Choice of boundary condition for lattice-Boltzmann simulation of moderate-Reynolds-number flow in complex domains

Rupert W. Nash,[1,*] Hywel B. Carver,[1,2] Miguel O. Bernabeu,[1,2] James Hetherington,[1,3]
Derek Groen,[1] Timm Krüger,[1,4] and Peter V. Coveney[1,†]

[1]*Centre for Computational Science, Department of Chemistry, University College London,*
20 Gordon Street, London, WC1H 0AJ, United Kingdom
[2]*CoMPLEX, University College London, Physics Building, Gower Street, London, WC1E 6BT, United Kingdom*
[3]*Research Software Development Team, Research Computing and Facilitating Services, University College London,*
Podium Building - 1st Floor, Gower Street, London, WC1E 6BT, United Kingdom
[4]*Institute for Materials and Processes, School of Engineering, University of Edinburgh, King's Buildings,*
Mayfield Road, Edinburgh, EH9 3JL, United Kingdom

Modeling blood flow in larger vessels using lattice-Boltzmann methods comes with a challenging set of constraints: a complex geometry with walls and inlets and outlets at arbitrary orientations with respect to the lattice, intermediate Reynolds (Re) number, and unsteady flow. Simple bounce-back is one of the most commonly used, simplest, and most computationally efficient boundary conditions, but many others have been proposed. We implement three other methods applicable to complex geometries [Guo, Zheng, and Shi, Phys. Fluids **14**, 2007 (2002); Bouzidi, Firdaouss, and Lallemand, Phys. Fluids **13**, 3452 (2001); Junk and Yang, Phys. Rev. E **72**, 066701 (2005)] in our open-source application HEMELB. We use these to simulate Poiseuille and Womersley flows in a cylindrical pipe with an arbitrary orientation at physiologically relevant Re number (1–300) and Womersley (4–12) numbers and steady flow in a curved pipe at relevant Dean number (100–200) and compare the accuracy to analytical solutions. We find that both the Bouzidi-Firdaouss-Lallemand (BFL) and Guo-Zheng-Shi (GZS) methods give second-order convergence in space while simple bounce-back degrades to first order. The BFL method appears to perform better than GZS in unsteady flows and is significantly less computationally expensive. The Junk-Yang method shows poor stability at larger Re number and so cannot be recommended here. The choice of collision operator (lattice Bhatnagar-Gross-Krook vs multiple relaxation time) and velocity set (D3Q15 vs D3Q19 vs D3Q27) does not significantly affect the accuracy in the problems studied.

I. INTRODUCTION

In the last two decades, lattice-Boltzmann methods (LBM) [1–3] have been widely studied and used for fluid flow problems. We are particularly interested in its application to the study of hemodynamics: the flow behavior of blood under physiological conditions. Accurate simulation of the flow of blood in an individual could have near-term clinical benefits for, *inter alia*, the treatment of aneurysms [4–6] or stenoses [7–9].

Applications such as these have a number of challenges [retaining computational performance in a relatively sparse, three-dimensional (3D) fluid domain; capturing the complex rheology of a dense suspension of deformable particles; accounting for the compliance of vessel walls] but here we address the choice of boundary condition method in complex domains. We will not examine the rheology and compliance problems in this article, restricting ourselves to a Newto-nian fluid within a rigid-walled system, but instead aim to provide recommendations for the choice of boundary condition method(s) for a complex geometry. This choice is examined in the context of two other variables: the discrete velocity set and lattice-Boltzmann collision operator. Bearing in mind recent controversy over the reproducibility of computational science [10,11], we have released the source code of our application HEMELB for the community's scrutiny and use.

Lätt *et al.* [12] considered five boundary conditions and assessed their accuracy. However, they studied only boundary conditions in which the wall passes through a lattice point, immediately restricting their results to boundaries that are normal to one of the Cartesian directions of the underlying grid. Boyd *et al.* [13] compared simple bounce-back to the Guo-Zheng-Shi method [14] in simulations of arterial bifurcations, finding that the two methods give different results when stenosis is present, but without any analysis suggesting which, if either, is more accurate. Stahl *et al.* [15] performed LBM simulations with simple bounce-back boundary conditions to examine the effect of staircased boundaries (where walls that are not aligned with the underlying grid are approximated by a set of grid edges) on the measurements of shear stress both at the wall and in the bulk flow, finding errors in the shear stress of up to 35% for two-dimensional (2D) channel flow and 28% for 2D bent-pipe flow. They also introduced a method for estimating the local wall normal from the shear stress tensor. The endothelial shear stresses in large arterial trees were studied by Melchionna *et al.* [16] with their code

---

*Present address: EPCC, University of Edinburgh, JCMB, The Kings Buildings, Mayfield Road, Edinburgh EH9 3JZ, United Kingdom; rupert.nash@ed.ac.uk

†p.v.coveney@ucl.ac.uk

MUPHY [17]. Pan *et al*. [18] have studied the effect of different boundary conditions in porous media for flows at low Reynolds (Re) number, concluding that interpolation at the boundaries significantly improves accuracy.

In this paper we comprehensively compare the accuracy of simulations performed with some of the most widely used solid-fluid boundary conditions against analytical solutions relevant for simulations of blood flow in the larger vessels. To the best of our knowledge, this is the first like-for-like comparison of the popular boundary conditions for lattice Boltzmann in geometries and under conditions relevant for hemodynamics and other moderate-Re-number problems. Other comparison studies cover this case, instead focusing on straight boundaries [12] and porous media [18].

The outline of the paper is as follows. In Sec. II we briefly introduce the lattice-Boltzmann method and discuss the collision operators, velocity sets, and boundary conditions used. Our open-source software [19] and simulation protocols are described, and results given in Sec. III. We present our conclusions in Sec. IV.

II. THE LATTICE-BOLTZMANN METHOD

Here we provide a brief summary of the lattice-Boltzmann method (LBM); for a full derivation, we refer the reader to one of the many thorough descriptions available [1–3]. LBM operates at a mesoscopic level, storing a discrete-velocity approximation to the one-particle distribution functions of the Boltzmann equation of kinetic theory [20], $\{f_i(\mathbf{r},t)\}$, on a regular lattice, with grid spacing Δx. The set of velocities $\{\mathbf{c}_i\}$ is chosen such that the distances traveled in one time step (Δt), $\mathbf{e}_i = \mathbf{c}_i \Delta t$, are lattice vectors and to ensure Galilean invariance [21]. When one only wishes to reproduce Navier-Stokes dynamics, the set is typically a subset of the Moore neighborhood, including the rest vector. For 3D simulations, the most commonly used sets have 15, 19, and 27 members (termed D3Q15, D3Q19, and D3Q27, respectively). The LBM can be shown, through a Chapman-Enskog expansion, to reproduce the Navier-Stokes equations in the quasi-incompressible limit with errors proportional to the Mach (Ma) number squared.

In the absence of forces, the density $\rho(\mathbf{r},t)$ and the velocity $\mathbf{u}(\mathbf{r},t)$ at a fluid site can be calculated from the distributions by

$$\rho = \sum_i f_i, \tag{1}$$

$$\rho\mathbf{u} = \sum_i f_i \mathbf{c}_i. \tag{2}$$

Advancing the system one time step is conceptually divided into two stages. The first is collision, which relaxes the distributions towards a local equilibrium (we denote the postcollisional distributions as $f_i^\star$):

$$f_i^\star(\mathbf{r},t) = f_i(\mathbf{r},t) + \hat{\Omega}(f_i(\mathbf{r},t)), \tag{3}$$

where $\hat{\Omega}$ is the collision operator. The second is streaming, where the postcollisional distributions are propagated along the lattice vectors to new locations in the lattice, defining the distributions of the next time step:

$$f_i(\mathbf{r} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i^\star(\mathbf{r},t). \tag{4}$$

A. Collision operators

The collision operator approximates the microscopic interparticle interactions. Here we summarize the two considered in this article: lattice Bhatnagar-Gross-Krook and multiple relaxation time.

Lattice Bhatnagar-Gross-Krook (LBGK) [21,22] approximates the collision process as relaxation towards a local equilibrium (see below), in a discrete velocity analog of the Boltzmann-BGK approximation from kinetic theory [23],

$$\hat{\Omega}(f_i) = -\frac{\left(f_i - f_i^{\text{eq}}\right)}{\tau}\Delta t, \tag{5}$$

where τ is the relaxation time. In this case all modes relax towards equilibrium at the same rate. This can be shown, through a Chapman-Enskog expansion (see, e.g., [2,24]), to reproduce the Navier-Stokes equations with a kinematic viscosity ν given by

$$\nu = c_{\text{s}}^2(\tau - \Delta t/2). \tag{6}$$

For the equilibrium distribution, we use a second-order (in velocity space) approximation to a Maxwellian distribution,

$$f_i^{\text{eq}}(\rho,\mathbf{u}) = \rho w_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_{\text{s}}^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_{\text{s}}^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_{\text{s}}^2}\right), \tag{7}$$

where the weights w_i and speed of sound c_{s} depend on the choice of velocity set. LBGK is simple to implement and gives excellent performance; it is therefore widely used.

