.42%	2.38	1.12%	1.06%	1.09%	1.15%	1.29%	1.68%
σ_w	5.16%	6.57%	4.30%	2.96%	2.42%	2.36%	4.15%	N/A

results and analytical solutions, calculated by $\sqrt{\frac{\sum_{i,j}(f_{num}-f_{ana})^2}{\sum_{i,j} f_{ana}^2}}$ where f is for u and σ on a cross-section of the pipe at the same representative time instants as embedded in Fig. 7(b). Besides, we compared the strain rate (not shown) calculated from en-route VLBM implementation and post-processing of velocity field using FDM. The VLBM is more accurate than the 3-point FDM but nearly equal accurate between VLBM and 5-point FDM.

3.2.2 Computational results vs. experimental measurements for turbulent pulsatile flows

We now validate the reliability and accuracy of the numerical simulation for more realistic pulsatile flows, relatively high base flow Reynolds number Re_0, by comparing velocity and WSS waveforms with laboratory measurements[55]. Time-resolved particle image velocimetry (PIV) techniques were used to acquire two components of velocity vector data on the plane of laser illumination that included the longitudinal axis of an acrylic pipe. The optical system consisted of a single frequency continuous-wave AR+ laser, model Spectra Physics Millennia Vs of 5.5W with continuous output power at 532 nm. A fast frame-rate CMOS camera, Vision Research v710 Phantom, was used to acquire flow images with a 1200 x 800 pixel resolution. In each experiment, 20,000 images were acquired with a rate of 1 kHz. Hollow glass particles of 9-13μm diameter (type borosilicate glass spheres LaVision 110P8) were used as tracers to visualize the flow. The PIV images were processed using an in-house modified code based on the open-source PIVLab software for MATLAB platforms. This code is a multi-pass PIV FFT (fast Fourier transform)-correlation based algorithm. The initial interrogation size was 128 x 128 pixels. It is reduced to a final window size of 16 x 16 pixels during three iterations in order to improve the signal-to-noise ratio. Five different experiments were carried out with Re_0 in the range 535 to 4825 and α from 11.91 to 23.82.

We simulated all the five flows along the downstream region from location 183 to 184.55 in the pipe. The diameter of the pipe and the fluid properties, i.e., density and viscosity, are identical to the experiment. In the simulating region, velocity has been measured at 61 downstream locations. A velocity profile at each location contains velocity information at 39 spatial locations along the pipe diameter and 400 temporal samples during a pulsatile period, i.e., 0.25Hz. We use the velocity profiles at downstream locations at 183 and 184.55 as the inflow and outflow boundary conditions, respectively. The spatial-temporal velocity information at each location is introduced in the D3Q19 lattice cell model on the corresponding cross-section via interpolation. The comparisons of (a) velocity waves at the center (r/R=0), middle location (r/R=0.5), and near the boundary (r/R=0.98) and (b) WSS wave between numerical simulation (lines) and experimental measurement (symbols) for the case of $Re_0 = 4735$ and $\alpha = 11.91$ are shown in Fig. 8. The

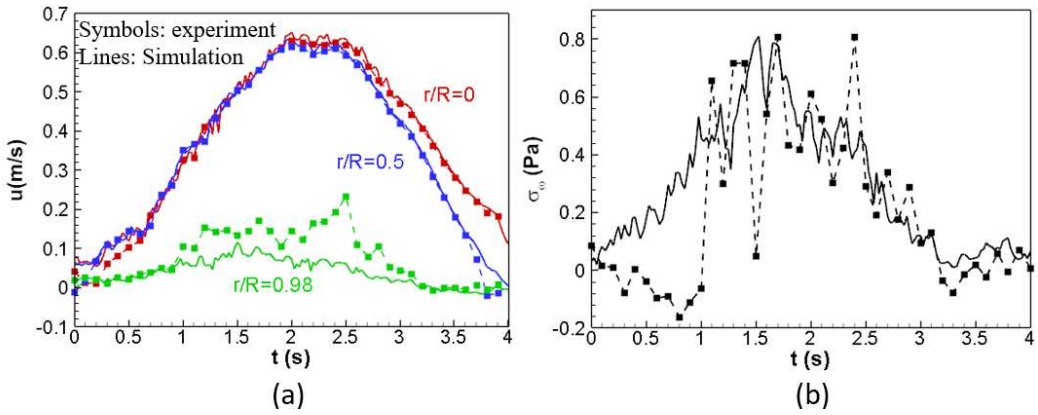


Fig. 8 Comparisons of (a) velocity waves at the center ($r/R = 0$), middle location ($r/R = 0.5$), and near boundary ($r/R = 0.98$) and (b) WSS wave between numerical simulation (lines) and experimental measurement (symbols) with $Re_0 = 4735$ and $\alpha = 11.91$.

WSS representing the experimental results in Fig. 8 (b) was calculated from the experimentally measured velocity fields via a 5-point forward FDM. Figure 8(a) shows that the simulated velocity waves in the inner pipe at $r/R = 0$ and 0.5, agree with the experimental measurements very well. Whereas at the inner wall of the pipe ($r/R = 0.98$), both velocity (Fig. 8(a)) and WSS (Fig. 8(b)) waves from simulation still agree with the experimental measurements, but the simulation results are consistently less noisy than those from experiments.

