On the inclusion of mass source terms in a single-relaxation-time lattice Boltzmann method

Olav Aursjø, Espen Jettestuen, Jan Ludvig Vinningland, and Aksel Hiorth

Citation: Physics of Fluids **30**, 057104 (2018); doi: 10.1063/1.5024641
View online: https://doi.org/10.1063/1.5024641
View Table of Contents: http://aip.scitation.org/toc/phf/30/5
Published by the American Institute of Physics

On the inclusion of mass source terms in a single-relaxation-time lattice Boltzmann method

Olav Aursjø,[1] Espen Jettestuen,[1] Jan Ludvig Vinningland,[1] and Aksel Hiorth[1,2]
[1]International Research Institute of Stavanger (IRIS), P.O. Box 8046, N-4068 Stavanger, Norway
[2]The National IOR Centre of Norway, University of Stavanger, N-4036 Stavanger, Norway

(Received 2 February 2018; accepted 20 April 2018; published online 10 May 2018)

We present a lattice Boltzmann algorithm for incorporating a mass source in a fluid flow system. The proposed mass source/sink term, included in the lattice Boltzmann equation, maintains the Galilean invariance and the accuracy of the overall method, while introducing a mass source/sink term in the fluid dynamical equations. The method can, for instance, be used to inject or withdraw fluid from any preferred lattice node in a system. This suggests that injection and withdrawal of fluid does not have to be introduced through cumbersome, and sometimes less accurate, boundary conditions. The method also suggests that, through a chosen equation of state relating mass density to pressure, the proposed mass source term will render it possible to set a preferred pressure at any lattice node in a system. We demonstrate how this model handles injection and withdrawal of a fluid. And we show how it can be used to incorporate pressure boundaries. The accuracy of the algorithm is identified through a Chapman-Enskog expansion of the model and supported by the numerical simulations. *Published by AIP Publishing.* https://doi.org/10.1063/1.5024641

I. INTRODUCTION

The lattice Boltzmann method[1–3] (LBM) has over the last couple of decades been successfully applied to a wide variety of fluid dynamical systems, such as single- and multi-phase flow in complex porous geometries,[4–11] thermal flows, and phase changes,[12] along with numerous other subjects. Common to many of these applications is that fluid enters and exits the system domain. Typically, such a behavior is described by assigning specific boundary conditions to the system to drive and modify the fluid flow.[13,14] Inspired by how mass has been added in systems describing reactive flows,[2,15–17] another way to allow fluid entering or exiting the system domain is to introduce a mass source or sink term in the fluid dynamical equations describing the fluid behavior. Injecting/withdrawing fluid through a mass source/sink will, in many circumstances, be superior to applying a boundary condition. It would, for instance, not be ideal to introduce a boundary in the middle of a system in order to inject fluid there, which would typically imply some kind of interpolation or extrapolation scheme. Instead, this can easily be facilitated by having a mass source there. Particularly, in microfluidics,[18,19] where it can be sensible to run 2D simulations of a lab-on-a-chip system, it would be practical to describe out-of-plane injection of fluid as a mass source in the 2D description of the system.

Although injection/withdrawal of fluids is the most obvious application of having a mass source in the fluid dynamical equations, it should be emphasized that the use is not limited to this application alone. For instance, since mass density and pressure typically are related through an equation of state, one can also utilize a mass source to enforce pressure changes in a system. And, in multi-phase settings, one can imagine that mass sources and sinks could be used to describe a reaction where one fluid phase is transformed into another.

We will in this paper introduce, in the single-relaxation-time (SRT) LBM, a source term capable of describing generic time dependent inhomogeneous mass sources. This proposed term is local both in space and time. The added term preserves the Galilean invariance and the accuracy of the overall method. To our knowledge, Halliday *et al.*[20] are the first to propose a method for including a mass source into the SRT model. In their work, they propose a system-specific mass source term where they employ a non-local scheme to calculate the spatial derivatives that appear in their source term. Cheng and Li[21] propose a more general mass source term in their SRT model, where they in a similar fashion apply a non-local scheme for temporal and spatial derivatives in order to get rid of undesirable terms in the macroscopic continuum equations. The non-locality of these schemes leads to an unfortunate increase in communication between neighboring lattice sites. This is especially inefficient for parallelization. Kuzmin, Guo, and Mohamad[22] propose a mass source term for a multi-relaxation-time (MRT) model. This term does not leave the resulting fluid dynamical equations Galilean invariant (GI). Wang *et al.*[23] have recently proposed a mass source term in an SRT model in order to simulate wave making. They have based their model on much the same foundation as our proposed model. However, the presented continuum equations lack Galilean invariance, and the resulting mass source term differs considerably from the one presented in this current paper.

In this paper, we present a local lattice Boltzmann algorithm for including a general mass source term that results in Galilean invariant continuum equations. We will in Sec. II introduce the set of Galilean invariant continuum equations we seek to solve with our proposed LBM. In Sec. III, we describe the SRT model where the mass source term is included. Numerical examples, demonstrating some of the capabilities of the proposed LBM, are presented in Sec. IV. And, in

Sec. V, we summarize the work and give some final concluding remarks.

II. GOVERNING CONTINUUM EQUATIONS

The state of a moving fluid may be described through its velocity $\mathbf{u}(\mathbf{x}, t)$, its pressure $p(\mathbf{x}, t)$, and its mass density $\rho(\mathbf{x}, t)$ at any position $\mathbf{x}$ and at any time t. These quantities are governed by an equation of state, together with mass and momentum conservation. Using the Einstein summation convention, where Latin indices denote Cartesian spatial components, mass and momentum conservation may, in component form, be described through[24,25]

$$\partial_t \rho + \partial_i(\rho u_i) = q, \tag{1}$$

$$\rho\left(\partial_t u_i + u_j \partial_j u_i\right) = -\partial_i p + F_i + \partial_j \tau_{ij}, \tag{2}$$

where $q(\mathbf{x}, t)$ is a mass source or sink, $p(\mathbf{x}, t)$ is the pressure of the fluid, $F_i(\mathbf{x}, t)$ is the component of any applied volume force, and $\tau_{ij}(\mathbf{x}, t)$ are the components of the deviatoric stress tensor. For Newtonian fluids,

$$\tau_{ij} = \rho\nu\left[\partial_i u_j + \partial_j u_i - \frac{2}{d}\partial_k u_k \delta_{ij}\right] + \xi \partial_k u_k \delta_{ij} \tag{3}$$

expresses the viscous stress tensor, where δ_{ij} is the Kronecker delta. Here, ν and ξ are, respectively, the kinematic shear viscosity and the bulk (or volume) viscosity of the fluid and d is the number of spatial dimensions in the system. For the fluid motion to be fully described, an additional equation of state is required to describe the relation between the pressure and the mass density of the fluid.

