



OPEN

# Volumetric lattice Boltzmann method for wall stresses of image-based pulsatile flows

Xiaoyu Zhang<sup>1</sup>, Joan Gomez-Paz<sup>2</sup>, Xi Chen<sup>1</sup>, J. M. McDonough<sup>3</sup>, Md Mahfuzul Islam<sup>1</sup>, Yiannis Andreopoulos<sup>2</sup>, Luoding Zhu<sup>4</sup> & Huidan Yu<sup>1,5</sup>✉

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 computational platform, to compute wall stresses in image-based pulsatile flows using the volumetric lattice Boltzmann method (VLBM). The novelty includes: (1) a unique image processing to extract flow domain and local wall normality, (2) a seamless connection between image extraction and VLBM, (3) an en-route calculation of strain-rate tensor, and (4) GPU acceleration (not included here). We first generalize the streaming operation in the VLBM and then conduct application studies to demonstrate its reliability and applicability. A benchmark study is 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 agreements with Womersley's analytical solutions for laminar pulsatile flows and concurrent laboratory measurements for turbulent pulsatile flows. An application study is to quantify the pulsatile hemodynamics in image-based human vertebral and carotid arteries including velocity vector, pressure, and wall-shear stress. The computed velocity vector fields are in reasonably well agreement with MRA (magnetic resonance angiography) measured ones. This computational platform is good for image-based CFD with medical applications and pore-scale porous media flows in various natural and engineering systems.

Pulsatile flows are omnipresent in a wide range of engineering and scientific systems. Examples include pulmonary ventilating<sup>1,2</sup> and blood circulating<sup>3</sup> in biological flows, sediment transport in coastal flows<sup>4</sup>, and reciprocating flow in internal combustion engines<sup>5</sup>. Understanding turbulence in pulsatile flows is critically important to both knowledge advancement and technological innovation but challenges have remained due to the complexities 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<sup>6,7</sup>. Since the seminal work of Reynolds<sup>8</sup>, significant efforts have been made to study the transition to turbulence in pulsatile flows<sup>9–13</sup>. The focus has been on the exploration of criteria for laminar, transitional, and turbulent flow in pulsatile flows from different perspectives including hydrodynamic stability<sup>14</sup> and instability<sup>15</sup>, pulsatility<sup>16</sup>, and turbulence dynamics<sup>17</sup>. 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'). The wall stresses play central roles, not only in aneurysm initiation, growth, and rupture<sup>18</sup> but also in the development of

<sup>1</sup>Department of Mechanical and Energy Engineering, Indiana University-Purdue University, Indianapolis (IUPUI), Indianapolis, IN 46202, USA. <sup>2</sup>Department of Mechanical Engineering, The City College of New York, New York, NY 10031, USA. <sup>3</sup>Departments of Mechanical Engineering and Mathematics, University of Kentucky, Lexington, KY 40506, USA. <sup>4</sup>Department of Mathematical Sciences, Indiana University-Purdue University, Indianapolis (IUPUI), Indianapolis, IN 46202, USA. <sup>5</sup>Department of Vascular Surgery, Indiana University School of Medicine, Indianapolis, IN 46202, USA. ✉email: whyu@iupui.edu

atherosclerosis<sup>19</sup>. With the advances in medical imaging, scientific modeling, and computational power, image-based computational fluid dynamics (ICFD) has emerged<sup>20–26</sup> 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 angiogram (MRA), 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<sup>19,27</sup>, 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.

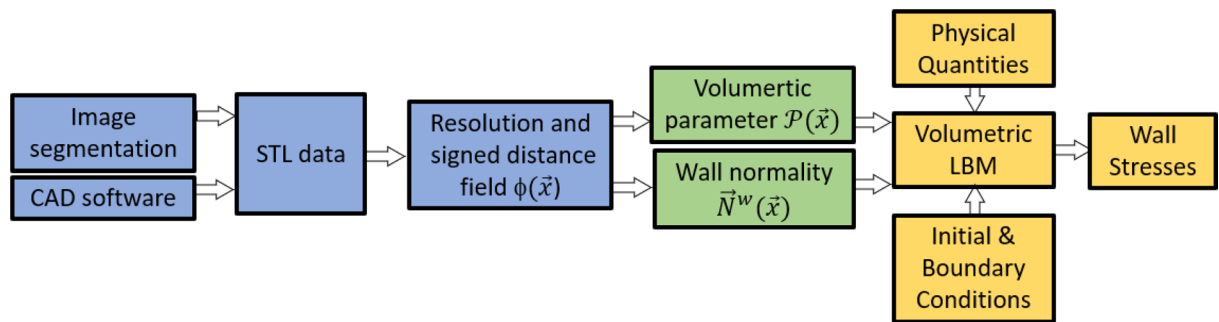
A general process of ICFD for wall stresses mainly involves four tasks: (1) *extraction* of flow domain and local normal orientation from image data, (2) *connection* between output of image processing and input of CFD, (3) *quantification* of 4-D velocity and pressure fields employing physical parameters 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 1<sup>st</sup>-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 for a regular cubic mesh with a Cartesian coordinate system 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 flows using the volumetric lattice Boltzmann method (VLBM)<sup>28</sup> 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<sup>29–32</sup> (not included in this paper) to significantly mitigate the computation burden. The kinetic-based lattice Boltzmann modeling has emerged for simulating a broad class of complex flows including pore-scale porous media flow<sup>33–37</sup>, multiphase/multicomponent flows<sup>38–42</sup>, and turbulence<sup>40,43–47</sup>. The main advantages related to 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<sup>29,31,32,40,43–48</sup> to achieve fast computation. The VLBM was specially developed for dealing with arbitrarily complicated flow domains, e.g., porous media and blood capillaries, with or without moving boundaries. In this work, we generalize the streaming operation from its original formulation<sup>28</sup>.

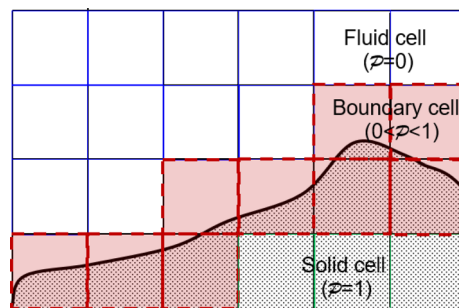
Aiming toward broad applications for not only biomedical flows but also parametric-designed flows in nature and engineering, we start with STL (Standard Triangle Language) data format for images in this computational platform. Medical images from CTA or MRA are in DICOM (Digital Imaging and Communications in Medicine) format. Image segmentation is required to extract the anatomical vessel before the vessel geometry is fed to the VLBM for hemodynamics. The raw DICOM images usually involve noises from different resources and cleaning up the noises is always a critical part of the tasks for an image segmentation to extract the anatomical vessels. Such a task is usually case to case with a lot of uncertainties, related to the imaging modality, machine resolution, scanning skills, disease conditions, etc. Although we have developed techniques for DICOM image segmentation<sup>30,49,50</sup>, we strongly recommend having rigorous image segmentation with expertise, instead of simply running software with extensive manual inputs, as it will affect the quantification of the hemodynamics. All the modern CAD (Computer-Aided Design) software packages such as SolidWorks export their native file format into STL format. The conversion from DICOM segmentation to an STL format<sup>51</sup> is straightforward.

We conduct two studies to demonstrate the reliability and applicability of the computational method. The first is a systematic benchmark study of pulsatile flows in an image-based pipe. 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 second is an application study to quantify 4-D hemodynamics in MRA-based human cerebral and carotid arteries.

The remainder of the paper is organized as follows. In “Volumetric lattice Boltzmann modeling and formulation for wall stresses in ICFD”, we present the computational method including the VLBM, the image processing, the connection of both, and the en-route calculation of WSS and WNS. A benchmark study of pulsatile flows, both laminar and turbulent flows, in an image-based pipe and an application study of blood flow in MRA-based human arteries are carried out in “Benchmark study of oscillating flows in a 3-D pipe” and “Application study”, respectively. Finally, “Summary and discussion” concludes the paper with a summary and discussion.