Multiple relaxation time (MRT) collision operators, developed at the same time as LBGK [25], generalize the notion of relaxation towards local equilibrium (the same as above) by allowing different relaxation rates for different moments of the distributions, potentially improving stability properties and accuracy [26]. The eigenvalue of the collision matrix which corresponds to the relaxation of shear stress λ_{shear} determines the viscosity as in (6) ($\tau \to 1/\lambda_{\text{shear}}$). We use the MRT operator on the D3Q15 and D3Q19 lattices, as described by d'Humières *et al*. [26]. Here, the distribution function f_i are transformed into the moment basis m_i by the matrix M_{ij} and the different moments can be relaxed towards equilibrium at different rates, before being projected back into the distribution space for advection,

$$\hat{\Omega}(f_i) = -\sum_j M_{ij}^{-1} s_j\big(m_j - m_j^{\text{eq}}\big), \tag{8}$$

where s_j is the relaxation rate for mode j.

B. No-slip boundary conditions

Boundary conditions for LBM have some additional complications compared to boundary conditions for Navier-Stokes-based methods, due to the LBM's mesoscopic nature. Typically, one wishes to impose conditions on the macroscopic, hydrodynamic variables (p, $\mathbf{u}$) but these must be implemented through a closure relation for the mesoscopic distributions. There is no single, obviously superior choice. In this section, we will briefly review some commonly used boundary condition methods for lattice-Boltzmann models which impose the no-slip condition that the velocity of fluid adjacent to a wall is equal to the velocity of the wall.

Many boundary condition methods do not vary their behavior with respect to the location of the walls in relation to the Eulerian grid. The wall is often assumed to pass infinitesimally close to a point, the grid point remaining inside the fluid domain (sometimes referred to as "wet node" boundary conditions [12]), or alternatively the boundary is considered to be halfway along the lattice vector to a point outside the fluid. In the case of complex or non-lattice-aligned domains, these methods (and simple bounce-back; see below) will always cause a first-order *modeling* error, irrespective of the order of numerical accuracy of the resulting lattice-Boltzmann method. (Typically second-order accuracy is sought, since this is the inherent accuracy of standard lattice-Boltzmann methods in bulk fluids.) This point has been studied numerically by Stahl *et al.* [15]. Other boundary conditions allow the wall to be at an arbitrary position along the link between a solid and a fluid site. These reduce the modeling error of fixed wall position methods, but often at the price of increased complexity and/or the requirement of data from neighboring fluid lattice points. Furthermore, a number of methods suffer reduced accuracy at sites in corners, which can reduce the accuracy of simulations throughout the domain.

There are a number of popular methods [27–30] that can only operate on straight, axis-aligned planes and force the boundary to pass directly through the lattice point. Malaspinas *et al.* [31] generalized the regularized method [29] to cope accurately with corner nodes. The authors acknowledge that it fails for the case of a D3Q15 lattice and a right-angled corner and this method also forces the boundary to pass through a lattice point. These methods are, therefore, unsuitable for problems involving complex boundaries.

Simple bounce-back (SBB) is perhaps the most widely used boundary condition for solid walls, positioning them halfway along the lattice vector from fluid to solid. It is straightforward to implement and gives second-order accurate simulation results for flat boundaries aligned with the Cartesian axes of the lattice [32], although in more complex cases this degrades to first-order accuracy [33]. It is also computationally cheap and local in its operation. SBB ensures conservation of mass up to machine precision. In this work we use the halfway bounce-back scheme [34]. Despite SBB exhibiting the modeling error mentioned above, we include it in this study due to its wide use.

The Bouzidi-Firdaouss-Lallemand (BFL) method [35] starts with simple bounce-back, but interpolates the value of the to-be-propagated distribution with the distribution at the fluid site which standard bounce-back would stream it to. They present two variants: one using linear interpolation and another using quadratic interpolation. In the present work, we restrict our attention to the linear case, due to its locality and smaller communication requirement (indeed, it can be implemented without any interprocess communication above the normal lattice-Boltzmann streaming step, albeit at a price of revisiting the sites adjacent to the wall). Bouzidi *et al.* claim that both variants show second-order convergence, but with the linear method having a poorer prefactor [35]. The method as presented by Bouzidi *et al.* may, depending upon the distance of the wall from the fluid site, fail when computing $f_i(\mathbf{x}, t + \Delta t)$ where $\mathbf{x} - \mathbf{e}_i$ lies outside the fluid domain, if the site at $\mathbf{x} + \mathbf{e}_i$ is also outside. Since this can occur in a curved domain, in these cases we fall back to performing SBB. While this does degrade accuracy slightly, it does appear to offer good stability of the simulation.

Guo, Zheng, and Shi (GZS) [14] present a boundary condition which decomposes the unknown distributions at the wall into equilibrium and nonequilibrium parts. The equilibrium part uses the density of the fluid site and a linearly extrapolated velocity such that the velocity at the solid wall is as imposed. For the nonequilibrium part, the value from the fluid site (or the next site into the bulk) is used. The method as described in [14] has the same problem as BFL. We again fall back to SBB when this occurs.

Junk and Yang (JY) [36] propose a correction to the simple bounce-back method. They claim an advantage compared to interpolation-based methods such as GZS and BFL as their method is completely local and is able to handle nonstraight boundaries where sites have lattice vectors in opposite directions, both crossing the solid-fluid boundary. The method arises from their analysis [37] of boundary conditions for LBM in terms of general methods for studying finite difference schemes rather than the standard Chapman-Enskog expansion. By adding correction terms to the collision operator and then discretizing them in an optimal (under their analysis framework) manner, they ensure the Navier-Stokes equations are obeyed with the correct boundary conditions.

The multireflection method by Ginzburg and d'Humières [38] uses linear combination of five neighboring distribution functions along a link direction plus a correction to determine the unknown direction. This combination is obtained from a Chapman-Enskog expansion at the boundary sites, which is third-order accurate for steady flows. Chun and Ladd [39] present a method based only on interpolation of the equilibrium part of the distribution functions, relying on the fact that the nonequilibrium part is always one order higher in the Chapman-Enskog expansion. Chun and Ladd demonstrate numerically that their method is advantageous for problems with many small gaps, such as porous media.

It is well known that LBGK combined with SBB does not locate the plane of zero velocity exactly halfway along the link for flows with varying velocity gradients [40] and that this can be a significant issue in, for example, porous media [18]. The effective width of the channel in lattice units H, for a Poiseuille flow in a 2D lattice-aligned channel, driven by a constant body force is given by [Eq. (19) from [41]]

$$H^2 = N_x^2 + 48\nu^4 - 1, \tag{9}$$

where N_x is the number of lattice points across the channel and ν is the viscosity in lattice units. While this error can be reduced by optimizing the choice of MRT relaxation times [42], it can only reach zero for particularly simple cases. Typically at Re $\geqslant 1$ the larger system sizes and smaller relaxation times combine to reduce the relative error. For example, with $N_x = 40$ and $\tau = 0.55$ the effective width is 39.987, a relative error of 0.03%. Due to the smallness of these errors in the problems studied here, we do not consider them further.

We also note that some authors propose the use of grid refinement [43], finite difference [44], and finite volume [45,46] discretizations of the discrete Boltzmann equation as methods for improving accuracy around nonplanar boundaries, but we will restrict our attention here to implementations on

a single, regular grid. In the case of moving boundaries, a re-filling procedure [47] can be used. Additionally, the immersed boundary method (IBM) [48] has been used in conjunction with the LBM to simulate rigid [49] and deformable [50] boundaries. IBM requires a further layer of fluid sites outside the walls as well as another, simpler boundary condition at the edge of the halo region, but admits an extension to moving boundaries in a natural way.

C. Open boundary conditions

For inlets, we use Ladd's method [34] to impose the expected velocity profile (parabolic, Womersley). This is a modification of SBB, with a correction term $-2w_i \rho \mathbf{u} \cdot \mathbf{c}_i / c_s^2$, where $\mathbf{u}$ is the imposed velocity at the halfway point of the link. However, using this at outlets as well will cause the total mass in the system to increase unboundedly, due to the unbalanced mass fluxes into and out from the system since LBM has a finite compressibility [51].

Alternatively, one could impose mixed Dirichlet-Neumann boundary conditions at the outlet:

$$p = p_0, \tag{10a}$$

$$\mathbf{u}_\parallel = 0, \tag{10b}$$

$$\hat{\mathbf{n}} \cdot \nabla u_\perp = 0, \tag{10c}$$

where $\hat{\mathbf{n}}$ is the inward pointing normal of the open boundary.

A number of authors [28,52–54] have proposed open boundary conditions that fulfill some or all of these requirements, however, the techniques cited are only suitable for inlets aligned with the lattice's axes. We have therefore developed a simple method that meets these requirements.