As can be seen from Eq. (1), we have here included a source/sink term $q(\mathbf{x}, t)$, allowing mass to be added to, or removed from, the described system. Combining Eqs. (1) and (2) it can now be shown that the momentum conservation equation takes the form

$$\partial_t(\rho u_i) + \partial_j \Pi_{ij} = F_i + q u_i, \tag{4}$$

where the components of the momentum flux density tensor $\Pi_{ij} = \rho u_i u_j + p\delta_{ij} - \tau_{ij}$. Notice here that a term qu_i has, due to the inclusion of the source q in Eq. (1), appeared on the right-hand side of the momentum conservation equation. Here, it should be pointed out that this term qu_i is essential to maintain Galilean invariance in the LBM presented in Sec. III.

III. MODEL DESCRIPTION

The lattice Boltzmann method[1–3] describes, on a mesoscopic scale, the behavior of a physical system in terms of density distribution functions $f_\alpha(\mathbf{x}, t)$. The $f_\alpha(\mathbf{x}, t)$, defined in a discrete space-time, describes the density of particles at position $\mathbf{x}$ at time t with velocity $\mathbf{c}_\alpha$. In the models, $\mathbf{c}_\alpha$ represents a finite set of discrete velocities, giving one discrete velocity for each lattice direction α. The distribution functions evolve according to the equation

$$f_\alpha(\mathbf{x} + \mathbf{c}_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) = \Omega_\alpha, \tag{5}$$

where Ω_α represents a collision term. The general expression for Ω_α, using the Bhatnagar-Gross-Krook (BGK) collision operator, is

$$\Omega_\alpha = \Omega_\alpha^{\mathrm{BGK}} + \Delta\Omega_\alpha = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{\mathrm{eq}}\right) + \Delta\Omega_\alpha, \tag{6}$$

where $\tau\Delta t$ is the relaxation time of the process and $\Delta\Omega_\alpha$ is a source-and-correction term for recovering the macroscopic equations. The equilibrium distribution is usually on the form[2]

$$f_\alpha^{\mathrm{eq}}(\rho, \mathbf{u}) = w_\alpha \rho\left(1 + \frac{c_{\alpha i} u_i}{c_s^2} + \frac{Q_{\alpha ij} u_i u_j}{2c_s^4}\right), \tag{7}$$

where c_s^2 is a constant relating to the chosen lattice structure and

$$Q_{\alpha ij} = c_{\alpha i} c_{\alpha j} - c_s^2 \delta_{ij}. \tag{8}$$

Here we split the source-and-correction term $\Delta\Omega_\alpha$ into a force term $\Delta\Omega_\alpha^F$ and a mass source term $\Delta\Omega_\alpha^q$, leaving $\Delta\Omega_\alpha = \Delta\Omega_\alpha^F + \Delta\Omega_\alpha^q$. The force dependent term[26]

$$\Delta\Omega_\alpha^F = w_\alpha\left(1 - \frac{1}{2\tau}\right)\left(\frac{c_{\alpha i} F_i}{c_s^2} + \frac{Q_{\alpha ij} u_i F_j}{c_s^4}\right)\Delta t, \tag{9}$$

where F_i are the components of the applied volume force, and the mass-source dependent term

$$\Delta\Omega_\alpha^q = w_\alpha\left(1 - \frac{1}{2\tau}\right)\left(1 + \frac{c_{\alpha i} u_i}{c_s^2} + \frac{Q_{\alpha ij} u_i u_j}{2c_s^4}\right) q\Delta t, \tag{10}$$

where q is the mass source/sink. This last term allows us to inject or withdraw fluid from the system. By correspondingly defining the sum

$$\sum_\alpha f_\alpha(\mathbf{x}, t) = \rho(\mathbf{x}, t) - \frac{1}{2} q(\mathbf{x}, t)\Delta t \tag{11}$$

and the weighted sum

$$\sum_\alpha f_\alpha(\mathbf{x}, t) c_{\alpha i} = \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) - \frac{1}{2}\left[F_i(\mathbf{x}, t) + q(\mathbf{x}, t) u_i(\mathbf{x}, t)\right]\Delta t, \tag{12}$$

the distribution functions are related to the macroscopic mass density $\rho(\mathbf{x}, t)$ and momentum density $\rho(\mathbf{x}, t)\, u_i(\mathbf{x}, t)$. Combing Eqs. (11) and (12), the fluid velocity is given as

$$u_i(\mathbf{x}, t) = \frac{1}{\sum_\alpha f_\alpha}\left[\sum_\alpha f_\alpha c_{\alpha i} + \frac{1}{2} F_i(\mathbf{x}, t)\Delta t\right]. \tag{13}$$

It may be shown, through a Chapman-Enskog expansion[27] of Eq. (5), that the continuum equations of Eqs. (1) and (2) are recovered in the long-wavelength limit. To achieve this, the kinematic viscosity and the bulk viscosity must, respectively, be related to the relaxation time parameter τ by $\nu = c_s^2(\tau - 1/2)\Delta t$ and $\xi = 2\rho\nu/d$, for a d-dimensional lattice. The pressure is related to the mass density through the equation of state $p = c_s^2 \rho$. The details of this derivation are given in the Appendix.

Recognizing that, in our lattice Boltzmann (LB) model, the pressure of the fluid is directly related to the mass density through the equation of state $p = c_s^2 \rho$ and that the mass density is given by Eq. (11), it is observed that a source term

$$q(\mathbf{x}, t)\Delta t = 2\left[\hat{p}(\mathbf{x}, t) c_s^{-2} - \sum_\alpha f_\alpha(\mathbf{x}, t)\right] \tag{14}$$

will presumptively fix the pressure to $\hat{p}$ at a given position. Applying Eq. (14) to Eq. (11) will, per definition, fix the macroscopic mass density (and pressure) calculated at that lattice node. However, it is the source term contribution $\Delta\Omega_\alpha^q$ in the collision step that determines how the distribution functions are modified and thereby what functions are being propagated to the neighboring lattice nodes. This is essential in transferring the correct pressure. We will in Sec. IV B show how this can be used to implement fixed pressure boundary conditions.