**Figure 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 VLBM (yellow), enabling seamless connection between both.



**Figure 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.

### Volumetric lattice Boltzmann modeling and formulation for wall stresses in ICFD

As schematized in Fig. 1, the VLBM for wall stresses in ICFD consists of three components: image acquisition and segmentation (blue), determination of volumetric parameter and wall normality (green), and CFD using VLBM 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 will be generalized from its original formula<sup>28</sup>.

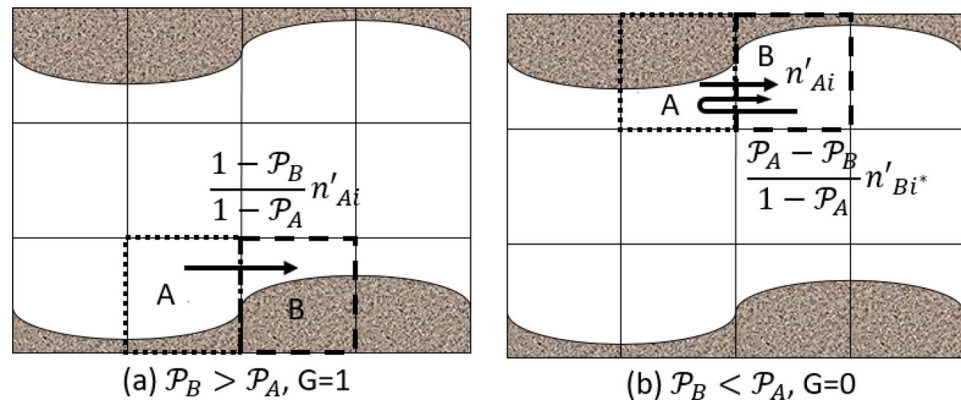
**Formulation of volumetric lattice Boltzmann method.** The VLBM<sup>28</sup> was developed specifically for treating arbitrarily oriented boundaries with or without boundary movement. In the VLBM, fluid particles are uniformly distributed in lattice cells. By introducing a volumetric parameter  $\mathcal{P}(\vec{x}, t)$ , 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.

Formulation of the VLBM 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 VLBM 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 calculation from STL image data. Thus, the collision reverts to the conventional operation in LBM and the migrating operation is omitted.

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

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

where  $c (= \frac{\delta x}{\delta t})$  with  $\delta x$  and  $\delta t$  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  $w_i$  are  $w_0 = \frac{1}{3}$ ,  $w_{1-6} = \frac{1}{18}$ , and  $w_{7-18} = \frac{1}{36}$ .



**Figure 3.** Illustration of two streaming scenarios when the current cell B at  $\vec{x}$  (dashed) and upwind cell A (dotted) 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.

The VLBM deals with the time evolution of the particle population in each lattice cell, and the volumetric LBE with BGK (Bhatnagar, Gross, and Krook) approximation<sup>52</sup> for the collision operation reads

$$n_i(\vec{x} + \vec{e}_i\delta t, t) - n_i(\vec{x}, t) = -\frac{1}{\tau} [n_i(\vec{x}, t) - n_i^{eq}(\vec{x}, t)] + F_i(\vec{x}, t), i = 0 \sim 18 \quad (1)$$

where  $n_i(\vec{x}, t)$ ,  $n_i^{eq}(\vec{x}, t)$ ,  $\tau$ , and  $F_i(\vec{x}, t)$  are the particle population, the equilibrium particle population, the relaxation time due to fluid-particle collisions, and the forcing term, respectively, in cell  $\vec{x}$  at time  $t$ . The forcing term is formulated<sup>44,53</sup> as

$$F_i(\vec{x}, t) = -3\omega_i N \frac{\vec{e}_i \cdot \vec{a}(\vec{x}, t)}{c^2} \delta t \quad (2)$$

with  $\vec{a}$  the acceleration due to an external force. The equilibria for incompressible flows are

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

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 VLBM is  $n_i(\vec{x}, t) = [1 - \mathcal{P}(\vec{x})]f_i(\vec{x}, t)$ . Since  $\Delta V_f(\vec{x}, t) = [1 - \mathcal{P}(\vec{x})]\Delta V$ ,  $n_i(\vec{x}, t)$  collapses to  $f_i(\vec{x}, t)$  in fluid 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.

$$n'_i(\vec{x}, t) = n_i(\vec{x}, t) - \frac{1}{\tau} [n_i(\vec{x}, t) - n_i^{eq}(\vec{x}, t)] + F_i(\vec{x}, t) \quad (4)$$

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<sup>28</sup>, the streaming equation is formulated as follows.

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

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 first term is the streaming term. All the particles in the upwind neighboring cell,  $n'_i(\vec{x} - \vec{e}_i\delta t, t)$ , stream to the current cell but the current cell can only accept  $[1 - \mathcal{P}(\vec{x})] \cdot n'_i(\vec{x} - \vec{e}_i\delta t, t)$ . This part assumes that the upwind cell is a fluid cell. The second term is the bounce-back term.  $[1 - \mathcal{P}(\vec{x} + \vec{e}_{i*}\delta t)] \cdot n'_{i*}(\vec{x}, t)$  particles stream to the upwind neighboring cell from the current cell and  $\mathcal{P}(\vec{x} + \vec{e}_{i*}\delta t) \cdot n'_{i*}(\vec{x}, t)$  particles bounce back to the current cell. This part assumes that the current cell is a fluid cell. However, either the current or the neighboring cell can be a boundary cell, with which Eq. (5) becomes problematic because the capability to stream and/or receive particles is determined by the relative fluid volume between the two 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. 3a, 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 3b 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

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

and generalize Eq. (5) 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:

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

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

and

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

where  $p_0$  and  $\rho_0 (= 1)$  are reference pressure and density in lattice units, respectively.

The LBM can handle arbitrary oriented boundaries 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 VLBM is the determination of  $\mathcal{P}(\vec{x})$  for a flow domain.

**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 VLBM as the CFD solver, a 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. 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 STL image processing (blue part) and the input of VLBM (yellow part), enabling a seamless connection between image information and CFD. The geometrical process consists of three main components described in the following subsections.

**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<sup>54</sup>. 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. When a flow domain is closed and bounded by an interface  $\Gamma$  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

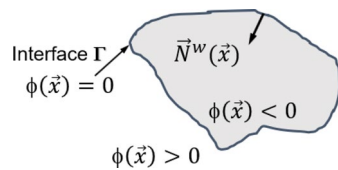
$$\phi(\vec{x}) = \begin{cases} > 0, & \vec{x} \in \Gamma \\ = 0, & \vec{x} \text{ on } \Gamma \\ < 0, & \vec{x} \notin \Gamma \end{cases} \quad (11)$$

with

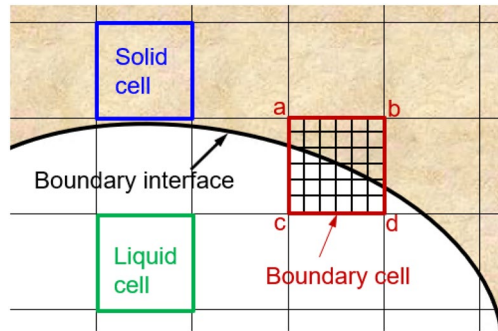
$$|\nabla \phi| = 1$$

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*<sup>55</sup>, and *ac-reinit.m*<sup>56</sup> 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.

**Volumetric parameter  $\mathcal{P}(\vec{x})$  and wall-normal direction  $\vec{N}^w(\vec{x})$  from signed distance field  $\phi(\vec{x})$ .** For an illustration, we use a 2-D lattice, in Fig. 5, to describe how to determine the  $\mathcal{P}$  value for a cell from the



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



**Figure 5.** Identification of fluid cell, solid cell, and boundary cell based on the  $\phi$  signs at four vertexes of each cell. All negative  $\phi$ s mean a fluid cell (green,  $\mathcal{P} = 0$ ), all positive  $\phi$ s 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,  $0 < \mathcal{P} < 1$ ).

signed distance field  $\phi(\vec{x})$ . Given the interface, which is the boundary of the flow domain, the sign  $\phi$  at each node of the lattice indicates the area, either fluid (white) or solid (orange), and the node location. If the  $\phi$ s 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  $\phi$ s 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.