Assume that, at the start of a time step, all distributions for an inlet or outlet site are known. LBM then proceeds as normal: a collision step, followed by a streaming step; the distributions that would have come from exterior sites now have an undefined value. We close the system by constructing a "phantom site" (indicated below with a subscript $\mathcal{P}$) beyond the boundary, whose hydrodynamic variables are estimated based on the imposed values and those at the inlet or outlet site (indicated with subscript $\mathcal{I}$). Note that there is one phantom site per missing distribution, i.e., the phantom sites of adjacent inlet or outlet sites are unrelated. This is to eliminate the need for communication between neighboring sites. The equilibrium distribution for the missing direction is computed at the phantom site and then streamed into place. For the density, we assume the target pressure p_0, i.e., we perform a zero-order extrapolation from the outlet plane. Condition (10b) is enforced by projecting away any velocity component not parallel to the inlet or outlet normal, $\hat{\mathbf{n}}$. For condition (10c), we take first-order finite-difference approximations for the derivatives, giving

$$\hat{\mathbf{n}} \cdot \nabla u_\perp \approx \frac{u_\perp(\mathbf{r}_\mathcal{I}, t) - u_\perp(\mathbf{r}_\mathcal{P}, t)}{c_{i\alpha} \cdot \hat{\mathbf{n}} \Delta t} = 0. \tag{11}$$

Hence

$$\mathbf{u}(\mathbf{r}_\mathcal{P}, t) \approx (\mathbf{u}(\mathbf{r}_\mathcal{I}, t) \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}. \tag{12}$$

For lattice sites that are adjacent to both open and closed boundaries, we use the above method on those links that cross the inlet or outlet and the solid wall boundary condition on those links that cross the solid wall of the domain.

III. SIMULATIONS

Our goal is to determine which combination of boundary condition, collision operator, and velocity set gives the best all-round accuracy in a nontrivial geometry (i.e., with curved surfaces), with a focus on computational hemodynamics. We also assess the computational requirements of the different models.

We compare against analytical solutions, which restricts us to relatively simple domains: We choose a cylinder and a torus. For the cylinder we use both steady, Hagen-Poiseuille flow and a time-dependent, sinusoidal Womersley flow. For the torus, we use only steady flow. By choosing a non-axis-aligned orientation for the cylinder, we better mimic a typical production simulation of the human vasculature. The orientation $\hat{\mathbf{n}}$ was chosen pseudorandomly from the unit sphere, subject to the constraint that $\hat{\mathbf{n}} \cdot \hat{\mathbf{e}}_i \leqslant 0.9, \forall i$. The value is

$$\hat{\mathbf{n}} = [-0.299, 0.382, 0.874]. \tag{13}$$

Our approach is to select parameters in lattice units, but with physiologically relevant Re and Womersley (α) numbers.

A. Software: HEMELB

The simulations in this paper were performed with HEMELB [55], a lattice-Boltzmann-based fluids solver, which includes capability for *in situ* imaging of flow-fields and real-time steering [56]. It is a distributed memory application, parallelized with MPI. We have shown that HEMELB's computational performance scales linearly up to at least 32 768 cores [57]. We have released the software online [19], under the open-source GNU Lesser General Public License (LGPL), to enable interested researchers to reproduce our results as well as to use the software for novel problems.

HEMELB has several linked components, described in Groen *et al.* [57]. We have recently redeveloped the lattice-Boltzmann core to allow for easy switching between use of different velocity sets, collision operators, and boundary conditions, through a statically polymorphic, object-oriented design. This avoids any run-time overhead due to dynamic polymorphism [58]. The individual software components are tested through a battery of over 100 unit and regression tests, which are run nightly by our continuous integration server.

HEMELB includes, among other features, the following: the D3Q15, D3Q19, and D3Q27 velocity sets; the lattice Bhatnagar-Gross-Krook (LBGK) and multiple relaxation time (MRT) collision operators; and, the simple bounce-back (SBB), Guo-Zheng-Shi (GZS), Bouzidi-Firdaouss-Lallemand (BFL), and Junk-Yang (JY) boundary condition methods. HEMELB does not currently support the combination of the GZS boundary condition with the MRT collision operator; this will be addressed in a future release.

The software includes a separate tool for defining the simulation domain. This requires either a geometric primitive or a general surface, meshed with triangles. The user can then place inlets and outlets, specify their pressure and/or velocity boundary conditions, and select the fineness of the lattice. This setup tool then generates the input for HEMELB itself, producing

a description of each fluid site and, if needed, the location of the wall. For the cylinders used in this work, we directly use the cylinder, rather than approximating it with triangles.

B. Convergence analysis

In this section we report on a series of simulations of Hagen-Poiseuille flow over a range of resolutions and Re numbers, defined here as

$$\text{Re} = \frac{U_{\max} D}{\nu}, \tag{14}$$

where $U_{\max}$ is the maximum velocity, D the pipe diameter, and ν the kinematic viscosity. The velocity $\mathbf{U}$ is

$$\mathbf{U} = \frac{\partial_n p}{4\rho\nu}\left(\left(\frac{D}{2}\right)^2 - r^2\right)\hat{\mathbf{n}}, \tag{15}$$

where r is the distance from the cylinder axis, defined by $\hat{\mathbf{n}}$, and $\partial_n p$ is the pressure gradient along the axis.

In Table I we list all the parameters chosen for the simulations. The range of Re spans typical values for cerebral arteries in the human body [59]. For each case we vary the diameter D from 12 to 192 lattice units; the length of tube used is given by $L = 4D$. Due to the finite speed of sound in LBM ($c_s = 1/\sqrt{3}$ for the models used here), we list the Ma numbers (Ma $\equiv U_{\max}/c_s$); the lowest resolution simulations in each have extremely high values and will consequently have poor accuracy, but this allows us to assess the convergence behavior at modest computational expense. Next we show the value of the LBGK relaxation time τ which must be greater than $\Delta t/2$ [1] and not be much greater than Δt [60]. For the MRT simulations, we use the parameters from [26] except for the stress relaxation rate where we use $s_9 = s_{11} = 1/\tau$, i.e., $s_1 = 1.6\Delta t$, $s_2 = 1.2\Delta t$, $s_4 = 1.6\Delta t$, and $s_{14} = 1.2\Delta t$.

TABLE I. Parameters for convergence analysis. All values are given in lattice units. Parameters are, from left to right, as follows: the Re number; the cylinder diameter D; the predicted Ma number; the LBGK relaxation time τ; the relative density difference imposed $\Delta\rho/\rho_0$; the momentum diffusion time $T_{\text{mom}} \equiv D^2/\nu$, and the sound propagation time $T_s \equiv L/c_s$.

Re	D	Ma	τ	$\Delta\rho/\rho_0$	T_{mom}	T_s
1	12	0.0241	1.00	0.01850	864	42
1	24	0.0120	1.00	0.00463	3460	83
1	48	0.0060	1.00	0.00116	13 800	166
1	96	0.0030	1.00	0.00029	55 300	333
30	12	0.3610	0.75	0.13900	1730	42
30	24	0.1800	0.75	0.03470	6910	83
30	48	0.0902	0.75	0.00868	27 600	166
30	96	0.0451	0.75	0.00217	111 000	333
100	12	0.4810	0.60	0.07410	4320	42
100	24	0.2410	0.60	0.01850	17 300	83
100	48	0.1200	0.60	0.00463	69 100	166
100	96	0.0601	0.60	0.00116	276 000	333
300	12	0.7220	0.55	0.05560	8640	42
300	24	0.3610	0.55	0.01390	34 600	83
300	48	0.1800	0.55	0.00347	138 000	166
300	96	0.0902	0.55	0.00087	553 000	333

We hold τ constant in lattice units while refining the spatial resolution, which implies diffusive scaling of the time step (when converted to physical units), i.e., $\Delta t \propto \Delta x^2$. The density, and hence pressure, difference $\Delta\rho$ driving the flow is also reported; this must remain much less than the reference density of the simulation ρ_0 in order to keep compressibility errors small. Finally, we list the time for momentum to diffuse across the cylinder's diameter, $T_{\text{mom}} \equiv D^2/\nu$, and the time for a sound wave to propagate the length of the cylinder, $T_s \equiv L/c_s$, to give some idea of the time required for the simulation to converge to a steady state. To determine whether a simulation has indeed converged, we compute the maximum difference between flow fields at two times:

$$\Delta u(t_1, t_2) = \frac{\max_{\mathbf{r}} \|\mathbf{u}(\mathbf{r}, t_1) - \mathbf{u}(\mathbf{r}, t_2)\|}{U_{\max}}, \tag{16}$$

and require that $\Delta u(t, t+1) < 10^{-7}$.

We use a simple initialization procedure, initializing to a uniform density fluid at rest. This approach is general and can be applied to any geometry without requiring any preprocessing step [61–63], but does require longer simulation times until a steady solution is reached. For each simulation we compare the Poiseuille solution, $\mathbf{U}(\mathbf{r})$, with the velocity field found by simulation, $\mathbf{u}(\mathbf{r}, t)$. We define the velocity error as

$$\mathbf{u}^\star(\mathbf{r}, t) \equiv \mathbf{u}(\mathbf{r}, t) - \mathbf{U}(\mathbf{r}), \tag{17}$$

and use the ℓ^2 norm scaled by the predicted velocity range as our measure of error:

$$\epsilon_u^2(t) = \frac{\sqrt{\sum_{\mathbf{r}} \mathbf{u}^{\star 2}}}{\sqrt{N} \max_{\mathbf{r}} \|\mathbf{U}\|}. \tag{18}$$

These are evaluated over all fluid sites in the central 90% of the cylinder, thus excluding the inlet and outlet sites. For the pressure gradient, we use the measured difference at the edge of this volume divided by the distance between the two planes. This is to disentangle the errors due to the open boundary condition method from the no-slip condition. We will return in the future to a full validation of the open boundary condition method.

The simulations were performed on HECToR, the United Kingdom national supercomputer, using up to 544 cores. The number of cores for each run was chosen to minimize the run time while remaining efficient which we have shown to occur at around 10^3 sites per core [57].