4. Summary and discussion

We have presented a unique computational method to quantify 4-D wall stresses of image-based pulsatile flows using the volumetric LBM. With the introduction of a volumetric parameter to distinguish solid, liquid, and boundary cells, the volumetric LBM is well-suited for dealing with complicated flow domains. The bounce-back boundary condition has been formulated in the streaming operation, thus avoiding complications of arbitrary boundaries. In this work, we have generalized the equation for the streaming operation from its original formation. The original streaming equation, Eq. (5), only accounts for streaming from a boundary cell to a fluid cell, which is problematic when irregular boundaries are dealt with. We generalized the streaming operation from one boundary cell to another boundary cell, Eq. (6), by introducing a parameter G, see Eq. (7), to distinguish two streaming scenarios. Starting from the provided image information, generally, in STL data format, we first construct the signed distance field based on the concept of level-set methods and directly calculate the volumetric parameter, $\mathcal{P}(\vec{x})$, and the local wall normality, $\vec{N}^w(\vec{x})$. The $\mathcal{P}(\vec{x})$ and $\vec{N}^w(\vec{x})$, together with inlet/outlet boundary conditions and initial conditions are directly fed to the volumetric LBM, resulting in a seamless connection between the image processing and CFD. In the volumetric LBM, the strain rate tensor can be calculated from the non-equilibrium particle population; therefore, the wall stresses can be obtained en-route during the volumetric LBM implementation. These features, together with GPU parallel computing, dramatically reduce resource requirements for image-based computation and ease the computational burden for pulsatile flows.

We applied the computational method to solve pulsatile flows in an image-based pipe. We first simulated a Womersley (laminar) flow with Womersley number α and the base flow Reynolds number Re_0 of 6.89 and 9.4, respectively. The computational results were in good agreement with analytical solutions, with average relative errors around 1.52% and 3.98% for mean downstream velocity and mean WSS, respectively. We found that the volumetric LBM en-route calculation of WSS is more accurate than the 3-point centered FDMs. We further simulated turbulent flows by varying Re_0 from 535 to 4375. The computational results of velocity and WSS are again in good agreement with the experimental measurements.

Although static boundaries have been the focus in this work, the method can be easily generalized for moving boundaries as the next step. The original volumetric LBM is derived for moving boundaries, and we have herein indicated how this might be applied. The formulation of volumetric LBEs consists of three parts: (1) collision taking into account the momentum exchange between the moving boundary and the flow; (2) streaming accompanying a volumetric bounce-back procedure in boundary cells; and (3) boundary-induced volumetric fluid migration moving the residual fluid particles into the flow domain when the boundary passes over a boundary cell toward a solid cell, and mass conservation is guaranteed in volumetric LBM (no need to solve a pressure Poisson equation as in NS formulations). If time-varying image information is available, the presented computational method can be generalized to solve pulsatile flow problems in image-based moving boundaries.

Beyond this work, we are actively conducting three research projects using the presented computational platform. They are (1) the hemodynamic WSS in the inner choroid prime's endothelium for complement damage, (2) the drag force on particles in sand flow, and (3) effects of waste streams on ion exchange kinetics in the porous structure, all of which will employ the techniques presented herein.

Data availability: The datasets generated in the current study are available from the corresponding authors upon appropriate request for nonprofit use.

Acknowledgment: The research is supported by NSF through grant CBET 1803845 and monitored by Dr. R. Joslin. This work used the Extreme Science and Engineering Discovery Environment (XSEDE),

which is supported by National Science Foundation Grant No. ACI-1548562. The corresponding author would like to also acknowledge the IUPUI University Fellowship and IUPUI MEE Graduate Fellowship.