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  $\phi$  value at its  $\vec{x}_s^j$  using the known  $\phi$  values at 4 vertices of the cell through linear 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  $\vec{x}_b = \frac{\sum_{j=1}^{q^3} [1 - V_s^j(\vec{x}_s^j)] \vec{x}_s^j}{\sum_{j=1}^{q^3} [1 - V_s^j(\vec{x}_s^j)]}$ . Then based on the volumetric center

$\vec{x}_b$  and the distance functions  $\phi$ s at the eight vertices of the cubic cell, linear interpolation results in the direct distance of the volumetric center to the boundary surface.

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. We use  $\vec{x}_{ver}^j$  with  $j = 1, \dots, 8$  to denote a vertex's location. With the available distance field  $\phi(\vec{x})$ , the gradient  $\nabla\phi(\vec{x}_{ver}^j)$  results in a vector indicating the normal direction and the steepness of the local boundary surface. Using a linear interpolation from the distance of  $\vec{x}_{ver}^j$  with  $j = 1, \dots, 8$  to the  $\vec{x}_b$ , we get the  $\nabla\phi(\vec{x}_b)$ . The normal direction of the boundary cell can then be determined as follows

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

**En-route calculation of wall stresses.** In hydrodynamics, the strain-rate tensor is defined as  $S_{\alpha\beta} \equiv \frac{1}{2} \left( \frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} \right)$ , where  $\alpha (= 1, 2, 3)$  and  $\beta (= 1, 2, 3)$  represent the indices of the axial directions in a 3-D Cartesian coordinate system. We use these notations in the LBM, it can be calculated from the non-equilibrium distribution function as<sup>57</sup>

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

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<sup>44</sup>.

In the VLBM, Eq. (13) becomes

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

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

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

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

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

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

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

$$T_\alpha^{(WNS)} = (N_\beta^w T_{\gamma\beta} N_\gamma^w) N_\alpha^w \quad (17)$$

Here  $\gamma (= 1, 2, 3)$  is also an index for the axial direction in the 3-D Cartesian coordinate system. The WSS is then computed as the difference between the overall stress and its projection onto the normal:

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

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 VLBM implementation.

## Benchmark study of oscillating flows in a 3-D pipe

We now apply the VLBM to simulate pulsatile laminar and turbulent 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.

**Computation set-up.** 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 pulsatile flow along the pipe in the  $z$ -direction is driven by either a pressure gradient,  $P(t)$  or a velocity profile,  $\vec{u}_{in}(x, y, t)$ , which will be specified below. A no-slip boundary condition is applied on the pipe wall, realized by the bounce-back boundary condition. It is noted that in VLBM, the bounce-back boundary condition is involved in the streaming operation, see Eqs. (6) and (7), regulated by the volumetric parameter  $\mathcal{P}(\vec{x})$ . The initial conditions are constant pressure and zero velocity in the entire flow domain. A uniform mesh in the VLBM is used with the cell number,  $N_D$ , across the pipe diameter to represent the spatial resolution.

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. As mentioned in “[Volumetric parameter  \$\mathcal{P}\(\vec{x}\)\$  and wall-normal direction  \$\vec{N}^w\(\vec{x}\)\$  from signed distance field  \$\phi\(\vec{x}\)\$](#) ”, when the volumetric parameter  $\mathcal{P}(\vec{x})$  of a boundary cell at location  $\vec{x}$  is calculated, the boundary cell is divided into  $q^3$  sub-cells and  $q$  should be appropriately chosen to balance between accuracy and computation cost. We perform an analysis on this theme. First, we evaluate the influence of the spatial resolution represented by  $N_D$  through the relative error,  $\delta_V$  (%), of the  $\mathcal{P}(\vec{x})$ -based fluid volume (summation of  $1 - \mathcal{P}(\vec{x})$  over the entire computation domain) from the specified pipe volume ( $2\pi R^2 L$ ). Table 1 shows, when  $q = 8$  is selected, the error reduces to below 1% when  $N_D = 127$ . We then stick to  $N_D = 127$  to study the effects of  $q$  on the computation time and accuracy of  $\mathcal{P}(\vec{x})$  in Table 2. It is seen that when  $q$  increases, the computation time increases significantly but the accuracy remains almost the same. Since our general goal for accuracy is less than 1%, we use  $q = 8$  to compute the  $\mathcal{P}(\vec{x})$  field with  $N_D > 127$  in what follows. The wall-clock time was recorded on a computer with Inter®Xeon® CPU E5-1650 v3 @ 3.50 GHz, RAM 32 GB, 64-bit Operating System, x64-based processor, and Matlab R2020a.

|                |     |     |      |
|----------------|-----|-----|------|
| $N_D$          | 65  | 91  | 127  |
| $\delta_V$ (%) | 1.8 | 1.3 | 0.35 |

**Table 1.** Influence of spatial resolution  $N_D$  on the accuracy of  $\mathcal{P}(\vec{x})$  at  $q = 8$ .

| $q$               | 4    | 8    | 12   | 16   |
|-------------------|------|------|------|------|
| Wal-clocktime (s) | 203  | 244  | 344  | 588  |
| $\delta_V$ (%)    | 0.35 | 0.34 | 0.34 | 0.34 |

**Table 2.** Effects of  $q$  on the computation time and accuracy of  $\mathcal{P}(\vec{x})$  at  $N_D=127$ .

**Laminar pulsatile flows.** In this section, we simulate a Womersley (laminar) flow driven by a pulsatile pressure gradient,  $P(t) = P_s + Re(P_0 e^{i\omega t})$ . We use  $P_s = P_0 = 0.02 \text{ Pa/m}$  and  $\omega = 6.28 \text{ s}^{-1}$ . The pressure gradient is introduced as a body force (external force)  $F_i$  in Eq. (1) via Eq. (2) where  $\vec{a}(t) = \frac{P(t)}{\rho} \vec{k}$  along the z-direction. With the body force to drive the flow, we impose a periodic boundary condition along z-direction. The density and kinematic viscosity of the fluid is  $1.0 \text{e}3 \text{ kg/m}^3$  and  $1.0 \text{e}-6 \text{ m}^2/\text{s}$ , respectively.

We first perform a grid-function convergence check to determine the appropriate spatial resolution. Seven  $N_D$ s of 31, 59, 87, 117, 153, 181, 209, and 241 are used. Figure 6a shows the effects of  $N_D$  on the spatial mean downstream velocity  $u$  and mean shear stress  $\sigma$  on a cross-section. It is seen that each of the curves becomes flat as  $N_D$  increases. The quantitative measurement of the convergence is shown in Fig. 6b, in which the convergence rate  $\delta$  is calculated as the relative difference of the quantity of interest—velocity or shear stress—between two successive spatial resolutions. It indicates that when  $N_D \geq 150$ ,  $\delta < 0.5\%$  for both velocity and shear stress. For the simulations of laminar and turbulent flows below,  $N_D$  is in the range of 150–240. Correspondingly, the relaxation time  $\tau$  is in the range of 0.55–0.95.