IV. NUMERICAL EXAMPLES

In our numerical examples, we have implemented the described numerical scheme using a D2Q9 lattice. Lattice velocities and the corresponding lattice weights are presented in Table I. In this model, the lattice constant $c_s^2 = 1/3$. We have, in the numerical simulations, put the initial particle mass density $\rho_{0\mathrm{LB}}$, lattice step Δx_{LB}, and lattice time step Δt_{LB} to unity. (In Secs. IV A and IV B, the subscript/superscript LB will denote the non-dimensional lattice Boltzmann parameters.)

The spatial resolution of the system is $\Delta x = L_{\mathrm{phys}}/N_{\mathrm{LB}}$, where L_{phys} is the physical length of the system, while N_{LB} is the number of lattice points which represents this length numerically. The physical time step Δt of the simulation is related to the physical spatial resolution Δx, but the exact relation depends on the physical parameter that connects time and space. Keeping the ratio of LB viscosity to physical kinematic viscosity $\nu_{\mathrm{phys}} = \nu_{\mathrm{LB}}(\Delta x)^2/\Delta t$ constant yields a time step $\Delta t = (\Delta x)^2 \nu_{\mathrm{LB}}/\nu_{\mathrm{phys}}$. Keeping, on the other hand, the ratio of LB velocity to physical velocity $\mathbf{u}_{\mathrm{phys}} = \mathbf{u}_{\mathrm{LB}}\Delta x/\Delta t$ constant yields a time step $\Delta t = \Delta x \mathbf{u}_{\mathrm{LB}}/\mathbf{u}_{\mathrm{phys}}$. The first relation is termed diffusive scaling ($\Delta t \propto (\Delta x)^2$), while the latter relation is termed linear scaling ($\Delta t \propto \Delta x$). Both scaling relations have been used in the simulations presented here.

With a constant ratio $\Delta\rho = \rho_{\mathrm{phys}}/\rho_{\mathrm{LB}}$, we have for the source term, introduced in Eq. (1),

$$q_{\mathrm{phys}} = q^{\mathrm{LB}} \frac{\Delta\rho}{\Delta t}. \tag{15}$$

With diffusive scaling, this yields $q^{\mathrm{LB}} \propto \nu_{\mathrm{LB}}/N_{\mathrm{LB}}^2$ and time $t_{\mathrm{LB}} = N_{\mathrm{LB}}^2/\nu_{\mathrm{LB}}$. And with linear scaling, the source term $q^{\mathrm{LB}} \propto N_{\mathrm{LB}}^{-1}$, the viscosity $\nu_{\mathrm{LB}} \propto N_{\mathrm{LB}}$, and time $t_{\mathrm{LB}} \propto N_{\mathrm{LB}}$.

A. Potential flow

The first numerical examples we present describe a potential flow generated by one source and one sink of same shape and strength, spatially separated by a given distance. To reduce possible precision degradation, due to lower accuracy boundary conditions, these first numerical examples only involve periodic boundary conditions.

The potential flow is generated by one source and one sink, mathematically described, in a d-dimensional system, as

$$q(\mathbf{x}, t) = q_0 \left[\prod_{n=1}^{d} \delta_\epsilon (x_n - x_n^{\mathrm{src}}) - \prod_{n=1}^{d} \delta_\epsilon \left(x_n - x_n^{\mathrm{sink}}\right) \right], \tag{16}$$

where x_n^{src} and x_n^{sink} are the spatial coordinates for the center points of, respectively, the source and the sink. Here,

$$\delta_\epsilon(x) = \begin{cases} \frac{1}{2\epsilon L}\left[1 + \cos\left(\frac{\pi x}{\epsilon L}\right)\right], & \text{if } |x| \le \epsilon L, \\ 0, & \text{if } |x| > \epsilon L, \end{cases} \tag{17}$$

where L is the length scale of the system and $0 \le \epsilon \le 1$. It can be shown that the Dirac delta function $\delta(x) = \lim_{\epsilon\to 0}\delta_\epsilon(x)$. In the following examples, ϵ is kept constant at $\epsilon = 5.0 \times 10^{-2}$.

The physical source amplitude

$$q_0^{\mathrm{phys}} = q_0^{\mathrm{LB}} \frac{\Delta\rho}{\Delta t} (\Delta x)^d \tag{18}$$

for a d-dimensional system. This means that $q_0^{\mathrm{LB}} \propto \Delta t N_{\mathrm{LB}}^d$. Note that the dimensionality of this relation depends on the number of spatial dimensions d in the system. With diffusive scaling this yields $q_0^{\mathrm{LB}} \propto \nu_{\mathrm{LB}} N_{\mathrm{LB}}^{d-2}$, while with linear scaling $q_0^{\mathrm{LB}} \propto N_{\mathrm{LB}}^{d-1}$.

1. 1D potential flow example

Let us first look at a one-dimensional version of this type of potential flow. In the simulations we have centered the sink around $x^{\mathrm{sink}} = 0$, while the source is centered around the midpoint of the system at $x^{\mathrm{src}} = N_{\mathrm{LB}}/2$. The system is spatially periodic. The number of lattice points N_{LB} is varied to study the effect of spatial resolution on the numerical results. The numerical parameters used in these simulations are given in Table II. The lower panel in Fig. 1 shows the source/sink term as a function of position.

In 1D, we know analytically that the absolute value of the fluid velocity $|u_{\mathrm{A}}(x_j)| = q_0^{\mathrm{LB}}/(2\rho_{0\mathrm{LB}})$ for all $x_j \in \{x |\, q(x) = 0\}$ when the system has reach a stationary state. The upper panel of Fig. 1 shows a representation of the mass flux $J(x) = \rho(x)\, u(x)$ as a function of position in this state.

We may, for the stationary velocity field, define an average relative error

$$E_u = \frac{\sum_j |u(x_j) - u_{\mathrm{A}}(x_j)|}{\sum_j |u_{\mathrm{A}}(x_j)|}, \tag{19}$$

where $x_j \in \{x |\ q(x) = 0\}$. Figure 2 shows the average relative errors E_u for all the sets of simulations listed in Table II. From the inset of this figure, we see that, for the simulation sets with fixed τ values (diffusive scaling), the error E_u is

TABLE I. Lattice weights and velocities for the D2Q9 lattice.