References

[1] K. A. Powers and A. S. Dhamoon, "Physiology, Pulmonary, Ventilation, and Perfusion," 2019.
[2] J. C. Lasheras, "The biomechanics of arterial aneurysms," *Annual Review of Fluid Mechanics,* vol. 39, pp. 293-319, 2007.
[3] L. C. Van Rijn, *Principles of sediment transport in rivers, estuaries and coastal seas* vol. 1006: Aqua publications Amsterdam, 1993.
[4] B. Semlitsch, Y. Wang, and M. Mihăescu, "Flow effects due to pulsation in an internal combustion engine exhaust port," *Energy conversion management,* vol. 86, pp. 520-536, 2014.
[5] D. Winter and R. Nerem, "Turbulence in pulsatile flows," *Annals of biomedical engineering,* vol. 12, pp. 357-369, 1984.
[6] M. Y. Gundogdu, M. O. Carpinlioglu, and T. Engineering, "Present state of art on pulsatile flow theory: Part 1: Laminar and transitional flow regimes," *JSME International Journal Series B Fluids,* vol. 42, pp. 384-397, 1999.
[7] O. Reynolds, "An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels," *Phil. Trans. R. Soc. Lond.,* vol. 174, pp. 935-982, 1883.
[8] P. Merkli and H. Thomann, "Transition to turbulence in oscillating pipe flow," *Journal of Fluid Mechanics,* vol. 68, pp. 567-576, 1975.
[9] M. OHMI, M. IGUCHI, and I. URAHATA, "Transition to turbulence in a pulsatile pipe flow part 1, waveforms and distribution of pulsatile velocities near transition region," *J Bulletin of JSME,* vol. 25, pp. 182-189, 1982.
[10] M. IGUCHI and M. OHMI, "Transition to turbulence in a pulsatile pipe flow: Part 2, characteristics of reversing flow accompanied by relaminarization," *J Bulletin of JSME,* vol. 25, pp. 1529-1536, 1982.
[11] M. Avila, A. P. Willis, and B. Hof, "On the transient nature of localized pipe flow turbulence," *J. Fluid Mech.,* vol. 646, pp. 127-136, 2010.
[12] D. Moxey and D. Barkley, "Distinct large-scale turbulent-laminar states in transitional pipe flow," *Proceedings of the National Academy of Sciences,* vol. 107, pp. 8091-8096, 2010.
[13] C. Thomas, A. P. Bassom, P. Blennerhassett, and C. Davies, "The linear stability of oscillatory Poiseuille flow in channels and pipes," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 467, pp. 2643-2662, 2011.
[14] D. Xu, A. Varshney, X. Ma, B. Song, M. Riedl, M. Avila*, et al.*, "Nonlinear hydrodynamic instability and turbulence in the pulsatile flow," *Proceedings of the National Academy of Sciences,* vol. 117, pp. 11233-11239, 2020.
[15] D. Xu and M. Avila, "The effect of pulsation frequency on transition in pulsatile pipe flow," *Journal of Fluid Mechanics,* vol. 857, pp. 937-951, 2018.
[16] Z. Cheng, T. Jelly, S. Illingworth, I. Marusic, and A. Ooi, "Forcing frequency effects on turbulence dynamics in pulsatile pipe flow," *International Journal of Heat and Fluid Flow,* vol. 82, p. 108538, 2020.
[17] M. R. Diagbouga, S. Morel, P. Bijlenga, and B. R. Kwak, "Role of hemodynamics in initiation/growth of intracranial aneurysms," *European Journal of clinical investigation,* vol. 48, p. e12992, 2018.
[18] F. Gijsen, Y. Katagiri, P. Barlis, C. Bourantas, C. Collet, U. Coskun*, et al.*, "Expert recommendations on the assessment of wall shear stress in human coronary arteries: existing methodologies, technical considerations, and clinical applications," *European Heart Journal,* vol. 40, pp. 3421-3433, 2019.
[19] C. A. Taylor and M. T. Draney, "Experimental and computational methods in cardiovascular fluid mechanics," *Annu. Rev. Fluid Mech.,* vol. 36, pp. 197-231, 2004.

[20] D. J. Withey and Z. J. Koles, "A review of medical image segmentation: methods and available software," *International Journal of Bioelectromagnetism,* vol. 10, pp. 125-148, 2008.
[21] C. A. Taylor and C. Figueroa, "Patient-specific modeling of cardiovascular mechanics," *Annual Review of Biomedical Engineering,* vol. 11, pp. 109-134, 2009.
[22] Y. Shi, P. Lawford, and R. Hose, "Review of zero-D and 1-D models of blood flow in the cardiovascular system," *Biomedical Engineering Online,* vol. 10, p. 33, 2011.
[23] J. M. Zhang, L. Zhong, B. Su, M. Wan, J. S. Yap, J. P. Tham*, et al.*, "Perspective on CFD studies of coronary artery disease lesions and hemodynamics: A review," *International Journal for Numerical Methods in Biomedical Engineering,* vol. 30, pp. 659-680, 2014.
[24] A. L. Marsden and M. Esmaily-Moghadam, "Multiscale Modeling of Cardiovascular Flows for Clinical Decision Support," *Applied Mechanics Reviews,* vol. 67, p. 030804, 2015.
[25] P. D. Morris, A. Narracott, H. von Tengg-Kobligk, D. A. S. Soto, S. Hsiao, A. Lungu*, et al.*, "Computational fluid dynamics modelling in cardiovascular medicine," *Heart,* vol. 102, pp. 18-28, 2016.
[26] A. Arzani and S. C. Shadden, "Wall shear stress fixed points in cardiovascular fluid mechanics," *Journal of Biomechanics,* vol. 73, pp. 145-152, 2018.
[27] H. Yu, X. Chen, Z. Wang, D. Deep, E. Lima, Y. Zhao*, et al.*, "Mass-conserved volumetric lattice Boltzmann method for complex flows with willfully moving boundaries," *Physical Review E,* vol. 89, p. 063304, 2014.
[28] S. An, H. Yu, and J. Yao, "GPU-accelerated Volumetric Lattice Boltzmann Method for Porous Media Flow," *J. Petro. Sci. Eng.,* vol. 156, pp. 546-552, 2017.
[29] S. An, H. Yu, Z. Wang, R. Chen, B. Kapadia, and J. Yao, "Unified Mesoscopic Modeling and GPU-accelerated Computational Method for Image-based Pore-scale Porous Media Flows," *Int. J. of Heat Mass Trans.,* vol. 115, pp. 1192-1202, 2017.
[30] Z. Wang, Y. Zhao, A. P. Sawchuck, M. C. Dalsing, and H. W. Yu, "GPU acceleration of Volumetric Lattice Boltzmann Method for patient-specific computational hemodynamics," *Computers & Fluids,* vol. 115, pp. 192-200, 2015.
[31] H. W. Yu, R. Chen, H. Wang, Z. Yuan, Y. Zhao, Y. An*, et al.*, "GPU accelerated lattice Boltzmann simulation for rotational turbulence," *Computers & Mathematics with Applications,* vol. 67, pp. 445-451, 2014.
[32] S. An, Y. Zhan, J. Yao, H. W. Yu, and V. Niasar, "A greyscale volumetric lattice Boltzmann method for upscaling pore-scale two-phase flow," *Advances in Water Resources,* vol. 144, p. 103711, 2020.
[33] R. Chen, J. Shao, Y. Zheng, H. Yu, and Y. Xu, "Lattice Boltzmann simulation for complex flow in a solar wall," *Communications in Theoretical Physics,* vol. 59, pp. 370 - 374, 2013.
[34] R. Chen, H. W. Yu, J. Zeng, and L. Zhu, "General power-law temporal scaling for unequal-size microbubble coalescence," *Physical Review E,* vol. 101, p. 023106, 2020.
[35] J. Z. Rou Chen, Huidan(Whitney) Yu, "Mechanism of damped oscillation in microbubble coalescence," *Computers & Fluids,* vol. 183, 2019.
[36] R. Chen, W. Yu, Y. Xu, and L. Zhu, "Scalings of inverse energy transfer and energy decay in 3-D decaying isotropic turbulence with non-rotating or rotating frame of reference," *Journal of Applied and Computational Mechanics,* vol. 5, pp. 639-646, 2019.
[37] R. Chen, H. Yu, and L. Zhu, "Effects of initial conditions on the coalescence of micro-bubbles," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* vol. 232, pp. 457-465, 2018.
[38] R. Chen, H. Yu, L. Zhu, L. Taehun, and R. Patil, "Spatial and Temporal Scaling of Unequal Microbubble Coalescence," *The AIChE Journal,* vol. 63, pp. 1441-1450, 2017.
[39] H. Yu, N. Li, and R. E. Ecke, "Scaling in laminar natural convection in laterally heated cavities: Is turbulence essential in the classical scaling of heat transfer?," *Physical Review E,* vol. 76, p. 026303, 2007.