**Order of accuracy.** The order of accuracy of a finite-difference-related numerical scheme is usually evaluated by the discretization error. As pointed out in Ferziger's textbook<sup>58</sup>, the exact solution of the discretized equations on the grid  $h(= \frac{2R}{N_D})$ ,  $\varphi_h$ , differs from the exact solution of the partial differential equation,  $\Phi$ , by the discretization error  $\delta_h$ , i.e.,  $\delta_h = \Phi - \varphi_h$ . For sufficiently fine grids, this discretization error is proportional to the leading term in the Taylor series:  $\delta_h \approx \alpha h^q + H$  where  $H$  stands for higher-order terms and  $\alpha$  depends on the derivatives at the given point but is independent of  $h$ . The discretization error can be estimated if solutions on systematically refined grids are compared through the expression  $\Phi = \varphi_h + \alpha h^b + H = \varphi_{3h} + \alpha (3h)^b + H$  with 3 times (for the cell-based cubic mesh) for the increase of reference spacing from  $h$ . The exponent  $b$  represents the order of the numerical scheme and can be estimated as follows

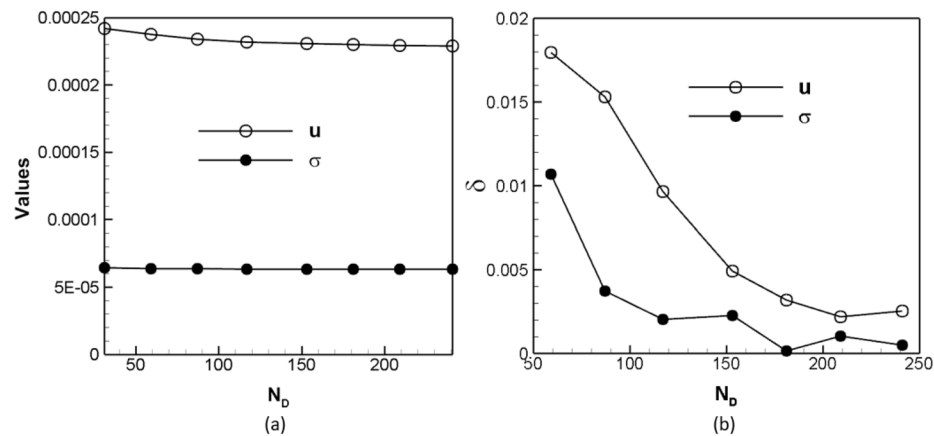
$$b = \frac{\log\left(\frac{\varphi_{3h} - \varphi_{9h}}{\varphi_h - \varphi_{3h}}\right)}{\log 3} \quad (19)$$

We simulated the flow with three successive resolutions with  $N_D = 31, 93, 279$ . Using the velocity magnitude at three locations  $r/R = 0, 0.5$ , and  $0.86$ , we calculate the orders of accuracy for the three locations using Eq. (19) and obtained  $b = 1.2, 1.2$ , and  $1.1$ , respectively, which indicates that the spatial accuracy of VLBM is 1.2 in interior and 1.1 at the boundary. It is well-known that the spatial accuracy of node-based LBM is theoretically no more than the second order in the interior. The accuracy reduces to the first order near the boundary with a bounce-back boundary condition. When the wall lies exactly at the nodes or middle of two neighboring nodes (half-way bounce-back).

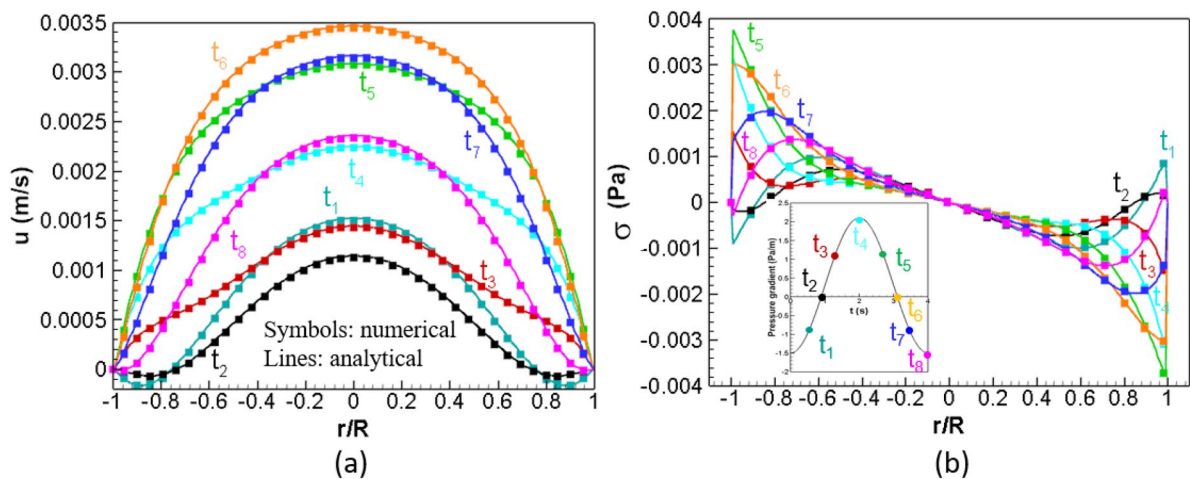
**Computational results vs. analytical solutions of Womersley flow.** We use the analytical solutions of Womersley flow<sup>59</sup> to verify the computational results. A Womersley flow, driven by a pulsatile pressure gradient,  $P = P_s + P_0 e^{i\omega t}$ , has the following analytical solutions of downstream velocity:

$$u(r, t) = \frac{P_s R^2}{4\mu} \left(1 - \frac{r^2}{R^2}\right) + \text{Re} \left\{ \frac{P_0 R^2}{i\mu\alpha^2} \left[ 1 - \frac{J_0\left(\frac{\alpha r}{R}\right)}{J_0(\alpha)} \right] e^{i\omega t} \right\} \quad (20)$$

where  $u$  is the downstream velocity,  $J_0(\cdot)$  is the Bessel function of the first kind and order zero,  $\text{Re}(\cdot)$  is the real part of a complex function,  $\alpha(= R\sqrt{\frac{\omega}{\nu}})$  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  $\sigma_w = \sigma|_{\text{wall}}$  can be calculated from Eq. (20). Keeping the same pipe configuration, we use  $P_s = 0.3 \text{ Pa}$ ,  $P_0 = 1.8 \text{ Pa}$ , and  $\omega = 1.57 \text{ s}^{-1}$  to simulate the flow in this part. The Womersley number  $\alpha$  and the base flow Reynolds number  $Re_0 = \frac{2Ru_0}{\nu}$ , 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  $\sigma$  (b) and good agreements between them are shown. More quantitative comparisons are presented in Table 3 with the relative difference between the numerical



**Figure 6.** Convergence checks for determining appropriate spatial resolution: (a) Spatial mean velocity  $u$  (empty circle) and shear stress  $\sigma$  (solid circle) and (b) relative differences of  $u$  and  $\sigma$  between two successive resolutions ( $\delta$ ) versus the cell number along the diameter ( $N_D$ ).



**Figure 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  $\sigma$  (b), respectively. The Womersley number  $\alpha$  and the base flow Reynolds number  $Re_0$  of the flow are 6.89 and 9.4, respectively.

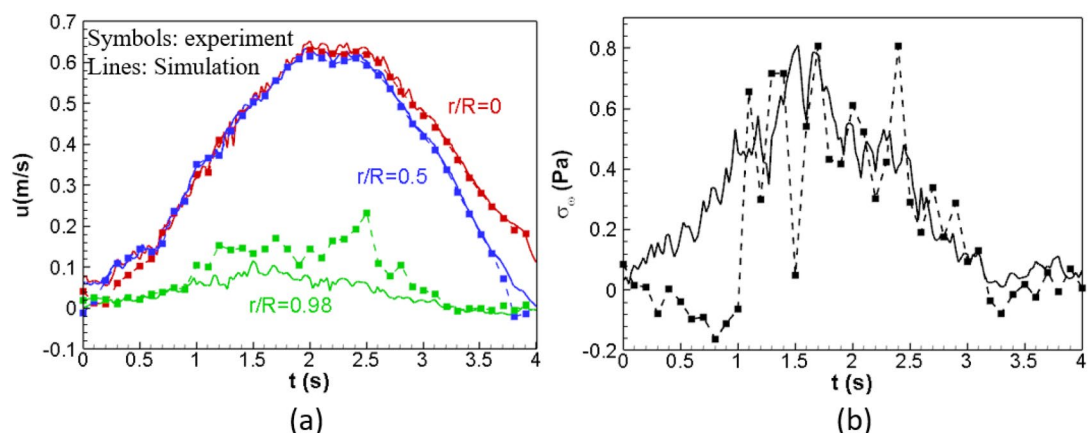
|            | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| $u$        | 2.42% | 2.38% | 1.12% | 1.06% | 1.09% | 1.15% | 1.29% | 1.68% |
| $\sigma_w$ | 5.16% | 6.57% | 4.30% | 2.96% | 2.42% | 2.36% | 4.15%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

**Table 3.** 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  $\sigma_w$  on a cross-section of the pipe at the same representative time instants as embedded in Fig. 7b. The average relative errors at the 8 time points are 1.53% for velocity and 3.98% for WSS. The analytical solution of  $\sigma_w$  at  $t_8$  tends to zero thus the relative error is not available.