The Junk-Yang method showed poor stability (instability being defined as one of the distribution functions becoming negative) and only producing a converged solution for the lowest resolution cases at low Re number. We therefore disregard the JY method from further consideration. We do, however, have some confidence in our implementation of the method as our unit test suite demonstrates (data not shown) that the implementation reduces to SBB when the wall is a plane normal to one of the axes and halfway between the site and its nonfluid neighbors, as expected [36].

In Fig. 1 we show the velocity errors as a function of increasing resolution for the remaining three boundary condition methods and the three velocity sets. We show only the results for the LBGK collision operator (the results with MRT are visually indistinguishable).

The results clearly show that SBB gives the expected first-order convergence as the resolution increases, while BFL and
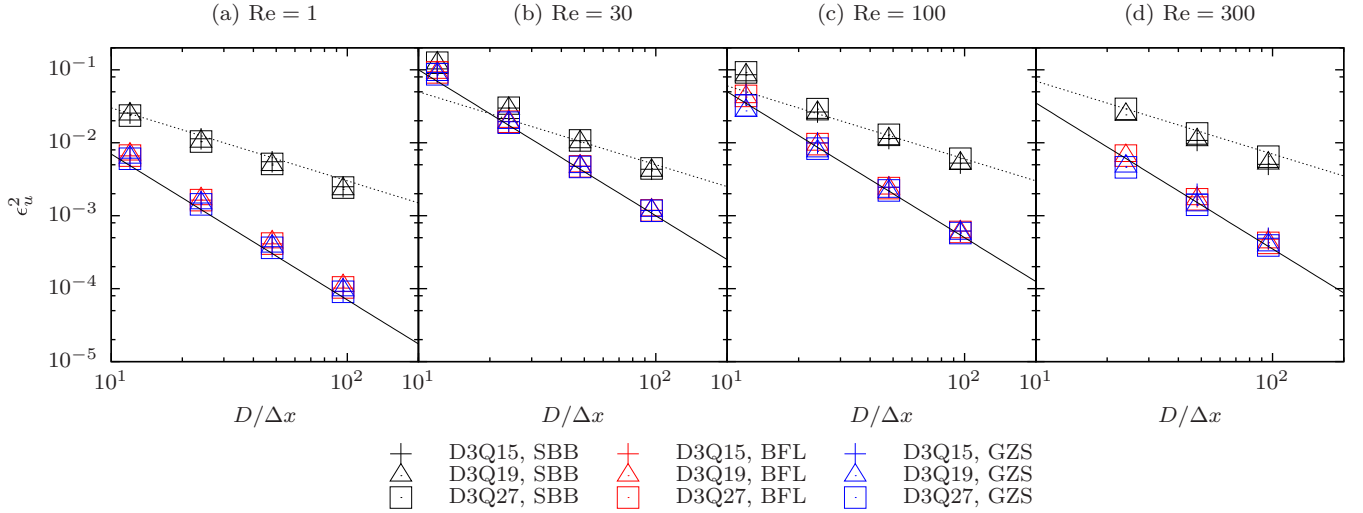
| (a) Re = 1 | (b) Re = 30 | (c) Re = 100 | (d) Re = 300 |

+	D3Q15, SBB	+	D3Q15, BFL	+	D3Q15, GZS
△	D3Q19, SBB	△	D3Q19, BFL	△	D3Q19, GZS
□	D3Q27, SBB	□	D3Q27, BFL	□	D3Q27, GZS

FIG. 1. (Color online) Convergence of velocity error residuals (ϵ_u^2) as defined in Eq. (18) vs diameter in lattice units. (a) Re number Re = 1; (b) Re = 30; (c) Re = 100; (d) Re = 300. Colors (black, red, blue) indicate boundary condition (SBB, BFL, GZS) and shapes (crosses, triangles, squares) indicate the velocity set (D3Q15, D3Q19, D3Q27). The solid lines are guides to the eye showing first-order (dashed) and second-order (solid) convergence. The simulations with Re = 300 and $D = 12\Delta x$ became unstable due to the high Ma number and are not shown.

GZS show second-order convergence with similar prefactors. GZS offers slightly superior accuracy than BFL, reaching up to 20% lower errors (Re = 300) but the relative benefit is highly variable, vanishing in some cases.

As the Re number is increased, we note that the errors also increase, particularly when going from Re = 1 to 30. This is likely due to the increase in the Ma number. The small improvement from Re = 30 to 300 is probably due to the decreasing density difference and hence reduced compressibilty errors.

The velocity set has a small effect on the measured accuracy: D3Q15 and D3Q19 are almost coincident. D3Q27 does offer a small benefit of 1%–4% for GZS and BFL, but worsens accuracy for SBB; this may be due to the greater number of sites which have to implement the boundary condition and therefore suffer from the first-order modeling error.

C. Womersley flow

Here we report on simulations of oscillatory flow, in order to explore the different boundary condition methods' effect on accuracy for time-dependent cases. The Womersley number (α) is a dimensionless number governing dynamical similarity in cases of oscillatory flow. It relates to the ratio of transient forces to viscous forces (or alternatively the ratio of diameter to boundary layer growth during one period of oscillation). In the case of a cylinder (radius R, axis $\hat{\mathbf{n}}$) and laminar flow with zero average pressure gradient [$\partial_n p(t) = (A/L)\sin\omega t$, $\omega \equiv 2\pi/T_{osc}$], it is defined as

$$\alpha = R\sqrt{\frac{\omega}{\nu}}, \tag{19}$$

where ν is the fluid viscosity. The time-dependent Navier-Stokes equations admit an analytic solution [64]:

$$\mathbf{u}(r,t) = -\text{Re}\left(\frac{A}{\omega\rho L}\left[1 - \frac{J_0(i^{3/2}\alpha\frac{r}{R})}{J_0(i^{3/2}\alpha)}\right]e^{i\omega t}\right)\hat{\mathbf{n}}, \tag{20}$$

where J_0 is the order-0 Bessel function of the first kind and Re(z) gives the real part of z.

In the larger arteries of the human body, Womersley numbers range approximately between 4 and 20 [59], so we select from this range. For these simulations, we define the Re number as that for a Poiseuille flow with a pressure gradient given by the amplitude of the imposed gradient. Based on measured Reynolds and Womersley numbers in human arteries [59], we have fit a simple power law $\alpha = A\text{Re}^\gamma$, giving $\gamma = 0.36$ and $A = 0.1$. We select parameters only from this curve within the Re − α plane, as shown in Table II. We use the same cylinder orientation as in Sec. III B and select a diameter $D = 48\Delta x$, as this gives a reasonable balance between computational cost and accuracy, such as would be chosen for production simulations; we keep $L = 4D$. We initialize the simulation to a constant pressure with zero velocity (with the phase of the pressure oscillation chosen such that the driving difference is zero at simulation start). We allow the simulation to run until $\Delta u(t, t - T_{osc}) < 10^{-7}$ is reached for all sample points during one oscillation; however, to reduce the amount of data collected, we record data only for those points within one lattice unit of an axis-normal

TABLE II. Parameters for Womersley flow simulations. All values are given in lattice units. Parameters are as follows: Re number (defined for a Poiseuille flow with the maximum pressure gradient); Womersley number; predicted Ma number (based on the maximum Poiseuille flow); the LBGK relaxation time; the maximum relative density difference; the period of the pressure oscillation, and the momentum diffusion time.

Re	α	Ma	τ	$\Delta\rho/\rho_0$	T_{osc}	T_{mom}	T_s
30	4	0.043	0.620	0.0040	5656	57600	333
100	8	0.078	0.565	0.0039	2608	107000	333
300	12	0.113	0.531	0.0027	2432	223000	333

plane halfway along the cylinder. This was reached in 6–25 oscillation periods.

We run these simulations for *all* combinations of LBM components and compute residuals at four sample points during one pressure oscillation period using Eq. (18). The four residuals are then reduced by taking the root-mean-square average and the maximum, respectively, effectively extending the averaging or maximization over time as well as space. We estimate the maximum as the maximum velocity over a full period at the center of the pipe, i.e.,

$$U_{\mathrm{W}} = \frac{A}{\omega\rho L} \left\| 1 - \frac{1}{J_0(i^{3/2}\alpha)} \right\|$$
$$= \frac{4}{\alpha^2} \left\| 1 - \frac{1}{J_0(i^{3/2}\alpha)} \right\| U_{\max}, \qquad (21)$$

where $U_{\max}$ is the maximum velocity of a Poiseuille flow driven by a pressure gradient of the amplitude of the Womersley flow.

In Fig. 2 we show an example of the simulated velocities, for the D3Q15 velocity set, the LBGK collision operator, and the BFL boundary condition. We plot the axial velocity, normalized by $U_{\max}$, against the radial coordinate at four evenly spaced moments during the period of oscillation. The agreement with the analytical profiles is excellent.