Lattice direction	$c_\alpha(\Delta x_{\mathrm{LB}}/\Delta t_{\mathrm{LB}})$	w_α
$\alpha = 0$	(0, 0)	4/9
$\alpha = 1, \ldots, 4$	(±1, 0), (0, ±1)	1/9
$\alpha = 5, \ldots, 8$	(±1, ±1), (∓1, ±1)	1/36

TABLE II. The parameter sets used in the 1D source-term simulations presented in Sec. IV A 1. All quantities are given in LB units.

	Diffusive scaling			Linear scaling	
	$\tau = 3/5$	$\tau = 1$	$\tau = 3/2$		
N_{LB}	q_0^{LB}	q_0^{LB}	q_0^{LB}	τ	q_0^{LB}
40	$(1/2)\times 10^{-3}$	$(5/2)\times 10^{-3}$	5.0×10^{-3}	0.7	1.0×10^{-3}
80	$(1/4)\times 10^{-3}$	$(5/4)\times 10^{-3}$	2.5×10^{-3}	0.9	
100	$(1/5)\times 10^{-3}$	1.0×10^{-3}	2.0×10^{-3}	1	
200	1.0×10^{-4}	$(1/2)\times 10^{-3}$	1.0×10^{-3}	1.5	
400	$(1/2)\times 10^{-4}$	$(1/4)\times 10^{-3}$	5.0×10^{-4}	2.5	
800	$(1/4)\times 10^{-4}$	$(1/8)\times 10^{-3}$	2.5×10^{-4}	4.5	

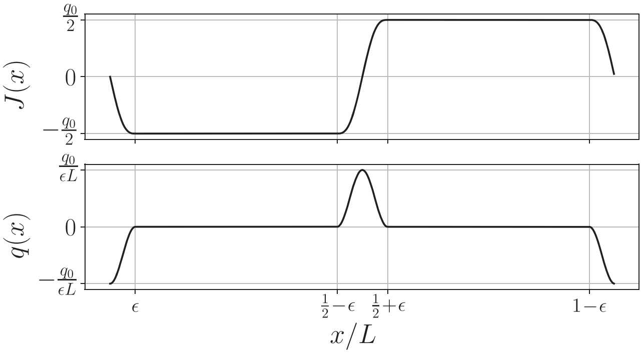

FIG. 1. Representation of the 1D system presented in Sec. IV A 1. The lower panel shows the applied mass source term $q(x)$, while the upper panel shows the resulting mass flux $J(x)$. These plots are from the simulation with $\tau = 3/2$ and system size $N_{\text{LB}} = 800$.

inversely proportional to the square of the system size N_{LB}, i.e., $E_u \sim N_{\text{LB}}^{-2}$. We observe deviations from this behavior for simulations exhibiting the highest spatial and temporal resolutions. This is due to rounding errors caused by the finite machine precision (8 bytes, unit roundoff $\varepsilon_{\text{r}} = 2.22 \times 10^{-16}$). For the set of simulations exhibiting linear scaling, we notice that the calculated relative errors are independent of grid resolution. The main plot of Fig. 2 shows the errors as a function of the source term amplitude q_0^{LB} given in lattice units. We observe that the error $E_u \sim (q_0^{\text{LB}})^2$. This is equivalent to E_u being proportional to the fluid velocity squared. It is commonly accepted that the error due to compressibility in the standard lattice Bhatnagar-Gross-Krook (LBGK) model goes as the Mach number squared.[28] We also observe that the average relative errors in density ρ, in the same manner, go as the fluid velocity squared. We therefore conclude that the observed errors in the simulations are due to the compressibility intrinsic to the LBGK model.

2. 1D potential flow example: Moving sources

We stated in Sec. II, concerning the fluid dynamical continuum equations, that introducing a source q in the continuity equation Eq. (1) also leads to a qu_i-term in Eq. (4). Overlooking this qu_i-term results in a set of continuum equations which are not Galilean invariant. In this current numerical example, we therefore compare our algorithm to a similar scheme which results in a Galilean variant (GV) momentum conservation equation, without the qu_i-term.

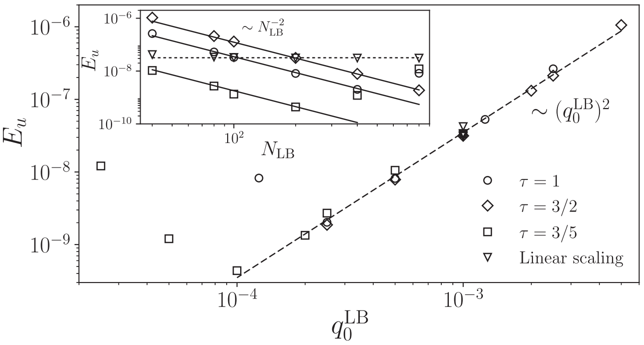

FIG. 2. Plots of the stationary state relative error E_u, emerging in the 1D simulations, calculated using Eq. (19). The inset shows the error as a function of system size N_{LB}. The solid lines there show slopes $\sim N_{\text{LB}}^{-2}$. The main plot shows the error as a function of q_0^{LB} given in lattice units. The dashed line shows a slope $\sim (q_0^{\text{LB}})^2$. All the parameter sets used in these simulations are presented in Table II.

To underline the importance of securing Galilean invariant fluid dynamical equations, we will now let the 1D source and sink described in Sec. IV A 1 move at a constant speed v_0. This means that the source will at time t be centered around a position $x^{\text{src}}(t) = x_0^{\text{src}} + v_0 t$, where $x_0^{\text{src}} = N_{\text{LB}}/2$, while the sink will be centered around a position $x^{\text{sink}}(t) = x_0^{\text{sink}} + v_0 t$, where $x_0^{\text{sink}} = 0$. If we, in addition, initialize the fluid velocity in the system to be v_0 in the same direction as the velocity of the source, we know, from the principle of Galilean relativity, that a coordinate transformation to a frame of reference where the source and sink are at rest should yield the same physical behavior as observed in the system where the source and sink are static.