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  $\sigma_w$  (WSS) on a cross-section of the pipe at the same representative time instants as embedded in Fig. 7b. 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. Last but not the least, we demonstrate how Eq. (7) improves the modeling of the generalized streaming operation in the VLBM. Keeping everything else the same, we use Eqs. (5) and (7) to solve the velocity field. The accuracy of the spatial mean velocity comparing with the analytical solutions from Eq. (20) at the 8 time-points in one cycle is presented in Table 4. The average relative error is reduced to 0.653% (Eq. (7)) from 0.721% (Eq. (5)).

|              | $t_1$  | $t_2$  | $t_3$  | $t_4$  | $t_5$  | $t_6$  | $t_7$  | $t_8$  |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Equation (5) | 0.717% | 0.709% | 0.713% | 0.716% | 0.724% | 0.732% | 0.728% | 0.726% |
| Equation (7) | 0.649% | 0.642% | 0.647% | 0.650% | 0.657% | 0.663% | 0.660% | 0.658% |

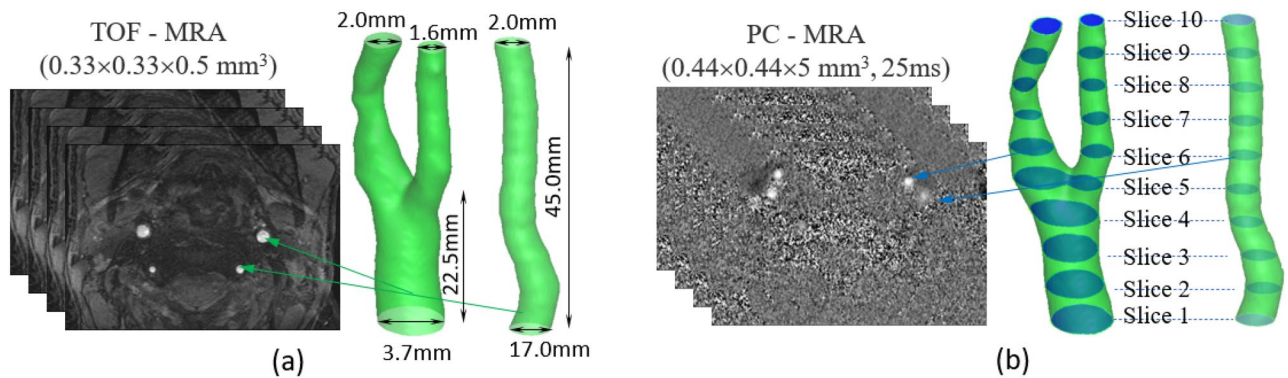
**Table 4.** Relative errors calculated by  $\sqrt{\frac{\sum_i (u_{num} - u_{ana})^2}{\sum_i u_{ana}^2}}$ , where  $u$  is the velocity at the centerline of cross section of a pipe at the same representative time. The average relative errors at 8 time points are 0.721% for the original equation, Eq. (5), and 0.653% for the generalized equation, Eq. (7).



**Figure 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  $\alpha = 11.91$ .

**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<sup>60</sup>. 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 5.5 W 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  $1200 \times 800$  pixel resolution. In each experiment, 20,000 images were acquired with a rate of 1 kHz. Hollow glass particles of 9–13  $\mu\text{m}$  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  $128 \times 128$  pixels. It is reduced to a final window size of  $16 \times 16$  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  $\alpha$  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 via direct numerical simulation. The diameter of the pipe (19.05 mm) and the fluid properties, i.e., density ( $998.008 \text{ kg/m}^3$ ) and kinematic viscosity ( $1.004 \times 10^{-6} \text{ m}^2/\text{s}$ ), 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., 0.25 Hz. We use the velocity profiles at downstream locations at 183 and 184.55 as the inflow and outflow boundary conditions, respectively. They are introduced in the VLBM via a non-equilibrium extrapolation scheme<sup>61</sup>. We use  $N_d = 243$  and  $\tau = 0.506$  for direct numerical simulations of turbulence. The spatial-temporal velocity information at each location is introduced in the D3Q19 lattice cell model on the corresponding cross-section via piecewise cubic Hermite 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 WSS representing the experimental results in Fig. 8b was calculated from the experimentally measured velocity fields via a 5-point forward FDM. Figure 8a 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. 8a) and WSS (Fig. 8b) waves from simulation still agree with the experimental measurements, but the simulation results are consistently less noisy than those from experiments.



**Figure 9.** (a) High resolution 3-D TOF-MRA images and the segmented left carotid artery (with a bifurcation) and vertebral artery (tube-like). (b) 4-D Low resolution 4-D PC-MRA images with 10 locations along the arteries and 20 time points in one cardiac cycle.

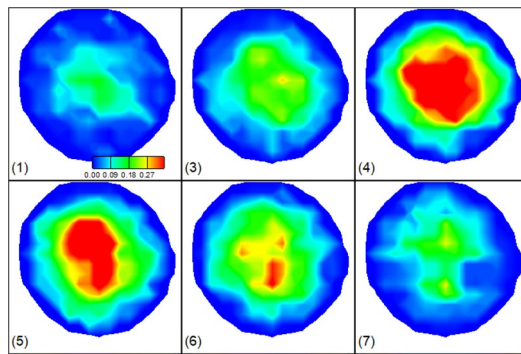
### Application study

Beyond the benchmark study, we now apply the presented computational method to a medical research project to explore the capability of non-invasive quantification of 4-D hemodynamics in image-based human vessels. Our focus is the pulsatile velocity, pressure, and WSS in human vertebral and carotid arteries based on MRA.

Cardiovascular disease is known to be strongly influenced by hemodynamic WSS. Although the advanced diagnostic imaging modalities such as digital subtraction angiography, CTA, ultrasound, and MRA can provide very good information of the vessel morphology and the total blood flow, they cannot measure the WSS, which is believed to play a pertinent role in the development of human vascular diseases as demonstrated in many studies<sup>62–65</sup>. Without the quantitative physiological parameters of the flow and blood-vessel interaction, it remains challenging to fully assess the true risk of vascular lesions to optimize treatment decisions. In this project, we aim to integrate patient-specific ICFD and WSS analysis into diagnostic MRA of vascular diseases focusing on carotid arteries. To demonstrate the reliability and applicability of the VLBM-based ICFD for realistic flow systems, we present one study case for the accuracy of the computed hemodynamics including velocity and WSS in human vertebral and carotid arteries following the flowchart in Fig. 1.