The ℓ^2 error norms for the simulations are shown in Table III. We see that all simulations well approximate the analytical solutions, with errors in the range 0.1%–4%. In Fig. 3, we show the range of error residuals against the Womersley number. This clearly shows the BFL and GZS methods offers superior accuracy to SBB in all cases, irrespective of the choice of collision operator and velocity set. Choosing MRT over LBGK does not offer any benefits, however, we have not varied the relaxation rates for the nonstress tensor moments of the distributions (e.g., by projecting out all the kinetic modes every time step or optimizing "magic numbers" [42]). We note

TABLE III. Error residuals for pulsatile flow simulations with different LBM components. On the left are the boundary condition method (BC); collision operator (CO); and velocity set (DmQn). On the right are the error residuals ϵ_u^2 for different values of Womersley number α (other parameters are as shown in Table II).

			ϵ_u^2		
BC	CO	DmQn	$\alpha = 4$	$\alpha = 8$	$\alpha = 12$
SBB	LBGK	D3Q15	0.0085	0.0250	0.0343
SBB	MRT	D3Q15	0.0083	0.0248	0.0373
SBB	LBGK	D3Q19	0.0079	0.0234	0.0321
SBB	MRT	D3Q19	0.0090	0.0264	0.0395
SBB	LBGK	D3Q27	0.0089	0.0273	0.0395
BFL	LBGK	D3Q15	0.0013	0.0079	0.0128
BFL	MRT	D3Q15	0.0013	0.0078	0.0132
BFL	LBGK	D3Q19	0.0011	0.0069	0.0107
BFL	MRT	D3Q19	0.0012	0.0072	0.0118
BFL	LBGK	D3Q27	0.0012	0.0074	0.0124
GZS	LBGK	D3Q15	0.0013	0.0082	0.0139
GZS	LBGK	D3Q19	0.0011	0.0071	0.0115
GZS	LBGK	D3Q27	0.0012	0.0075	0.0128

again that we have adopted the parameters from [26] without optimization. The choice of velocity set does play a small role, especially in the higher Re and Womersley number cases, leading to slightly improved results with larger velocity sets.

D. Dean flow

The problems in the cylinders studied above are fundamentally 2D flows. To more throroughly test the boundary conditions, we choose a further benchmark problem that is fully 3D: flow in a torus. For moderate Dean numbers (the dynamical similarity number for flow in curved pipes; see below) the primary, axial flow is a perturbation of a parabolic profile while the secondary flow in a cross-sectional plane is a
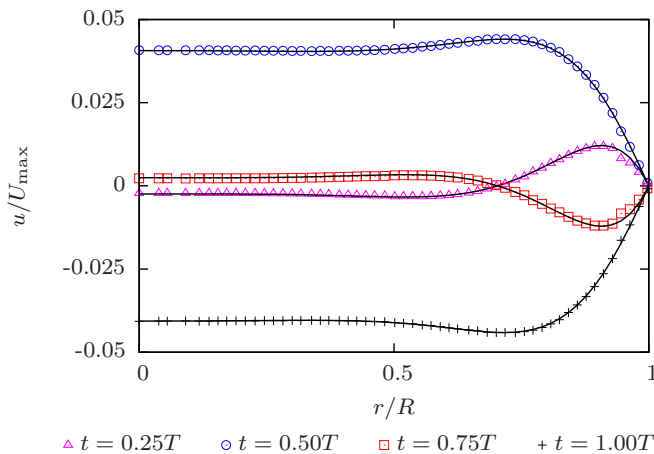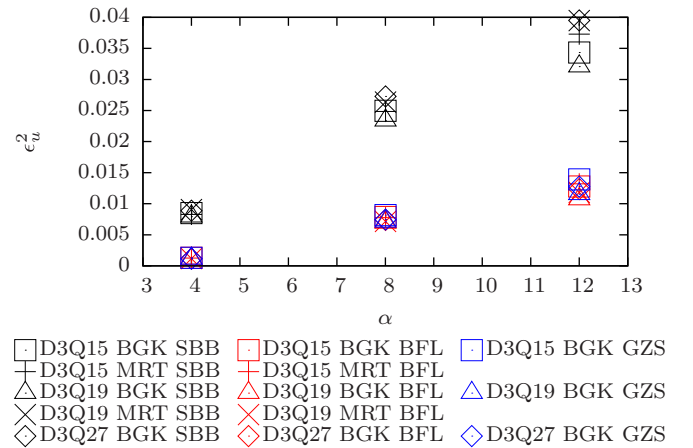

FIG. 2. (Color online) Simulated (D3Q15, LBGK, BFL) axial velocity (normalized by $U_{\max}$) vs the radial coordinate. The Womersley number $\alpha = 12$ and the Re number Re = 300. The data is shown at four moments during the pulsatile cycle: $t = T/4$ (magenta, maximum positive pressure gradient), $T/2$ (blue, zero pressure gradient), $3T/4$ (red, maximum negative pressure gradient), T (black, zero pressure gradient). The analytical solutions are shown with solid lines.

FIG. 3. (Color online) Error residuals ϵ_u^2 as defined in Eq. (18) for simulations of pulsatile flow vs Womersley number α. Colors (black, red, blue) indicate boundary condition (SBB, BFL, GZS) and shapes the collision operator and velocity set. LBGK is indicated by polygons (square, D3Q15; triangle, D3Q19; diamond, D3Q27) and MRT by crosses (+, D3Q15; ×, D3Q19).
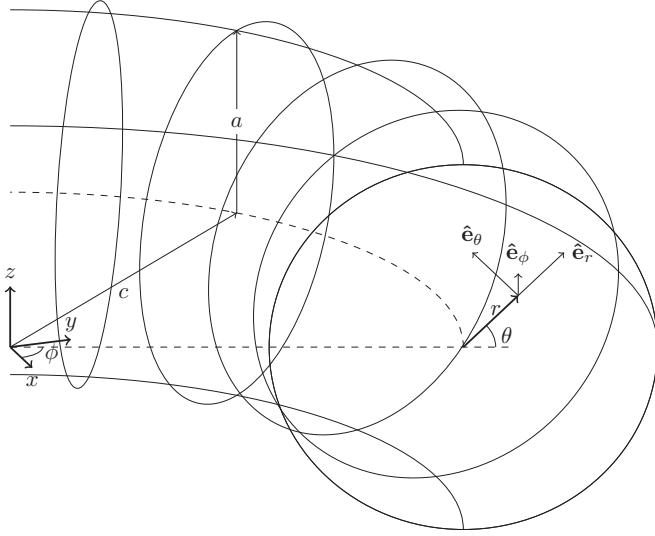
FIG. 4. Parameters and coordinate systems for the torus simulations. The radius of the tube is a and the distance from the center of the tube to the center of the torus is c. We define a toroidal coordinate system (r, θ, ϕ) where (r, θ) are polar coordinates in the cross-section plane and ϕ is the angle that the cross-section plane makes with the x-z plane.

pair of counter-rotation vortices—see the grayscale images in Fig. 5. We define the radius of the tube as a and the distance from the center of the tube to the center of the torus as c; our coordinate systems are illustrated in Fig. 4.

The characteristic number for flow in curved pipes is the Dean number κ given by [65,66]

$$\kappa = 4\text{Re}\sqrt{2\delta}, \qquad (22)$$

where Re is the Reynolds number based on the maxiuum axial velocity of the flow that would result from the same pressure gradient in a *straight* pipe and $\delta \equiv a/c$ is the curvature ratio. The Dean number is the square root of the product of

the inertial and centrifugal forces, scaled by the viscous forces [67].

Dean [65,66] was the first to derive an approximate solution for laminar, fully developed flow in a torus and much of the later work is reviewed in [67]. We choose to use the approximations for a weakly curved pipe from Siggers and Waters [68], due to their clarity of presentation. They define the velocity components $(U_r, U_\theta, U_\phi) \equiv \frac{\nu}{a}(u, v, w)$ and the stream function ψ by

$$u = \frac{1}{hr}\frac{\partial(h\psi)}{\partial\theta} \text{ and } v = -\frac{1}{h}\frac{\partial(h\psi)}{\partial r}, \qquad (23)$$

where $h = 1 + \delta r \cos\theta$. They then expand as a power series in κ,

$$w = \kappa \sum_{k=0}^{\infty} \kappa^{2k} w_k \text{ and } \psi = \kappa^2 \sum_{k=0}^{\infty} \kappa^{2k} \psi_k, \qquad (24)$$

and each term w_k and ψ_k as a series in δ:

$$w_k = \sum_{j=0}^{\infty} \delta^j w_{kj} \text{ and } \psi_k = \sum_{j=0}^{\infty} \delta^j \psi_{kj}. \qquad (25)$$

We use only the terms,

$$w_{00} = \frac{1}{4}(1 - r^2),$$
$$w_{01} = -\frac{3}{16}r(1 - r^2)\cos\theta, \qquad (26)$$
$$w_{02} = \frac{1}{128}(1 - r^2)(-3 + 11r^2 + 10r^2\cos 2\theta),$$

and

$$\psi_{00} = \frac{1}{2^{10} \times 3^2}r(1 - r^2)^2(4 - r^2)\sin\theta,$$

$$\psi_{01} = \frac{1}{2^{12} \times 3^2 \times 5}r^2(1 - r^2)^2(56 - 17r^2)\sin 2\theta, \quad (27)$$

$$\psi_{02} = \frac{1}{2^{17} \times 3^2 \times 5}r(1 - r^2)^2[-(133 - 976r^2$$
$$+ 327r^4)\sin\theta + 2r^2(499 - 172r^2)\sin 3\theta].$$

This expansion is accurate for $\delta \ll 1$ and $\kappa \lesssim 100$ [68].