In the Galilean variant algorithm we compare with, the mass-source dependent term presented in Eq. (10) is replaced by

$$\Delta\widetilde{\Omega}_\alpha^q = w_\alpha\left(1 - \frac{1}{2\tau}\right)\left(1 - \frac{Q_{\alpha ij}u_i u_j}{2c_s^4}\right)q\Delta t \tag{20}$$

and the momentum density is now given through $\sum_\alpha f_\alpha(\mathbf{x},t)c_{\alpha i} = \rho(\mathbf{x},t)u_i(\mathbf{x},t) - F_i(\mathbf{x},t)\Delta t/2$. In the current numerical example, $F_i(\mathbf{x},\ t) = 0$. These two changes lead, through a Chapman-Enskog expansion, to a momentum conservation equation where the qu_i-term is lacking.

All the simulations in this example have been conducted with a relaxation parameter $\tau = 1$. For the different lattice resolutions we have, as in the previous numerical example, used diffusive scaling of the numerical parameters in order to simulate the same physical system. These parameters are presented in Table III.

Figure 3 shows the density profiles at three different times during steady state, obtained for the highest grid resolution ($N_{\text{LB}} = 800$). The profiles obtained using our Galilean invariant (GI) scheme are shown in solid lines, while the results from the GV scheme are shown in dotted lines. First of all, we see that the source and sink generate, respectively, a pronounced increase and a decrease in density around the centers $x^{\text{src}}(t)$ and $x^{\text{sink}}(t)$, at points where $q(x,t)$ is non-zero. And, as the source and sink move in the positive x-direction, we see that this increase and decrease move with the source and sink. However, from the inset of Fig. 3, we see that, while our GI scheme leaves the densities, at points where no source or sink act,

TABLE III. The parameter sets used in the 1D simulations presented in Sec. IV A 2 where the source and sink are moving with a constant speed v_0^{LB}. In all these simulations, $\tau = 1$. All quantities are given in LB units.

N_{LB}	q_0^{LB}	v_0^{LB}	T_{LB}
40	$(5/2) \times 10^{-3}$	1.0×10^{-2}	8.0×10^3
80	$(5/4) \times 10^{-3}$	5.0×10^{-3}	3.2×10^4
100	1.0×10^{-3}	4.0×10^{-3}	5.0×10^4
200	$(1/2) \times 10^{-3}$	2.0×10^{-3}	2.0×10^5
400	$(1/4) \times 10^{-3}$	1.0×10^{-3}	8.0×10^5
800	$(1/8) \times 10^{-3}$	5.0×10^{-4}	3.2×10^6

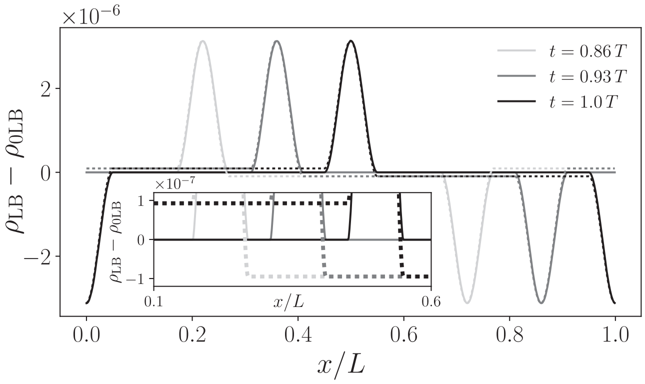

FIG. 3. Plot showing the deviation of the LB density $\rho_{\rm LB}$ from the initial density $\rho_{0\rm LB}$ as a function of the relative position in the system. The density profiles are from simulations where $\tau = 1$ and $N_{\rm LB} = 800$. The different shades of gray represent the density profiles at three different times. The profiles from the Galilean invariant source term are presented as solid lines, while the profiles obtained having a Galilean variant source term are presented as dotted lines. The inset highlights the larger density deviations obtained having a Galilean variant source term.

nearly unaltered, the GV scheme results in a consistent shift in the density at these points. We see, from the dotted lines in the inset, that the densities at points trailing behind the source are raised, while the densities at points ahead of the source are lowered. This is an undesired artifact from the lack of Galilean invariance.

To see the spatial resolution dependence of the observed densities, we look the average relative density error at time $T_{\rm LB}$ (see Table III),

$$E_\rho = \frac{\sum_j |\rho(x_j) - \rho_0(x_j)|}{\sum_j |\rho_0(x_j)|}, \tag{21}$$

where $x_j \in \{x|\ q(x) = 0\}$. In our simulations, $\rho_{0\rm LB} = 1$. In Fig. 4, we show the density error for all the simulations conducted. And we compare the results to the corresponding results obtained for the static case, presented in Sec. IV A 1. As stated in Sec. IV A 1, we see from the dashed line in the main plot of Fig. 4 that $E_\rho \sim (q_0^{\rm LB})^2$ for the simulations with a static source. The results obtained from the GI scheme show from the inset that $E_\rho \sim N_{\rm LB}^4$ initially, before the error converges toward the same absolute error as for the static source for higher lattice resolutions. The data set extracted from the GV simulations, on the other hand, shows that $E_\rho \sim N_{\rm LB}^2$ as for the static source simulations. However, the absolute error for the GV scheme is consistently approximately two orders of magnitude larger. The errors for the GV scheme and the static source case having the same slopes are consistent with the local densities obtained in the GV scheme converging, with increasing spatial resolution, toward a density which differs from ρ_0.

If we now look at the fluid velocities observed in the frame of reference where the source and sink are at rest, i.e., we subtract the source and sink velocity v_0, we can compare them with the velocity results obtained from the static source simulations. Figure 5 shows the average relative velocity error computed using Eq. (19) at time $T_{\rm LB}$. We see that while the velocity error for the GI and GV schemes are approximately the same for lower spatial resolutions, the errors observed in the GI simulations decrease faster with increasing spatial resolution. Note that the definition of momentum density in the GV algorithm does intrinsically not take into account momentum changes due to a source. This is favorable in physicals systems where the overall momentum of the system is conserved, due to the fact that the same amount of fluid with equal velocity is being injected as what is being withdrawn. In general, this is not the case.

We also note that, in our GI algorithm, conservation of momentum is only valid up to the accuracy of the numerical scheme. And, in the specific system simulated, we do notice a drift in momentum with time. This drift decreases with increasing grid resolution. In systems, like the current, without viscous dissipation such errors are not damped. Usually systems have to be driven by external forces to maintain the momentum. We therefore do not regard this drift as a concern.