The image data were acquired from the scanning of 3-D time-of-flight (TOF) and peripheral gated (PG) phase-contrast (PC) MRA on a 3T clinical MRI scanner (Siemens Magnetom Prisma) using a 64-channel head and neck coil. As shown in Fig. 9a, the TOF-MRA images with a high resolution of  $0.33^2 \times 0.5 \text{ mm}^3$  are to anatomically extract the artery system. The TOF-MRA images are on DICOM data. In general, the image segmentation can be done by different software such as Materialise Mimics and 3D Slicer and exported in an STL format to follow the illustrated flowchart in Fig. 1. Since we have developed a method to do image segmentation from DICOM format and calculate the volumetric parameter  $\mathcal{P}(\vec{x})$  and the normal direction vector  $\vec{N}^w(\vec{x}_b)$  of each boundary cell<sup>29,49</sup>, the STL format is not necessary. We segmented the left carotid (with a bifurcation) and vertebral (tube-like) arteries, shown in Fig. 9a from the 3-D TOF-MRA images. The PC-MRA images in Fig. 9b, provide the 4-D velocity profiles on the cross-sections from the bottom up with a spatial resolution,  $0.44^2 \times 5 \text{ mm}^3$  resulting in 10 slices in the vertical direction, and a time resolution of 25ms resulting in 20 time points in one cardiac cycle. We use the in-house Matlab code to extract the 4-D velocity vector fields from the PC-MRA images for both vertebral and carotid arteries based on the method we developed<sup>66</sup>. Figure 10 shows the velocity magnitude contours on Slice 1 of the vertebral artery at 7 representative time points in one cardiac cycle. The 4th time point corresponds to the peak systole of the heart pumping. We simulated the blood flow in both carotid and vertebral arteries. The dimensions of the two arteries are found in Fig. 9a. The TOF-MRA resolution results in grid sizes of  $106^2 \times 416$  for the carotid artery and  $60^2 \times 416$  for the vertebral artery. The time resolution is 1 ms via linear interpolation of the PC-MRA data in Fig. 9b. The blood flow is driven by the 3-D velocity profiles at the inlet (Slice 1) and outlet(s) (Slice 10). We impose the non-equilibrium extrapolation scheme<sup>61</sup> to introduce the velocity boundary conditions at the inlet and outlet(s). The density and kinematic viscosity of blood are  $1.06 \times 10^3 \text{ kg/m}^3$  and  $3.3 \times 10^{-6} \text{ m}^2/\text{s}$ , respectively. The relaxation time  $\tau = 0.505$ .

We first show the comparisons of computed vs. PC-MRA measured velocity magnitude contours in Fig. 11 (a) vertebral (Slice 4) and (b) carotid (Slice 5) arteries, to demonstrate the accuracy of the computation. On each panel, the top and bottom rows correspond to the peak systole and the end of diastole in a heart beating. Overall, the computed velocity agrees with the MRA measured ones reasonably well. The MRA measured velocity is by nature noisy, appearing at both systole (large velocity) and end of diastole (small velocity.) Fig. 12 shows the 3-D (a) velocity contours with streamlines, (b) pressure contours, and (c) WSS distribution of the carotid artery at the peak systole. While the large velocity and WSS deviate to the internal carotid artery (left), the large pressure deviates to the external artery (right). This is due to the asymmetry between the internal and external carotid arteries. Such a reveal of the 4-D in vivo hemodynamics in live human arteries is meaningful to the patient-specific diagnostics and therapeutics for the advance of precision medicine. Further postprocessing of the computed 4-D velocity, pressure, and WSS will produce more medical relevant results, which will be included in a separate paper in near future.



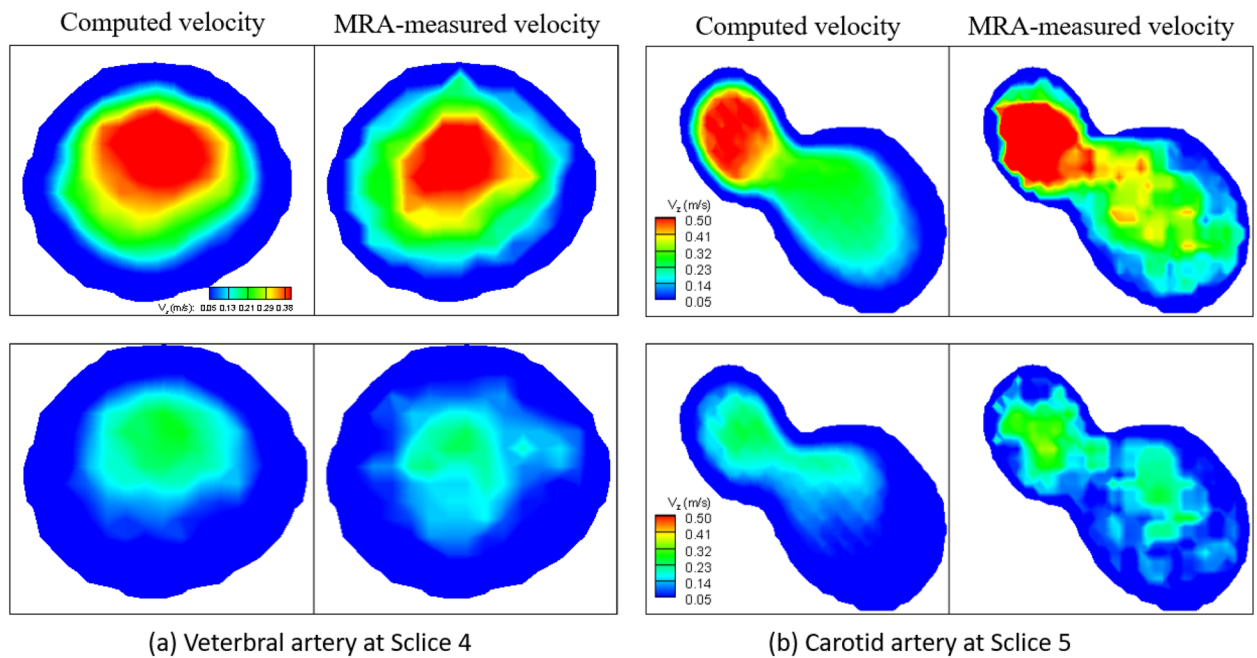
**Figure 10.** MRA-measured velocity magnitude contours on Slice 1 of the vertebral artery at 7 representative time points in a cardiac cycle. (4) corresponds the peak systole of the cycle.

## Summary and discussion

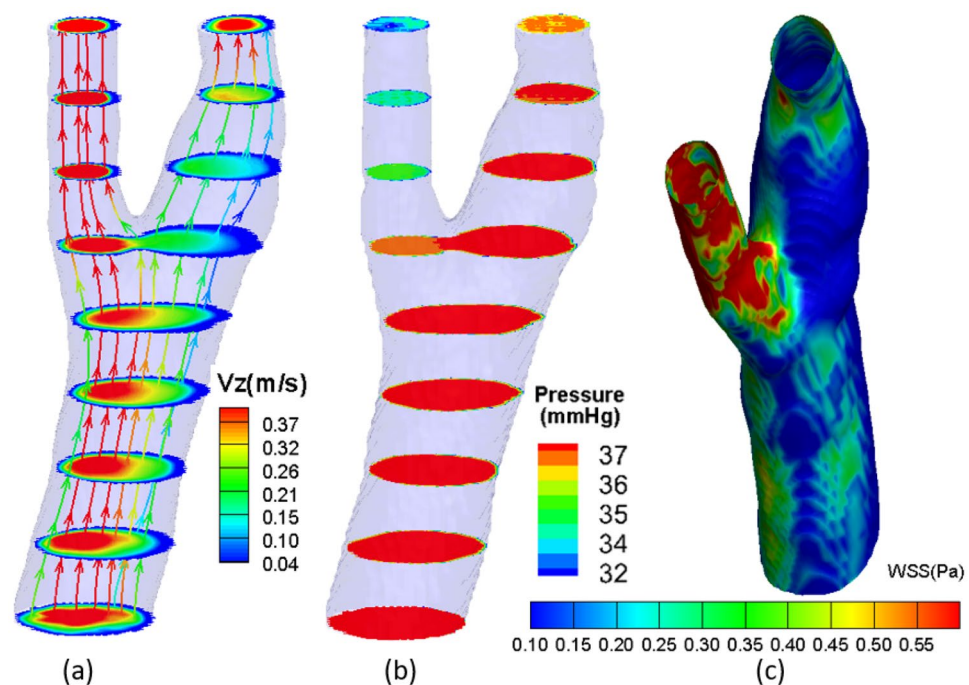
We have presented a unique computational method to quantify 4-D wall stresses of image-based pulsatile flows using the VLBM. With the introduction of a volumetric parameter to distinguish solid, liquid, and boundary cells, the VLBM is well-suited for dealing with complicated flow domains. The bounce-back 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 becomes problematic when the streaming is from a boundary cell to another boundary cell. We generalized Eq. (5), by introducing a parameter  $G$ , see Eq. (7), to distinguish two streaming scenarios and demonstrated the accuracy improvement. 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 VLBM, resulting in a seamless connection between the image processing and CFD. In the VLBM, the strain rate tensor can be calculated from the non-equilibrium particle population; therefore, the wall stresses can be obtained en-route during the VLBM implementation. These features, together with GPU parallel computing, dramatically reduce resource requirements for ICFD and ease the computational burden for pulsatile flows.