TABLE IV. Error residuals for flow in a toroidal pipe with different LBM components. On the left are the velocity set (DmQn) and collision operator (CO). On the right are the error residuals ϵ_u^2 for different boundary collision methods for the secondary flow ($\epsilon_{\text{plane}}^2$), the primary flow ($\epsilon_{\text{axial}}^2$), and the overall ($\epsilon_{\text{all}}^2$).

		SBB			BFL			GZS		
DmQn	CO	$\epsilon_{\text{plane}}^2$	$\epsilon_{\text{axial}}^2$	ϵ_{all}^2	$\epsilon_{\text{plane}}^2$	$\epsilon_{\text{axial}}^2$	ϵ_{all}^2	$\epsilon_{\text{plane}}^2$	$\epsilon_{\text{axial}}^2$	ϵ_{all}^2
					$\kappa \approx 112$, Re ≈ 200					
D3Q15	LBGK	0.076	0.022	0.079	0.076	0.018	0.078	0.076	0.018	0.079
D3Q15	MRT	0.076	0.021	0.079	0.076	0.018	0.078		–	
D3Q19	LBGK	0.075	0.020	0.078	0.076	0.017	0.078	0.076	0.017	0.078
D3Q19	MRT	0.075	0.021	0.078	0.076	0.017	0.078		–	
D3Q27	LBGK	0.076	0.022	0.079	0.076	0.017	0.078	0.076	0.017	0.078
					$\kappa \approx 224$, Re ≈ 400					
D3Q15	LBGK	0.153	0.058	0.164	0.152	0.059	0.163		–	
D3Q15	MRT	0.153	0.057	0.164	0.152	0.059	0.163		–	
D3Q19	LBGK	0.152	0.059	0.163	0.152	0.058	0.163		–	
D3Q19	MRT	0.151	0.060	0.163	0.152	0.058	0.162		–	
D3Q27	LBGK	0.152	0.059	0.163	0.152	0.058	0.163		–	

For our simulations, we select a curvature $\delta = 0.1$ and two *target* Dean κ_0 numbers (100 and 200), as representative of physiological values while not making the series expansion too inaccurate. The tube radius $a = 24\Delta x$ (hence $c = 240\Delta x$) and the shear relaxation time $\tau = 0.6\Delta t$ for $\kappa_0 = 100$ and $\tau = 0.55\Delta t$ for $\kappa_0 = 200$. We sample the flow at a plane halfway around the torus to keep data volumes small and impose a parabolic flow profile at the inlet and a constant pressure at the outlet. Since the imposed profile is incorrect, we simulate 90% of the full torus to allow distance for the flow to fully develop. One must also compute the actual Dean number, by equating the imposed flux,

$$Q_{\text{Pois}} = \frac{\pi a \nu \text{Re}_0}{4} = \frac{\pi a \nu \kappa_0}{16\sqrt{2\delta}}, \qquad (28)$$
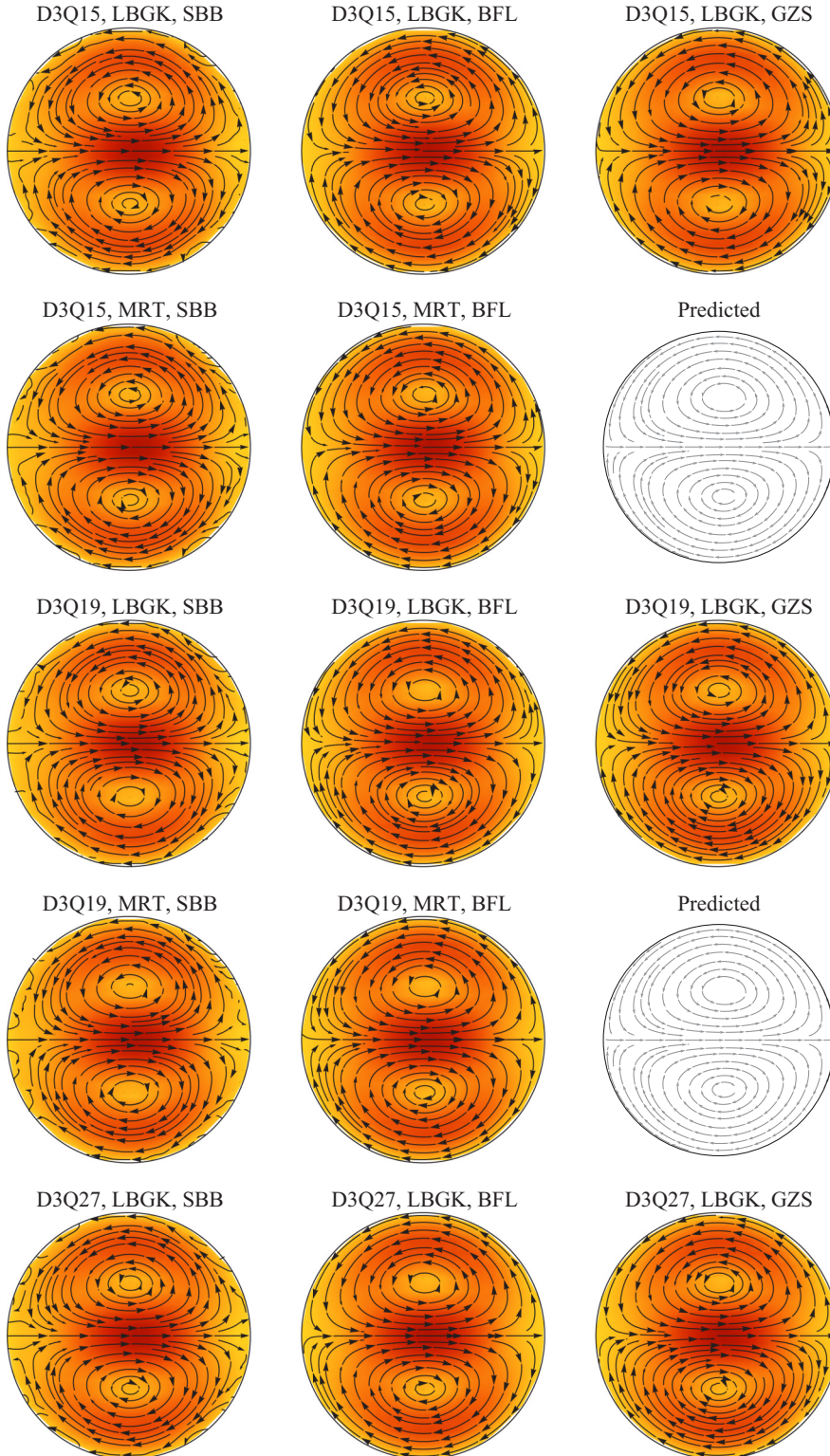

FIG. 5. (Color online) Streamlines for the secondary flow and color maps of the modulus of the error field in a torus, with Dean number $\kappa \approx 112$ and Re number Re ≈ 200. The inside of the pipe is on the left of the figure. Across columns, we vary the boundary condition (SBB, BFL, and GZS from left to right). Down the rows, we vary the combination of velocity set at collision operator (D3Q15+LBGK, D3Q19+LBGK, D3Q15+MRT, D3Q19+MRT, D3Q27+LBGK from top to bottom). The figures in color show the simulated results. The color field indicates the absolute error Eq. (31) where light gray (yellow online) is zero and dark (red online) is $0.15U_\phi(0,0)$. Since the combination of MRT and GZS is unavailable, we use these spaces to show the predicted secondary flow calculated using Eqs. (23)–(27) in grayscale.

to the flux computed from Eq. (24):

$$Q_\kappa = \int_0^1 \int_0^{2\pi} U_\phi a^2 r \, dr \, d\theta = \frac{\pi a \nu \kappa}{384}(48 + \delta^2), \quad (29)$$

giving

$$\kappa = \frac{12\kappa_0}{48 + \delta^2}\sqrt{\frac{2}{\delta}}. \quad (30)$$

This gives our actual Dean numbers as $\kappa \approx 111.78$ and $\kappa \approx 223.56$. We use the ℓ^2-norm error from Eq. (18) but also evaluate it for the primary flow and the secondary flow.