[40] H. Yu, S. S. Girimaji, and L. S. Luo, "DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method," *Journal of Computational Physics,* vol. 209, pp. 599-616, 2005.
[41] H. Yu, S. S. Girimaji, and L. S. Luo, "Lattice Boltzmann simulations of decaying homogeneous isotropic turbulence," *Physical Review E,* vol. 71, p. 016708, 2005.
[42] H. Yu and S. S. Girimaji, "Near-field turbulent simulations of rectangular jets using lattice Boltzmann method," *Physics of Fluids,* vol. 17, p. 125106, 2005.
[43] H. Yu, L. S. Luo, and S. S. Girimaji, "SCALAR MIXING AND CHEMICAL REACTION SIMULATIONS USING LATTICE BOLTZMANN METHOD," *International Journal of Computational Engineering Science,* vol. 03, pp. 73-87, 2002.
[44] H. Yu, R. Chen, H. Wang, Z. Yuan, Y. Zhao, Y. An*, et al.*, "GPU accelerated lattice Boltzmann simulation for rotational turbulence," *Computers & Mathematics with Applications,* vol. 67, pp. 445-451, 2014.
[45] S. Harris, "An introduction to the theory of the Boltzmann equation," *Holt Rinehart and Winston INC,* 1970.
[46] S. Chapman, T. G. Cowling, and D. Burnett, *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*: Cambridge university press, 1990.
[47] X. He and L. Luo, "A priori derivation of the lattice Boltzmann equation," *Physical Review E,* vol. 55, pp. R6833-R6336, 1997.
[48] S. Chen and G. D. Doolen, "Lattice Boltzmann method for fluid flows," *Annual Review of Fluid Mechanics,* vol. 30, pp. 329-364, 1998.
[49] T. Kamio, M. Suzuki, R. Asaumi, and T. Kawai, "DICOM segmentation and STL creation for 3D Printing: A Process and Software Package Comparison for Osseous Anatomy," 2020.
[50] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems," *Physical Review,* vol. 94, pp. 511-525, 1954.
[51] H. Yu, Y. Zhao, and C. Lin, "Unified Computational Method and System for in vivo Patient-Specific Hemodynamics," PCT/US2015/056942,US Patent Application Publication, Pub. No.: US 2017-0337327 A1, Pub. Date: Nov. 23, 2017.
[52] S. Osher, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.,* vol. 79, pp. 12-49, 1988.
[53] B. Stahl, B. Chopard, J. J. C. Latt, and fluids, "Measurements of wall shear stress with the lattice Boltzmann method and staircase approximation of boundaries," vol. 39, pp. 1625-1633, 2010.
[54] J. R. Womersley, "Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known," *J. Physiol.,* vol. 127, p. 553, 1955.
[55] J. Gomez, Yu, H., Andreopoulos, Y., "The role of flow reversals in transition to turbulence and relaminarization of pulsatile flows," *Journal of Fluid dynamics,* vol. in press., 2021.