We applied the computational method to solve the benchmark pulsatile flows in an image-based pipe. We first simulated a Womersley (laminar) flow with Womersley number  $\alpha$  and the base flow Reynolds number  $Re_0$  of 6.89 and 9.4, respectively. The computational results were in good agreement with the analytical solutions, with average relative errors around 1.52% and 3.98% for mean downstream velocity and mean WSS, respectively. We found that the VLBM 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 via direct numerical simulation. The computational results of velocity and WSS are again in good agreement with the experimental measurements. We have also presented one application study on the non-invasive quantification of 4-D hemodynamics, including velocity vector and pressure fields and the WSS distribution in MRA-based human carotid and vertebral arteries. The computed velocity agrees reasonably well with the MAR-measured velocity. Three more applications are currently active, including (1) quantification of hemodynamic WSS on image-based human choroidal vasculature endothelium, (2) Drag force on particles in sand flow, and (3) effects of waste streams on ion exchange kinetics in the porous structure. Application (1) is the first-ever computational quantification of WSS in the human choriocapillaris. The findings have been recently submitted for publication<sup>67</sup>. Applications (3) and (4) are in the code adaption stage and results are expected in near future.

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 VLBM 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 VLBM (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.



**Figure 11.** Comparisons of computed (left) vs. MRA-measured (right) velocity contours at (a) Slice 4 of vertebral artery at and (b) Slice 5 of carotid artery. On each panel, the top and bottom rows correspond to the peak systole and the end of diastole in a cardiac cycle.



**Figure 12.** 3-D (a) downstream velocity contours with streamlines, (b) pressure contours, and (c) of the carotid artery at peak systole.

### Data availability

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

Received: 2 April 2021; Accepted: 23 December 2021

Published online: 01 February 2022

## References

1. Powers, K. A. & Dhamoon, A. S. *Physiology, Pulmonary, Ventilation and Perfusion* (2019).
2. Jiang, F., Hirano, T., Ohgi, J. & Chen, X. A voxel image-based pulmonary airflow simulation method with an automatic detection algorithm for airway outlets. *Int. J. Numer. Methods Biomed. Eng.* **36**, e3305 (2020).
3. Lasheras, J. C. The biomechanics of arterial aneurysms. *Annu. Rev. Fluid Mech.* **39**, 293–319 (2007).
4. Van Rijn, L. C. *Principles of Sediment Transport in Rivers, Estuaries and Coastal Sea*. vol. 1006. (Aqua Publications, 1993).
5. Semitsch, B., Wang, Y. & Mihăescu, M. Flow effects due to pulsation in an internal combustion engine exhaust port. *Energy Convers. Manag.* **86**, 520–536 (2014).
6. Winter, D. & Nerem, R. Turbulence in pulsatile flows. *Ann. Biomed. Eng.* **12**, 357–369 (1984).
7. Gundogdu, M. Y., Carpinlioglu, M. O. & Engineering, T. Present state of art on pulsatile flow theory: Part 1: Laminar and transitional flow regimes. *JSME Int. J. Ser. B Fluids* **42**, 384–397 (1999).
8. Reynolds, O. 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. *Philos. Trans. R. Soc. Lond.* **174**, 935–982 (1883).
9. Merkli, P. & Thomann, H. Transition to turbulence in oscillating pipe flow. *J. Fluid Mech.* **68**, 567–576 (1975).
10. Ohmi, M., Iguchi, M. & Urahata, I. Transition to turbulence in a pulsatile pipe flow part 1, wave forms and distribution of pulsatile velocities near transition region. *J. Bull. JSME* **25**, 182–189 (1982).
11. Iguchi, M. & Ohmi, M. Transition to turbulence in a pulsatile pipe flow: Part 2, characteristics of reversing flow accompanied by relaminarization. *J. Bull. JSME* **25**, 1529–1536 (1982).
12. Avila, M., Willis, A. P. & Hof, B. On the transient nature of localized pipe flow turbulence. *J. Fluid Mech.* **646**, 127–136 (2010).
13. Moxey, D. & Barkley, D. Distinct large-scale turbulent-laminar states in transitional pipe flow. *Proc. Natl. Acad. Sci.* **107**, 8091–8096 (2010).
14. Thomas, C., Bassom, A. P., Blennerhassett, P. & Davies, C. The linear stability of oscillatory Poiseuille flow in channels and pipes. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **467**, 2643–2662 (2011).
15. Xu, D. *et al.* Nonlinear hydrodynamic instability and turbulence in pulsatile flow. *Proc. Natl. Acad. Sci.* **117**, 11233–11239 (2020).
16. Xu, D. & Avila, M. The effect of pulsation frequency on transition in pulsatile pipe flow. *J. Fluid Mech.* **857**, 937–951 (2018).
17. Cheng, Z., Jelly, T., Illingworth, S., Marusic, I. & Ooi, A. Forcing frequency effects on turbulence dynamics in pulsatile pipe flow. *Int. J. Heat Fluid Flow* **82**, 108538 (2020).
18. Diagbouga, M. R., Morel, S., Bijlenga, P. & Kwak, B. R. Role of hemodynamics in initiation/growth of intracranial aneurysms. *Eur. J. Clin. Invest.* **48**, e12992 (2018).
19. Gijzen, F. *et al.* Expert recommendations on the assessment of wall shear stress in human coronary arteries: Existing methodologies, technical considerations, and clinical applications. *Eur. Heart J.* **40**, 3421–3433 (2019).
20. Taylor, C. A. & Draney, M. T. Experimental and computational methods in cardiovascular fluid mechanics. *Annu. Rev. Fluid Mech.* **36**, 197–231 (2004).
21. Withey, D. J. & Koles, Z. J. A review of medical image segmentation: Methods and available software. *Int. J. Bioelectromagnet.* **10**, 125–148 (2008).
22. Taylor, C. A. & Figueroa, C. Patient-specific modeling of cardiovascular mechanics. *Annu. Rev. Biomed. Eng.* **11**, 109–134 (2009).
23. Shi, Y., Lawford, P. & Hose, R. Review of zero-D and 1-D models of blood flow in the cardiovascular system. *Biomed. Eng. Online* **10**, 33 (2011).
24. Zhang, J. M. *et al.* Perspective on CFD studies of coronary artery disease lesions and hemodynamics: A review. *Int. J. Numer. Methods Biomed. Eng.* **30**, 659–680 (2014).
25. Marsden, A. L. & Esmaily-Moghadam, M. Multiscale modeling of cardiovascular flows for clinical decision support. *Appl. Mech. Rev.* **67**, 030804 (2015).
26. Morris, P. D. *et al.* Computational fluid dynamics modelling in cardiovascular medicine. *Heart* **102**, 18–28 (2016).
27. Arzani, A. & Shadden, S. C. Wall shear stress fixed points in cardiovascular fluid mechanics. *J. Biomech.* **73**, 145–152 (2018).
28. Yu, H. *et al.* Mass-conserved volumetric lattice Boltzmann method for complex flows with willfully moving boundaries. *Phys. Rev. E* **89**, 063304 (2014).
29. An, S., Yu, H. & Yao, J. GPU-accelerated volumetric lattice Boltzmann method for porous media flow. *J. Petro. Sci. Eng.* **156**, 546–552 (2017).
30. An, S. *et al.* Unified mesoscopic modeling and GPU-accelerated computational method for image-based pore-scale porous media flows. *Int. J. Heat Mass Transf.* **115**, 1192–1202 (2017).
31. Wang, Z., Zhao, Y., Sawchuck, A. P., Dalsing, M. C. & Yu, H. W. GPU acceleration of volumetric lattice Boltzmann method for patient-specific computational hemodynamics. *Comput. Fluids* **115**, 192–200 (2015).
32. Yu, H. W. *et al.* GPU accelerated lattice Boltzmann simulation for rotational turbulence. *Comput. Math. Appl.* **67**, 445–451 (2014).
33. An, S., Zhan, Y., Yao, J., Yu, H. W. & Niasar, V. A greyscale volumetric lattice Boltzmann method for upscaling pore-scale two-phase flow. *Adv. Water Resour.* **144**, 103711 (2020).
34. Chen, R., Shao, J., Zheng, Y., Yu, H. & Xu, Y. Lattice Boltzmann simulation for complex flow in a solar wall. *Commun. Theor. Phys.* **59**, 370–374 (2013).
35. Jiang, F., Tsuji, T. & Hu, C. Elucidating the role of interfacial tension for hydrological properties of two-phase flow in natural sandstone by an improved lattice Boltzmann method. *Transp. Porous Media* **104**, 205–229 (2014).
36. Jiang, F. & Tsuji, T. Estimation of three-phase relative permeability by simulating fluid dynamics directly on rock-microstructure images. *Water Resour. Res.* **53**, 11–32 (2017).
37. Jiang, F., Yang, J., Boek, E. & Tsuji, T. Investigation of viscous coupling effects in three-phase flow by lattice Boltzmann direct simulation and machine learning technique. *Adv. Water Resour.* **147**, 103797 (2021).
38. Chen, R., Yu, H. W., Zeng, J. & Zhu, L. General power-law temporal scaling for unequal-size microbubble coalescence. *Phys. Rev. E* **101**, 023106 (2020).
39. Rou Chen, J. Z. & Yu, H. Mechanism of damped oscillation in microbubble coalescence. *Comput. Fluids* **183**, 113 (2019).
40. Chen, R., Yu, W., Xu, Y. & Zhu, L. Scalings of inverse energy transfer and energy decay in 3-D decaying isotropic turbulence with non-rotating or rotating frame of reference. *J. Appl. Comput. Mech.* **5**, 639–646 (2019).
41. Chen, R., Yu, H. & Zhu, L. Effects of initial conditions on the coalescence of micro-bubbles. *Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci.* **232**, 457–465 (2018).
42. Chen, R., Yu, H., Zhu, L., Taehun, L. & Patil, R. Spatial and temporal scaling of unequal microbubble coalescence. *AIChE J.* **63**, 1441–1450 (2017).
43. Yu, H., Li, N. & Ecke, R. E. Scaling in laminar natural convection in laterally heated cavities: Is turbulence essential in the classical scaling of heat transfer?. *Phys. Rev. E* **76**, 026303 (2007).
44. Yu, H., Girimaji, S. S. & Luo, L. S. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *J. Comput. Phys.* **209**, 599–616 (2005).
45. Yu, H., Girimaji, S. S. & Luo, L. S. Lattice Boltzmann simulations of decaying homogeneous isotropic turbulence. *Phys. Rev. E* **71**, 016708 (2005).
46. Yu, H. & Girimaji, S. S. Near-field turbulent simulations of rectangular jets using lattice Boltzmann method. *Phys. Fluids* **17**, 125106 (2005).