The simulations using the GZS boundary condition were unstable for the larger κ/Re case, but the remaining results are collected in Table IV, and in Figs. 5 and 6 we show streamline plots of the secondary flow, colored by the modulus of the local absolute error $\|\mathbf{u}^\star\| = \|\mathbf{u} - \mathbf{U}\|$ from Eq. (17),
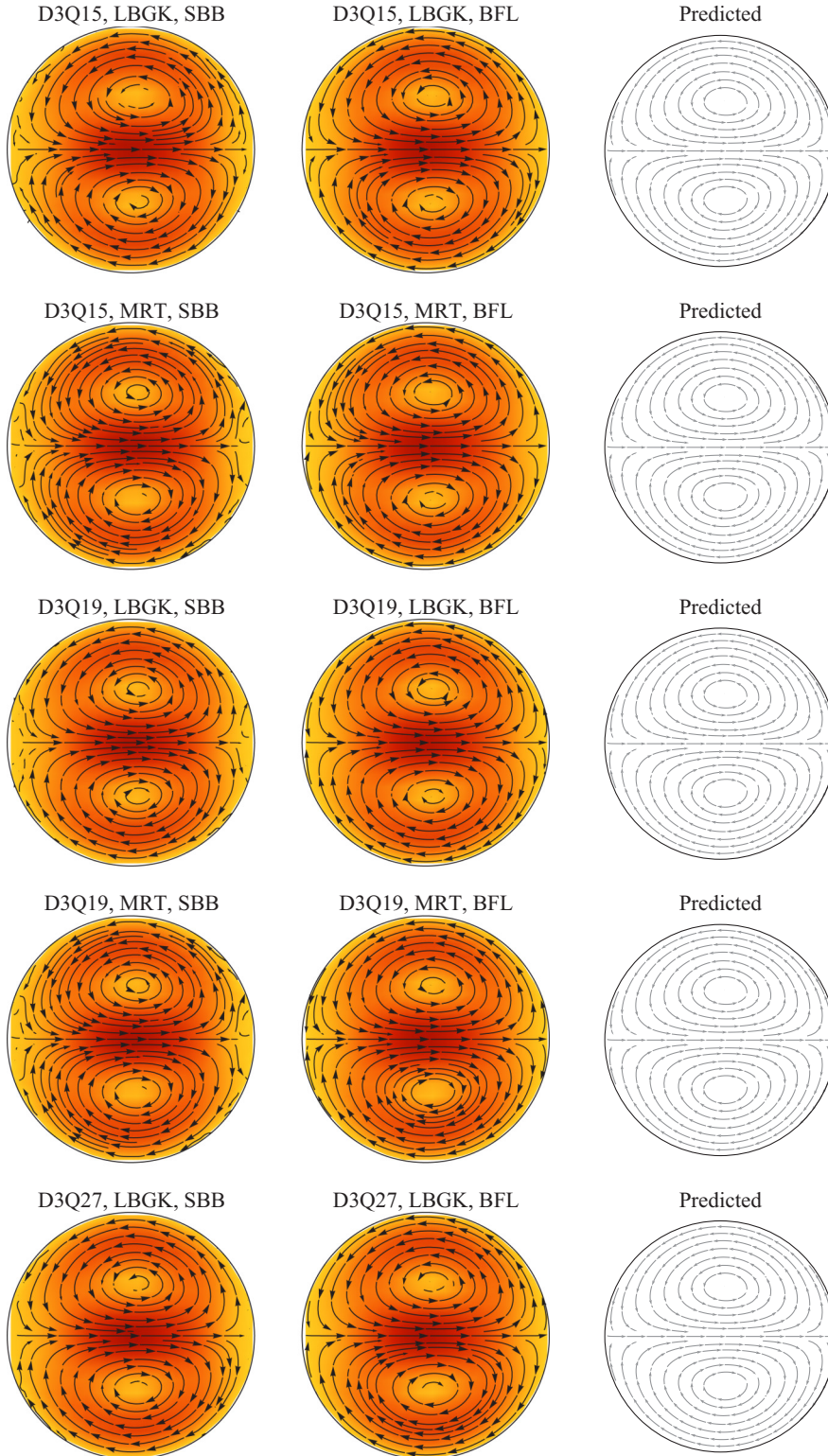

FIG. 6. (Color online) Streamlines for the secondary flow and color maps of the modulus of the error field in a torus, with Dean number $\kappa \approx 224$ and Re number Re ≈ 400. The inside of the pipe is on the left of the figure. Across columns, we vary the boundary condition (SBB, BFL, and GZS from left to right). Down the rows, we vary the combination of velocity set at collision operator (D3Q15+LBGK, D3Q19+LBGK, D3Q15+MRT, D3Q19+MRT, D3Q27+LBGK from top to bottom). The figures in color show the simulated results. The color field indicates the absolute error Eq. (31) where light gray (yellow online) is zero and dark (red online) is $0.3U_\phi(0,0)$. Since the GZS simulations were unstable, we use these spaces to show the predicted secondary flow calculated using Eqs. (23)–(27) in grayscale.

scaled by U_ϕ evaluated at the tube center, i.e.,

$$\epsilon_\kappa = \frac{\|\mathbf{u} - \mathbf{U}\|}{U_\phi(0,0)}. \tag{31}$$

The table of error norms shows that the error is dominated by the error in the secondary flow, which is approximately independent of the LBM model used. This is borne out by Figs. 5 and 6 which show the patterns of the secondary flow error (i.e., the coloring) remaining near constant. The $\kappa \approx 224$ case has much larger errors that vary little which we ascribe to the series solution becoming inaccurate.

We note that our simulations do locate the centers of the pair of vortices accurately. The streamlines for the SBB simulations show large errors in the velocity field's direction near the walls, due to the staircasing of the boundary. This is likely to cause the stress near the walls, which is a key hemodynamic factor, to have larger errors. The primary flow errors for the case $\kappa \approx 112$ are approximately 25% larger for SBB than for either BFL or GZS, again, due to the staircasing of the boundary. The velocity set and collision operator do not strongly affect the measured error, but there is a small benefit to using MRT over LBGK and for using the larger velocity sets.

E. Relative performance

The relative performance of the options is germane to the choice of which combination of lattice-Boltzmann components to use. In Table V we give the measured per-core performance for the lowest Re/α pulsatile flow simulations. All these runs were performed on HECToR, a Cray XE6 supercomputer with two 16-core AMD Opteron 2.3 GHz Interlagos processors per node. The simulations used 64 cores. The performance is given in millions of site updates per second (MSUPS) and is based on timings that include only the lattice-Boltzmann updates. We also estimate the relative time to perform one site update for the different methods, assuming that an SBB site update takes the same time as a bulk fluid site update. The simulation domain includes 347 401 sites of which 42 340 (12%) are at the solid-fluid boundary and hence use the various boundary condition methods.

SBB gives the best computational performance in all cases while BFL requires between 20% and 60% more compute time; the extra time needed remains approximately constant across all collision operators and velocity sets, but reduces as a

fraction of the time to perform a bulk site update. GZS requires approximately five times the computational effort compared to a bulk site.

For the three velocity sets, the number of distribution updates per second (i.e., SUPS × Q) remains approximately constant—D3Q15, 22.5; D3Q19, 22.8; D3Q27, 21.6. The cost of changing the collision operator from LBGK to MRT is large, decreasing performance by a factor of two. We do not place much emphasis on this result as the MRT collision operator is implemented naïvely and there is scope to improve the performance.

IV. CONCLUSION

The majority of benchmark problems reported in the lattice-Boltzmann literature use lattice-aligned geometries, rather than the complex domains required by many applications. We have performed a comparison between LBM simulation solutions, from our open source application HEMELB [19], and analytical solutions in a non-lattice-aligned, curved domain up to Re numbers of 300, with steady and unsteady flow. We have varied the resolution of the grid used and the different components of the algorithm (collision operator, velocity set, no-slip boundary condition). We find that at these moderate values of Re, the choice of velocity set and collision operator do not greatly affect accuracy or stability, but that the choice of no-slip boundary condition method is critical.

Recent studies [69,70] have shown that the commonly used D3Q15 and D3Q19 velocity sets can give strongly orientation (with respect to the underlying lattice) dependent results when simulating flows for Re numbers $\geqslant 250$, while the D3Q27 velocity set maintains good isotropy. The problems studied are in relatively complex cases (a circular pipe with a narrowing; circular and square pipes at higher Re number) but they are comparable to our simulations of flow in a toroidal pipe due to the strong three dimensionality of the Dean flow. We do not see any lattice artifacts in these simulations, even with Re ≈ 400 and the strong secondary flows. Even so, one must be aware of the possibility of anisotropy when simulating a complex flow and ensure that it does not occur in one's work.

The Junk-Yang method shows poor stability and is therefore unsuitable for our hemodynamic applications or other applications that require even moderate-Re numbers.

Simple bounce-back (SBB) shows first-order convergence over a wide range of resolutions and Re numbers, as expected. We confirm that the Bouzidi-Firdaouss-Lallemand (BFL) and (modified) Guo-Zheng-Shi (GZS) methods both show second-order convergence over a wide range of resolutions and Re numbers. For the steady flow problem, GZS has lower errors than BFL, by a variable amount (up to 20%). For the time-dependent problem, BFL has lower errors than GZS, by around 10%. GZS also became unstable for the largest Re number flows in curved pipes.

The typical goal of most computational modeling is to solve the desired system of equations to some problem-dependent accuracy in the minimum time. Considering the steady flow problem as a proxy, we can estimate which of the GZS and BFL boundary conditions will give the shortest simulation time. Taking an average increase in accuracy for GZS of $\sim$10% allows us to estimate that the GZS method can use a

TABLE V. Performance per core for different combinations of boundary condition, collision operator, and velocity set. The columns indicate the combination of collision operator and velocity set, while the rows indicate the boundary condition. Omitted simulations are indicated with a dash (–). The values are given in millions of site updates per second (MSUPS) followed by the time to perform one boundary condition site update, relative to SBB, in parentheses.

	LBGK			MRT	
	D3Q15	D3Q19	D3Q27	D3Q15	D3Q19
SBB	1.5 (1)	1.2 (1)	0.8 (1)	0.8 (1)	0.5 (1)
BFL	1.4 (1.6)	1.2 (1.4)	0.8 (1.3)	0.8 (1.3)	0.5 (1.2)
GZS	1.0 (4.9)	0.8 (5.1)	0.5 (6.1)	–	–

$\sim 5\%$ lower resolution to obtain the same accuracy. Due to the diffusive scaling of the time step, the GZS method will require fewer site updates than BFL by a factor of $\sim 0.95^5 \approx 0.79$ or 21% less. However, the BFL method is more computationally efficient, by $\sim 30\%$, for the typical surface:bulk ratios studied here, so the equivalent-accuracy BFL simulation will complete sooner.