3. 2D potential flow example

Let us now consider the two-dimensional version of the potential flow described in Sec. IV A 1. Our system is now an

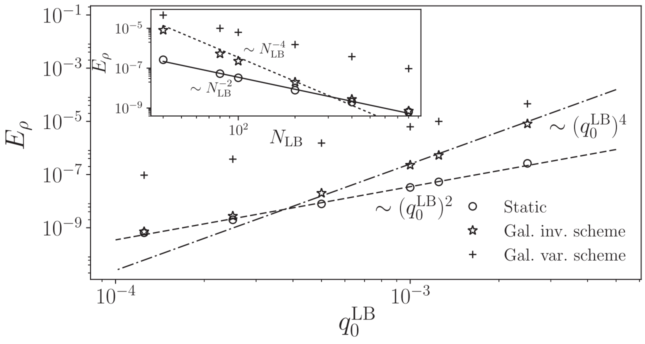

FIG. 4. Plots of the stationary state relative error E_ρ, emerging in the 1D simulations where the source and sink are moving at a constant speed, calculated using Eq. (21). The inset shows the error as a function of system size $N_{\rm LB}$. The solid line there shows a slope $\sim N_{\rm LB}^2$, while the dotted line shows a slope $\sim N_{\rm LB}^4$. The main plot shows the error as a function of $q_0^{\rm LB}$ given in lattice units. The dashed line shows a slope $\sim (q_0^{\rm LB})^2$, while the dashed-dotted line shows a slope $\sim (q_0^{\rm LB})^4$. All the parameter sets used in these simulations are presented in Table III.

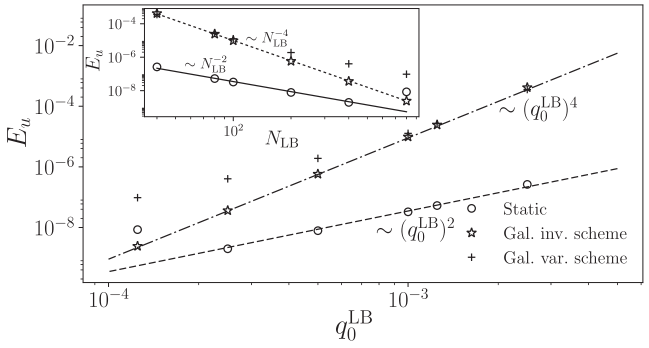

FIG. 5. Plots of the stationary state relative error E_u, emerging in the 1D simulations where the source and sink are moving at a constant speed, calculated using Eq. (19). The inset shows the error as a function of system size $N_{\rm LB}$. The solid line there shows a slope $\sim N_{\rm LB}^{-2}$, while the dotted line shows a slope $\sim N_{\rm LB}^{-4}$. The main plot shows the error as a function of $q_0^{\rm LB}$ given in lattice units. The dashed line shows a slope $\sim (q_0^{\rm LB})^2$, while the dashed-dotted line shows a slope $\sim (q_0^{\rm LB})^4$. All the parameter sets used in these simulations are presented in Table III.

TABLE IV. The parameter sets used in the 2D source-term simulations presented in Sec. IV A 3. All quantities are given in LB units.

	Diffusive scaling			Linear scaling	
	$\tau = 3/5$	$\tau = 1$	$\tau = 3/2$		
N_{LB}	q_0^{LB}	q_0^{LB}	q_0^{LB}	τ	q_0^{LB}
40	2.0×10^{-4}	1.0×10^{-3}	2.0×10^{-3}	0.7	4.0×10^{-4}
80				0.9	8.0×10^{-4}
100				1	1.0×10^{-3}
200				1.5	2.0×10^{-3}
400	…	…	…	2.5	4.0×10^{-3}

$L \times L$ box. The system is again spatially periodic. In the simulations, we have centered the sink around $x^{\text{sink}} = y^{\text{sink}} = 0$, while the source is centered around the midpoint of the system at $x^{\text{src}} = y^{\text{src}} = N_{\text{LB}}/2$. The number of lattice points N_{LB} is, in the same manner as in the 1D example, varied to study the effect of spatial resolution on the numerical results. The numerical parameters used in these simulations are given in Table IV.

To estimate an average relative error for the stationary velocity field, we calculate the mass flux from the source to the sink in the lower left quadrant of system and compare it to the theoretical mass flux in this quadrant. This is simply done by calculating the flux through lattice sites on the line l dividing this quadrant in two equal parts and having a line normal $\mathbf{n}(\mathbf{x}) = -(\mathbf{e}_x + \mathbf{e}_y)$. Figure 6 shows a schematic representation of the location of this line. We define, for the stationary velocity field, an average relative error

$$E_u = \frac{\sum_j |\mathbf{u}(\mathbf{x}_j) \cdot \mathbf{n}(\mathbf{x}_j) - q_0^{\text{LB}}/(4\rho_{0\text{LB}})|}{q_0^{\text{LB}}/(4\rho_{0\text{LB}})}, \tag{22}$$

where j runs over all lattice sites positioned on line l.

Figure 7 shows the relative errors E_u for all the sets of simulations listed in Table IV. From the inset of Fig. 7, we see that the error E_u for the simulation sets with fixed τ values is, as for the 1D case, inversely proportional to the square of the system size N_{LB}. For the set of simulations exhibiting linear scaling, the calculated relative errors are again independent of grid resolution. From the main plot of Fig. 7, we see that now $E_u \sim (q_0^{\text{LB}}/N_{\text{LB}})^2$. It can be shown that the maximal velocity, through line l, goes as $q_0^{\text{LB}}/N_{\text{LB}}$. This means that $E_u \sim [\max(u_{\text{LB}})]^2$, similar to the 1D example in Sec. IV A 1. And, in the same manner as in the 1D example, we attribute the observed errors to the inherent compressibility of the standard LBGK model.[28]

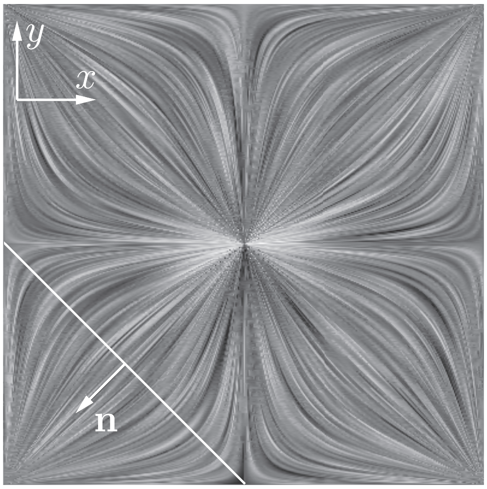

FIG. 6. Plot accentuating the velocity stream lines in the 2D system presented in Sec. IV A 3. This figure is obtained using a so-called line integral convolution (LIC) scheme.[29] (This representation does not give any indication about the speeds along the streamlines.)