Figures

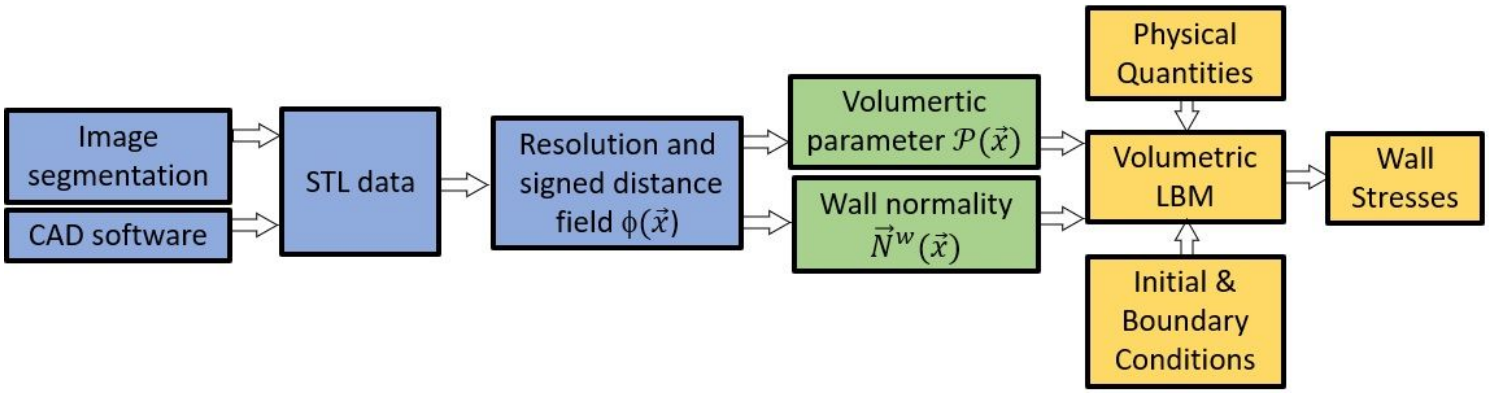


Figure 1

(see Manuscript file for full figure caption)

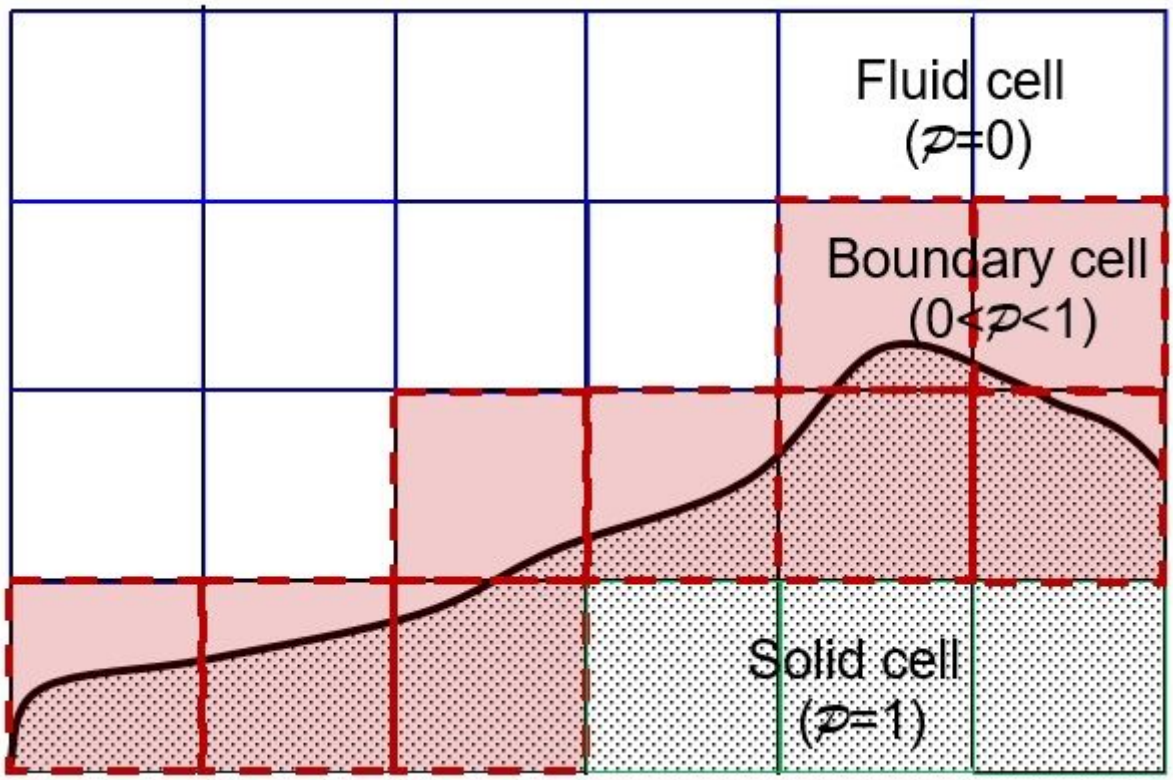


Figure 2

(see Manuscript file for full figure caption)