We therefore recommend BFL as the best all-round boundary condition of those tested, due to the good accuracy, performance, stability, and simplicity of implementation. The GZS method also offers good accuracy (in our modified form) but has worse stability at high Re number and much poorer performance. This last point may not matter in domains with a relatively low number of wall boundary sites. SBB is much less accurate, but may be acceptable for use in undemanding applications and when development time is at a premium. We believe that these results will prove helpful to the community when selecting methods for simulating hemodynamics and comparable applications with lattice-Boltzmann methods.

[1] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Clarendon Press, Oxford, 2001).

[2] S. Chen and G. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).

[3] C. K. Aidun and J. R. Clausen, Annu. Rev. Fluid Mech. **42**, 439 (2010).

[4] E. O. Kung, A. S. Les, F. Medina, R. B. Wicker, M. V. McConnell, and C. A. Taylor, J. Biomech. Eng. **133**, 041003 (2011).

[5] M. C. Villa-Uriol, I. Larrabide, J. M. Pozo, M. Kim, O. Camara, M. De Craene, C. Zhang, A. J. Geers, H. Morales, H. Bogunović, R. Cardenes, and A. F. Frangi, Phil. Trans. R. Soc. A **368**, 2961 (2010).

[6] X. He, G. Duckwiler, and D. Valentino, Comput. Fluids **38**, 789 (2008).

[7] R. L. Spilker, J. A. Feinstein, D. W. Parker, V. M. Reddy, and C. A. Taylor, Ann. Biomed. Eng. **35**, 546 (2007).

[8] C. A. C. Taylor and C. A. C. Figueroa, Annu. Rev. Biomed. Eng. **11**, 109 (2009).

[9] H. Tahir, A. G. Hoekstra, E. Lorenz, P. V. Lawford, D. R. Hose, J. Gunn, and D. J. W. Evans, Interface Focus **1**, 365 (2011).

[10] Z. Merali, Nature (London) **467**, 775 (2010).

[11] A. Morin, J. Urban, P. D. Adams, I. Foster, A. Sali, D. Baker, and P. Sliz, Science **336**, 159 (2012).

[12] J. Lätt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler, Phys. Rev. E **77**, 056703 (2008).

[13] J. Boyd, J. M. Buick, J. Cosgrove, and P. Stansell, Australas. Phys. Eng. Sci. Med. **27**, 207 (2004).

[14] Z. Guo, C. Zheng, and B. Shi, Phys. Fluids **14**, 2007 (2002).

[15] B. Stahl, B. Chopard, and J. Lätt, Comput. Fluids **39**, 1625 (2010).

[16] S. Melchionna, M. Bernaschi, S. Succi, E. Kaxiras, F. J. Rybicki, D. Mitsouras, A. Coskun, and C. L. Feldman, Comput. Phys. Commun. **181**, 462 (2010).

[17] M. Bernaschi, S. Melchionna, S. Succi, M. Fyta, E. Kaxiras, and J. K. Sircar, Comput. Phys. Commun. **180**, 1495 (2009).

[18] C. Pan, L.-S. Luo, and C. T. Miller, Comput. Fluids **35**, 898 (2006).

[19] http://ccs.chem.ucl.ac.uk/hemelb.

[20] R. L. Liboff, *Kinetic Theory: Classical, Quantum, and Relativistic Descriptions*, 2nd ed. (John Wiley & Sons, New York, 1998).

[21] Y. H. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[22] H. Chen, S. Chen, and W. H. Matthaeus, Phys. Rev. A **45**, R5339 (1992).

[23] Bhatnagar, E. P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).

[24] A. J. C. Ladd and R. Verberg, J. Stat. Phys. **104**, 1191 (2001).

[25] D. d'Humières, Progress in Aeronautics and Astronautics **159**, 450 (1992).

[26] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, Phil. Trans. R. Soc. A **360**, 437 (2002).

[27] T. Inamuro, M. Yoshino, and F. Ogino, Phys. Fluids **7**, 2928 (1995).

[28] Q. Zou and X. He, Phys. Fluids **9**, 1591 (1997).

[29] J. Lätt, Ph.D. thesis, Université de Genève, 2007.

[30] S. Ansumali and I. V. Karlin, Phys. Rev. E **66**, 026311 (2002).

[31] O. Malaspinas, B. Chopard, and J. Latt, Comput. Fluids **49**, 29 (2011).

[32] D. P. Ziegler, J. Stat. Phys. **71**, 1171 (1993).

[33] I. Ginzbourg and P. M. Adler, J. Phys. II France **4**, 191 (1994).

[34] A. J. C. Ladd, J. Fluid Mech. **271**, 285 (1994).

[35] M. Bouzidi, M. Firdaouss, and P. Lallemand, Phys. Fluids **13**, 3452 (2001).

[36] M. Junk and Z. Yang, Phys. Rev. E **72**, 066701 (2005).

[37] M. Junk and Z. Yang, J. Stat. Phys. **121**, 3 (2005).

[38] I. Ginzburg and D. d'Humières, Phys. Rev. E **68**, 066614 (2003).

[39] B. Chun and A. J. C. Ladd, Phys. Rev. E **75**, 066705 (2007).

[40] X. He, Q. Zou, L.-S. Luo, and M. Dembo, J. Stat. Phys. **87**, 115 (1997).

[41] L.-S. Luo, W. Liao, X. Chen, Y. Peng, and W. Zhang, Phys. Rev. E **83**, 056710 (2011).

[42] D. d'Humières and I. Ginzburg, Comput. Math. Appl. **58**, 18 (2009).

[43] O. Filippova and D. Hänel, J. Comput. Phys. **147**, 219 (1998).

[44] X. He, L.-S. Luo, and M. Dembo, J. Comput. Phys. **129**, 357 (1996).

[45] H. Chen, Phys. Rev. E **58**, 3955 (1998).

[46] S. Ubertini, G. Bella, and S. Succi, Phys. Rev. E **68**, 016701 (2003).

[47] G. Falcucci, M. Aureli, S. Ubertini, and M. Porfiri, Phil. Trans. R. Soc. A **369**, 2456 (2011).

[48] C. S. Peskin, Acta Numerica **11**, 479 (2002).

[49] Z.-G. Feng and E. Michaelides, J. Comput. Phys. **195**, 602 (2004).

[50] J. Zhang, P. C. Johnson, and A. S. Popel, Phys. Biol. **4**, 285 (2007).

[51] T. Krüger, F. Varnik, and D. Raabe, Phys. Rev. E **79**, 046704 (2009).

[52] S. S. Chikatamarla, S. Ansumali, and I. V. Karlin, Europhys. Lett. **74**, 215 (2006).

[53] K. Mattila, J. Hyväluoma, A. A. Folarin, and T. Rossi, Int. J. Numer. Meth. Fluids **63**, 638 (2010).

[54] K. Mattila, J. Hyväluoma, and T. Rossi, J. Stat. Mech. (2009) P06015.

[55] M. D. Mazzeo and P. V. Coveney, Comput. Phys. Commun. **178**, 894 (2008).

[56] M. D. Mazzeo, S. Manos, and P. V. Coveney, Comput. Phys. Commun. **181**, 355 (2010).

[57] D. Groen, J. Hetherington, H. B. Carver, R. W. Nash, M. O. Bernabeu, and P. V. Coveney, J. Comput. Sci. **4**, 412 (2013).

[58] A. Alexandrescu, *Modern C++ Design: Generic Programming and Design Patterns Applied* (Addison-Wesley, Boston, 2001).

[59] J. D. Bronzino, *Biomedical Engineering Fundamentals (The Biomedical Engineering Handbook)*, 3rd ed. (CRC Press, Boca Raton, 2006).

[60] D. J. Holdych, D. R. Noble, J. G. Georgiadis, and R. O. Buckius, J. Comput. Phys. **193**, 595 (2004).

[61] A. Caiazzo, J. Stat. Phys. **121**, 37 (2005).

[62] A. M. Artoli, A. G. Hoekstra, and P. M. A. Sloot, Comput. Fluids **35**, 227 (2006).

[63] R. Mei, L. S. Luo, P. Lallemand, and D. d'Humières, Comput. Fluids **35**, 855 (2006).

[64] L. Formaggia, A. Quarteroni, and A. Veneziani, *Cardiovascular Mathematics* (Springer-Verlag Italia, Milano, 2009).

[65] W. R. Dean, Philosophical Magazine **4**, 208 (1927).

[66] W. R. Dean, Philosophical Magazine **5**, 673 (1928).

[67] S. A. Berger, L. Talbot, and L. S. Yao, Annu. Rev. Fluid Mech. **15**, 461 (1983).

[68] J. H. Siggers and S. L. Waters, Phys. Fluids **17**, 077102 (2005).

[69] A. T. White and C. K. Chong, J. Comput. Physics **230**, 12 (2011).

[70] S. K. Kang and Y. A. Hassan, J. Comput. Phys. **232**, 100 (2012).