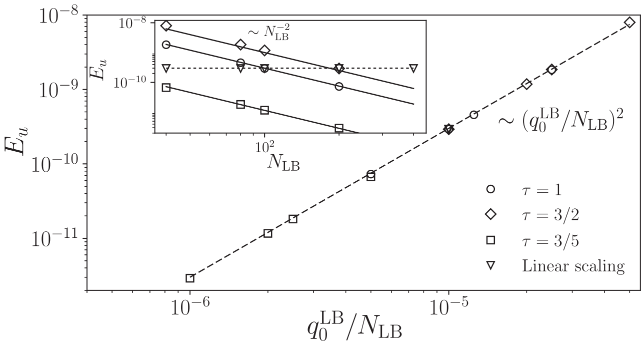

FIG. 7. The stationary state error E_u, emerging in the 2D simulations, found using Eq. (19). The inset shows the error as a function of the system size N_{LB}. The solid lines there show slopes $\sim N_{\text{LB}}^{-2}$. The main plot shows the error as a function of $q_0^{\text{LB}}/N_{\text{LB}}$ given in lattice units. The dashed line shows a slope $\sim (q_0^{\text{LB}}/N_{\text{LB}})^2$. All the parameter sets used in these simulations are presented in Table IV.

B. A mass source as a constant pressure boundary condition

As presented at the end of Sec. III, in Eq. (14), the current mass source term presumptively allows us to set the pressure at any preferred lattice site. We will in the following numerical example use this to fix inlet and outlet pressures in a channel of spatially varying width. From the Navier-Stokes equations, it is easily seen that, for incompressible Newtonian fluid flows, a homogeneous body force $\mathbf{F}$ contributes to the velocity field in the same manner as a constant pressure gradient. With this in mind, the stationary flow results, obtained from our pressure-driven simulations, are compared to equivalent flow results obtained using a constant homogeneous body force. This is to say, we compare two inherently weakly compressible numerical schemes, on the basis that these simulate the same incompressible flow in the low-Mach-number limit. In order to compare, the pressure field in the force-driven simulations must be calculated as $p_{\text{F}}(\mathbf{x}) = p_{\text{F}}^{\text{sim}}(\mathbf{x}) + \mathbf{F} \cdot (\mathbf{x}_{\text{out}} - \mathbf{x})$, following the Navier-Stokes equations and assuming that at the outlet $p_{\text{F}}(\mathbf{x}_{\text{out}}) = p_{\text{F}}^{\text{sim}}(\mathbf{x}_{\text{out}})$. In our simulations $p_{\text{F}}^{\text{sim}}(\mathbf{x}) = c_s^2 \rho(\mathbf{x})$. A fundamental distinction between the two driving mechanisms is that, in order to have flow, the pressure-driven system relies on a non-zero local pressure gradient, and, through the ideal gas equation of state in our LB model, a non-zero density gradient. In the force-driven system, this is not the case. Here, density variation effects are minimal and only a consequence of system geometry, not the driving mechanism, which is important to have in mind when comparing the two sets of simulations.

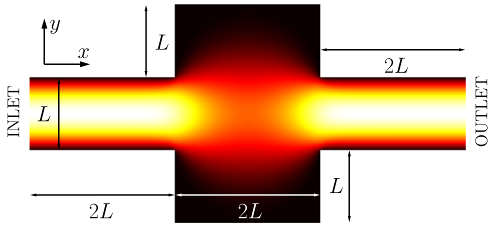

FIG. 8. An illustration of the stationary absolute velocity field observed in the simulations presented in Sec. IV B. Here, lighter shades represent higher speeds and darker shades represent lower speeds. This system of $N_{x\mathrm{LB}} \times N_{y\mathrm{LB}}$ lattice points has an overall length of $6L$ and an overall width of $3L$.

Figure 8 shows the geometry and dimensions of the channel used in these flow simulations. The no-slip boundary conditions at the walls are here implemented through a mid-grid bounce-back condition.[30,31]

In the pressure-driven simulations, to ensure correct distribution functions f_α arriving at inlet and outlet nodes from outside the system, a form of extrapolation is necessary. To do this, we define a set of ghost nodes, positioned outside the simulation domain, adjacent to the inlet and outlet nodes. Before the collision step, when the macroscopic quantities mass density ρ, density flux $\rho\mathbf{u}$, and the source term q are calculated, we calculate through an extrapolation scheme the distribution functions for these ghost nodes. For the ghost nodes adjacent to, respectively, the inlet and outlet nodes,

$$f_\alpha(y, x_{\mathrm{in}} - 1) = f_\alpha(y, x_{\mathrm{in}}) + f_\alpha^{\mathrm{eq}}(\Delta_{\mathrm{in}}, u(y, x_{\mathrm{in}})) \tag{23}$$

and

$$f_\alpha(y, x_{\mathrm{out}} + 1) = f_\alpha(y, x_{\mathrm{out}}) - f_\alpha^{\mathrm{eq}}(\Delta_{\mathrm{out}}, u(y, x_{\mathrm{out}})), \tag{24}$$

where $\Delta_{\mathrm{in}} = \sum_\alpha f_\alpha(y, x_{\mathrm{in}}) - \sum_\alpha f_\alpha(y, x_{\mathrm{in}} + 1)$ and $\Delta_{\mathrm{out}} = \sum_\alpha f_\alpha(y, x_{\mathrm{out}} - 1) - \sum_\alpha f_\alpha(y, x_{\mathrm{out}})$. After this extrapolation, the ghost nodes are treated as regular lattice nodes that undergo the collision and propagation steps. This will ensure that a set of appropriate distribution functions arrive at the inlet and outlet nodes from outside the simulation domain after propagation.