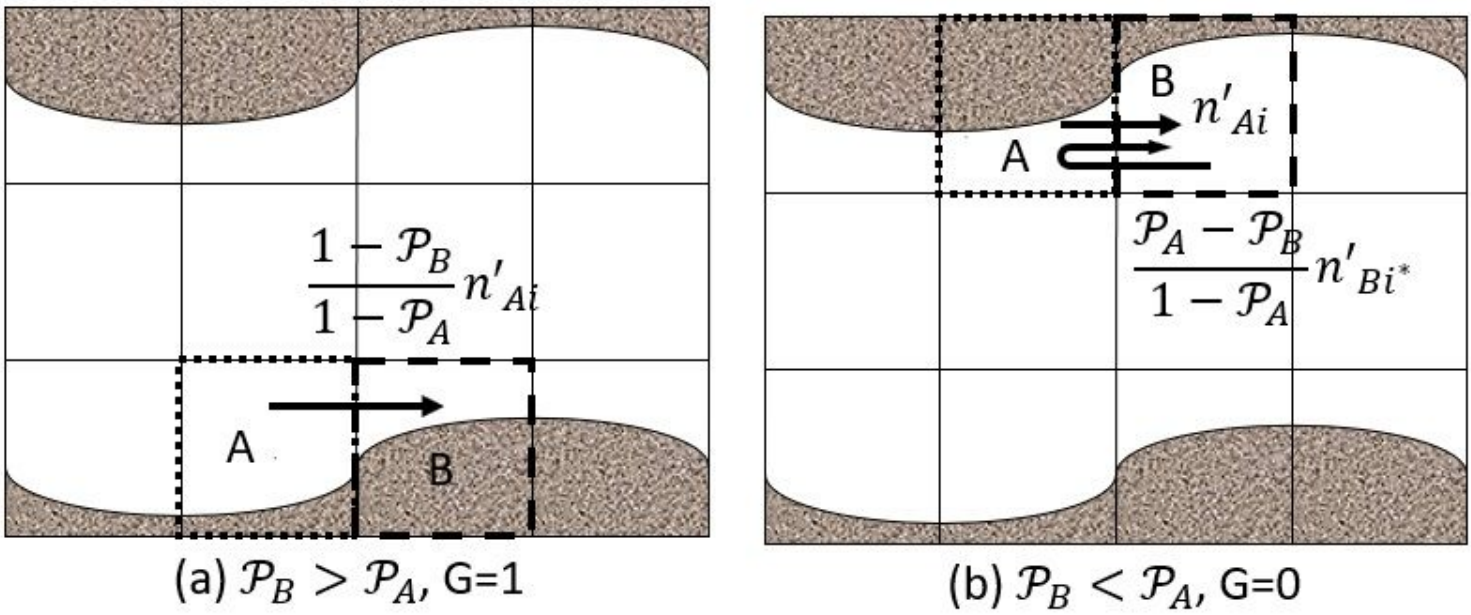


Figure 3

(see Manuscript file for full figure caption)

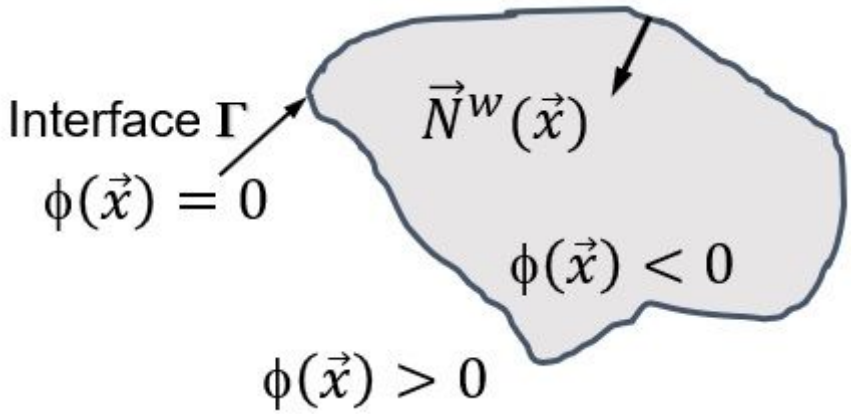


Figure 4

(see Manuscript file for full figure caption)