47. Yu, H., Luo, L. S. & Girimaji, S. S. Scalar mixing and chemical reaction simulations using lattice Boltzmann method. *Int. J. Comput. Eng. Sci.* **03**, 73–87 (2002).
48. Jiang, F., Matsumura, K., Ohgi, J. & Chen, X. A GPU-accelerated fluid–structure-interaction solver developed by coupling finite element and lattice Boltzmann methods. *Comput. Phys. Commun.* **259**, 107661 (2021).
49. Wang, Z., Zhao, Y., Yu, H., Lin, C. & Sawchuck, A. P. Fully parallelized Lattice Boltzmann scheme for fast extraction of biomedical geometry. *J. Parallel Distrib. Comput.* **128**, 126–136 (2019).
50. Domb, C. *Phase Transitions and Critical Phenomena*. Vol. 19. iii 7–40. (Academic Press, 2000).
51. Kamio, T., Suzuki, M., Asaumi, R. & Kawai, T. *DICOM Segmentation and STL Creation for 3D Printing: A Process and Software Package Comparison for Osseous Anatomy* (2020).
52. Bhatnagar, P. L., Gross, E. P. & Krook, M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.* **94**, 511–525 (1954).
53. Luo, L.-S. Theory of the lattice Boltzmann method: Lattice Boltzmann models for nonideal gases. *Phys. Rev. E* **62**, 4982 (2000).
54. Osher, S. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79**, 12–49 (1988).
55. *READ\_stl.m*, *VOXELISE.m*. [https://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation?s\\_tid=mwa\\_osa\\_a](https://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation?s_tid=mwa_osa_a).
56. *ac-reinit.m*. [https://www.mathworks.com/matlabcentral/fileexchange/24998-2d-3d-image-segmentation-toolbox?focused=3775789&s\\_tid=gn\\_loc\\_drop&tab=function](https://www.mathworks.com/matlabcentral/fileexchange/24998-2d-3d-image-segmentation-toolbox?focused=3775789&s_tid=gn_loc_drop&tab=function).
57. Stahl, B. & Chopard, B. Measurements of wall shear stress with the lattice Boltzmann method and staircase approximation of boundaries. *J. Latt. Fluids* **39**, 1625–1633 (2010).
58. Ferziger, J. H., Perić, M. & Street, R. L. *Computational Methods for Fluid Dynamics* Vol. 3 (Springer, 2002).
59. Womersley, J. R. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *J. Physiol.* **127**, 553 (1955).
60. Gomez, J., Yu, H., & Andreopoulos, Y. The role of flow reversals in transition to turbulence and relaminarization of pulsatile flows. *J. Fluid Dyn.* **917**, A27 (2021).
61. Guo, Z. L. & Zheng, C. *Theory and Applications of Lattice Boltzmann Method* (Science, 2009).
62. Gijzen, F., van der Giessen, A., van der Steen, A. & Wentzel, J. Shear stress and advanced atherosclerosis in human coronary arteries. *J. Biomech.* **46**, 240–247 (2013).
63. Cito, S., Mazzeo, M. D. & Badimon, L. A review of macroscopic thrombus modeling methods. *Thromb. Res.* **131**, 116–124 (2013).
64. Roy, D. *et al.* A literature review of the numerical analysis of abdominal aortic aneurysms treated with endovascular stent grafts. *Comput. Math. Methods Med.* (2012).
65. Coveney, P. V. & Shublaq, N. W. Computational biomedicine: A challenge for the twenty-first century. *Stud. Health Technol. Inform.* **174**, 105–110 (2012).
66. Wang, Z. *et al.* Using flow feature to extract pulsatile blood flow from 4D flow MRI images. in *Proceedings of the SPIE 10133, Medical Imaging 2017: Image Processing*. 101331O–1 (2017).
67. An, S. *et al.* Hemodynamic distribution in human choriocapillaris from donor specific and image-based computational hemodynamics. in *Proceedings of the National Academy of Sciences of the United States of America* (2021) submitted to *Investigative Ophthalmology & Visual Science*.

## Acknowledgements

The research is supported by NSF through grant CBET 1803845 and Indiana University Health Value Fund. 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.

## Author contributions

X.Z. did the modeling, simulation, verification, and validation, provided the raw data for Figs. 6, 7, 8 and Table 1, revised the manuscript. She prepared the new results for addressing reviewers' comments. J.G. did the experimental measurements for the 5 turbulent pulsatile flows and provided the data for validation in Fig. 8. X.C. produced all results in “Application study” section. J.M.M. revised the manuscript twice. M.M.I. analytically studied the Womersley flow and provided insights for the verification. Y.A. supervised J.G. experiment and wrote the experimental part. L.Z. contributed the GPU parallel computing resource for this work. H.Y. supervised X.Z. and X.C. for the results of the paper. She composed, revised, and submitted the manuscripts.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to H.Y.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022