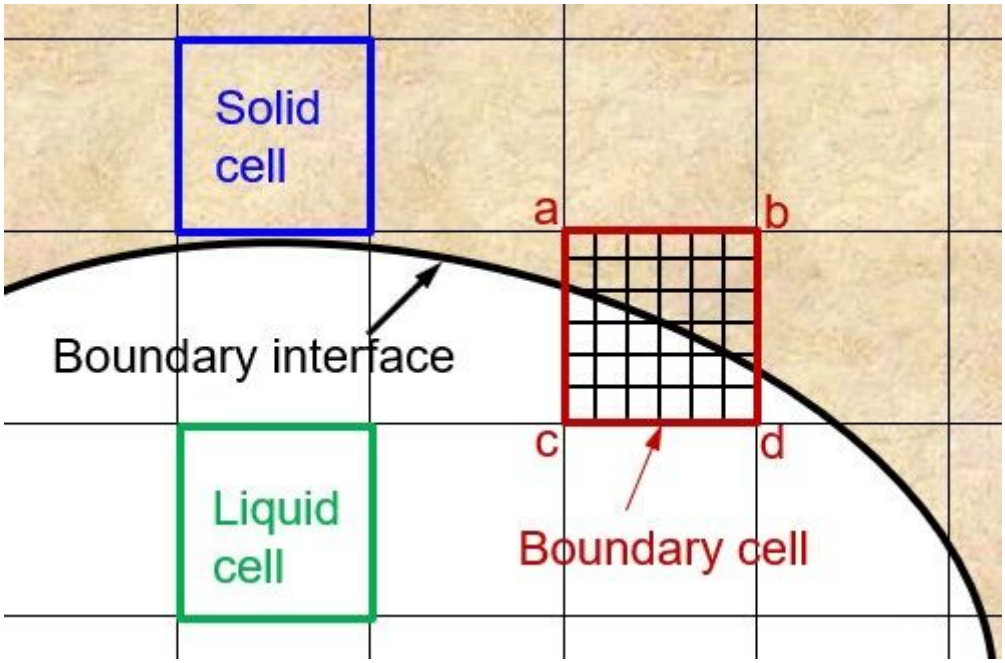


Figure 5

(see Manuscript file for full figure caption)

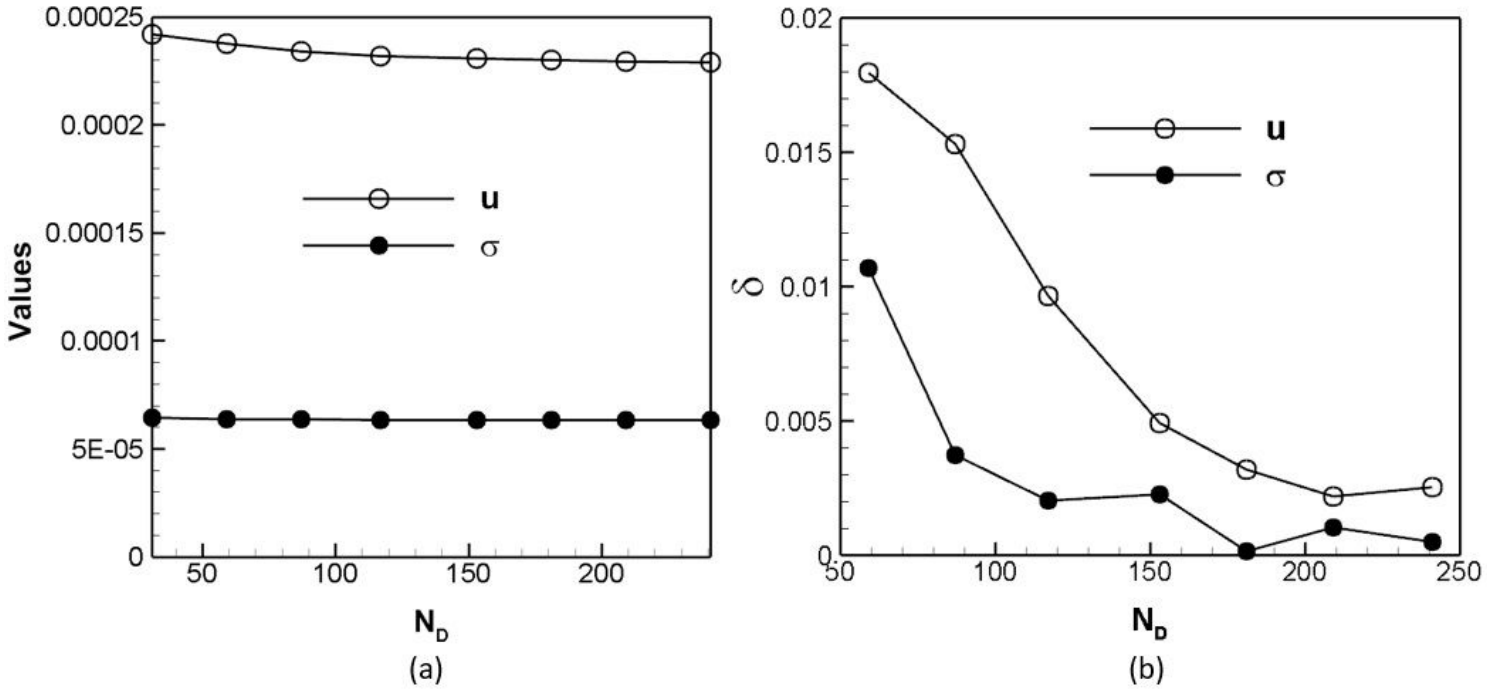


Figure 6

(see Manuscript file for full figure caption)

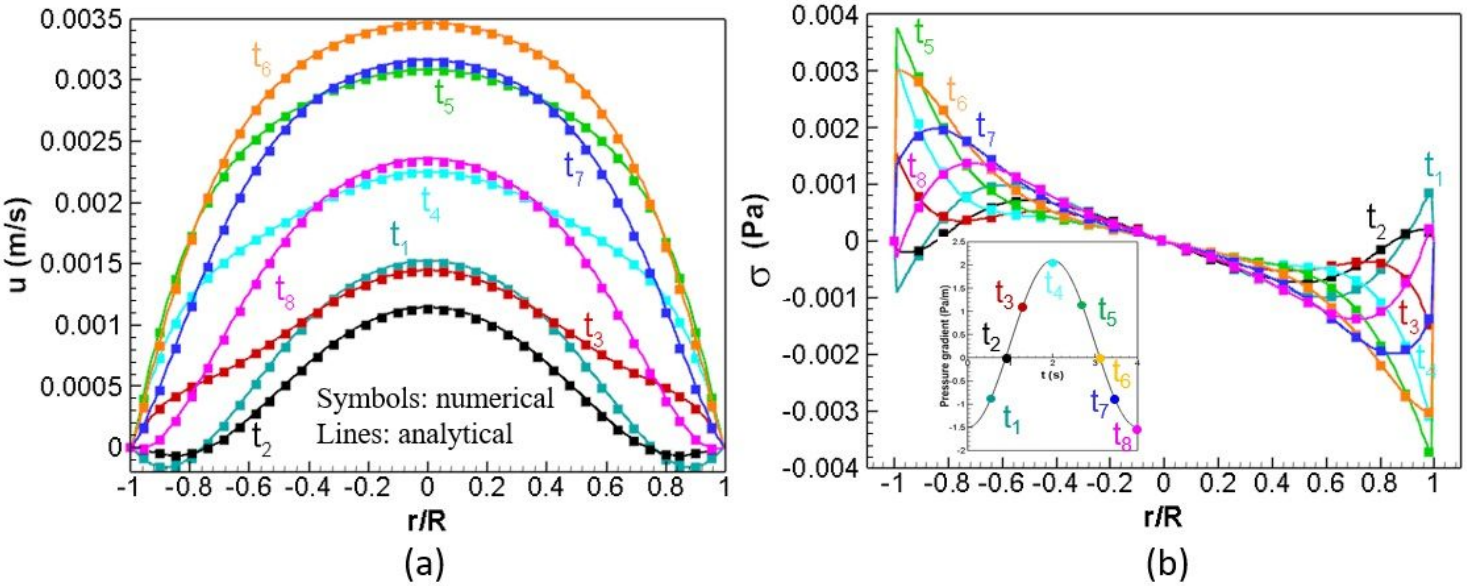


Figure 7

(see Manuscript file for full figure caption)

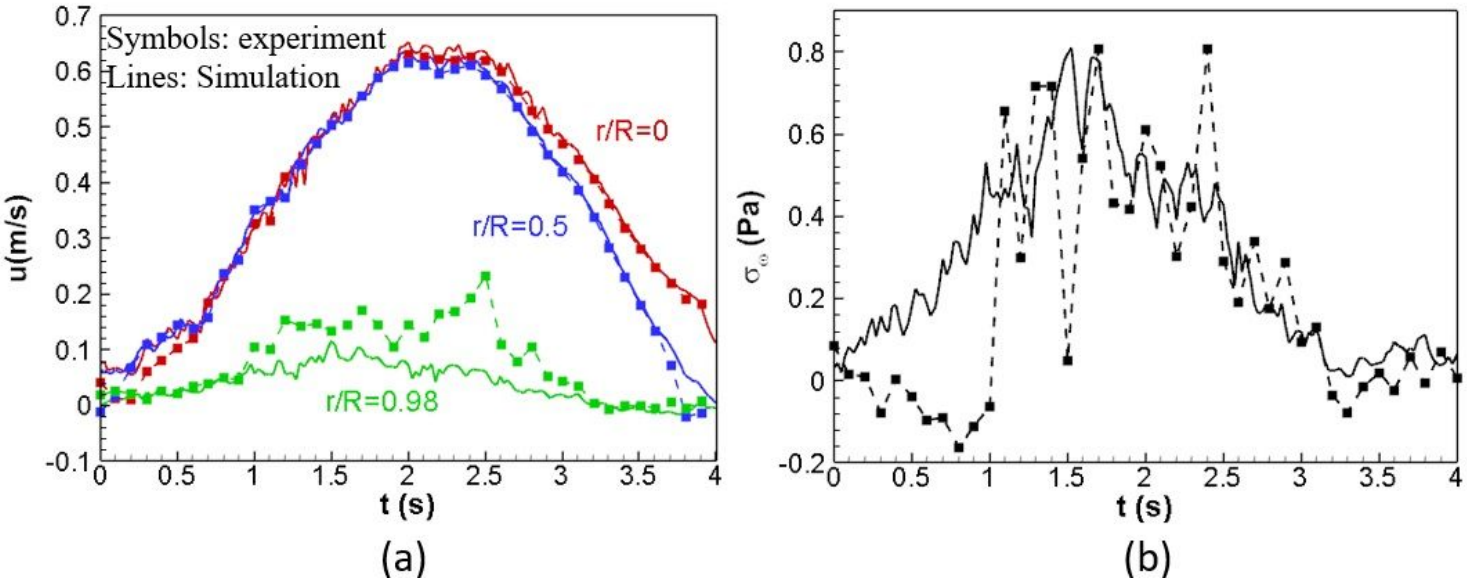


Figure 8

(see Manuscript file for full figure caption)