The pressures at the inlet and outlet nodes are fixed through the source term introduced in Eq. (14). In the same manner are the pressures at the ghost nodes, adjacent to the inlet and outlet nodes, fixed to, respectively, $p(y, x_{\mathrm{in}} - 1) = p(y, x_{\mathrm{in}}) + c_s^2 \Delta_{\mathrm{in}}$ and $p(y, x_{\mathrm{out}} + 1) = p(y, x_{\mathrm{out}}) - c_s^2 \Delta_{\mathrm{out}}$. This leads to, per definition, fixed macroscopic mass densities (and pressures) calculated at the inlet and outlet, as well as at the corresponding ghost nodes.

It is important to note that if the constant pressure nodes are not initialized to the desired pressure, a persistent oscillating pressure signal, or the so-called ringing, may arise from these nodes as a response to the sudden change in input.

In the current numerical example, the inlet and outlet nodes are initialized to have the densities corresponding to the desired constant pressures. To demonstrate how our source-term approach to pressure boundary conditions performs, we compare the pressure-driven simulations against force-driven simulations. The magnitude of the constant body force F_{LB} is put equal to the magnitude of the average pressure gradient $(p_{\mathrm{in}} - p_{\mathrm{out}})/(N_{\mathrm{LB}x} - 1)$ across the pressure-driven system. The force-driven system is periodic in the flow direction. Table V shows all the parameter sets used in these simulations. The simulations are run until they reach a stationary state and are stopped after T_{LB} time steps (see Table V). In this state, we have compared the results of the pressure-driven and force-driven simulations.

The pressure-driven simulations are in excellent agreement with the force-driven simulations. Figure 9 shows a comparison of the pressure profiles over the length of the system, along the mid-channel line. The upper panel of Fig. 9 shows the local relative difference $\lambda_p(x) = p_{\mathrm{P}}(x)/p_{\mathrm{F}}(x) - 1$ between the pressure-driven simulations and the force-driven ones. We see that this difference decreases in the direction of the outlet side. This is consistent with having close to equal fluid densities $\rho_{\mathrm{LB}}(\mathbf{x})$, at the outlet, in the two types of simulations.

As a global measure for the difference in pressure between the simulations, we define a stationary state average relative pressure difference

$$\Lambda_p = \frac{\sum_j |p_{\mathrm{P}}(\mathbf{x}_j) - p_{\mathrm{F}}(\mathbf{x}_j)|}{\sum_j |p_{\mathrm{F}}(\mathbf{x}_j)|}. \tag{25}$$

Figure 10 shows this relative difference Λ_p for all our simulations for all the parameter sets (given in Table V). From the inset of this figure, we see that, for the simulation sets with fixed τ values (diffusive scaling), Λ_p is close to inversely proportional to the square of the system size $N_{x\mathrm{LB}}$. However, at a closer inspection it can be observed that Λ_p decreases marginally, but consistently, faster than $N_{x\mathrm{LB}}^{-2}$. The main plot of Fig. 10 shows the relative difference Λ_p as a function of the maximum velocity $\max(u_{\mathrm{LB}})$ given in lattice units. It is seen that the difference in pressure behavior between the two types of simulations, to a large degree, is governed by the LB velocity u_{LB} of the system alone. The difference Λ_p is compared

TABLE V. The parameter sets used in the pressure boundary simulations presented in Sec. IV B. All quantities are given in LB units.

	Diffusive scaling								
	$\tau = 3/5$		$\tau = 1$		$\tau = 3/2$		Linear scaling		
$N_{x\mathrm{LB}}$	T_{LB}	F_{LB}	T_{LB}	F_{LB}	T_{LB}	F_{LB}	τ	T_{LB}	F_{LB}
30	1×10^4	1.0×10^{-5}	2×10^3	2.5×10^{-4}	1×10^3	1.0×10^{-3}	0.6	1×10^4	1.0×10^{-5}
60	4×10^4	1.25×10^{-6}	8×10^3	3.125×10^{-5}	4×10^3	1.25×10^{-4}	0.7	2×10^4	5.0×10^{-6}
120	1.6×10^5	1.5625×10^{-7}	3.2×10^4	$3.906\,25 \times 10^{-6}$	1.6×10^4	1.5625×10^{-5}	0.9	4×10^4	2.5×10^{-6}
150	2.5×10^5	8.0×10^{-8}	5×10^4	2.0×10^{-6}	2.5×10^4	8.0×10^{-6}	1	5×10^4	2.0×10^{-6}
300	1×10^6	1.0×10^{-8}	2×10^5	2.5×10^{-7}	1×10^5	1.0×10^{-6}	1.5	1×10^5	1.0×10^{-6}

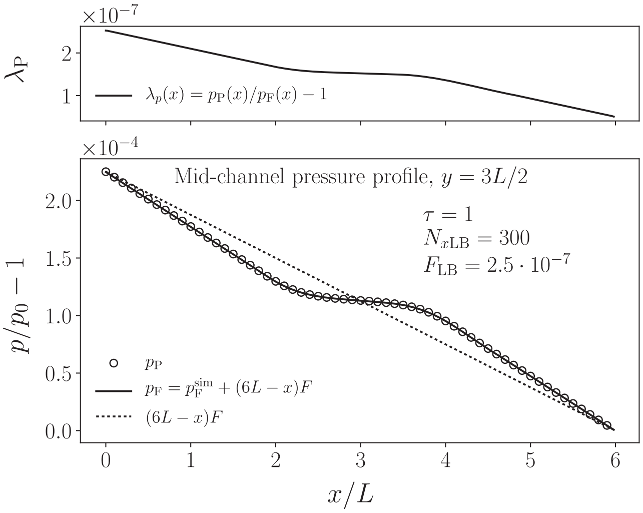

FIG. 9. Plots comparing pressures obtained from pressure-driven and force-driven simulations. The lower panel shows the pressure profile (in lattice units with $p_0 = 1/3$) at $y = 3L/2$ (see Fig. 8), for the simulations having $\tau = 1$ and $N_{x\mathrm{LB}} = 300$. Every 5th data point from the pressure-driven simulation is plotted as an open circle, while the data from the force-driven one is given as a continuous line. The dotted line represents the average pressure gradient across the system. The upper panel presents the corresponding local relative difference λ_p between the two simulations.

to a solid line ~ $[\max(u_{\mathrm{LB}})]^2$. Arguably, Λ_p increases slightly faster than this line. As stated earlier, it is commonly accepted that the error due to compressibility in the standard LBGK model goes as the Mach number squared.[28] If we assume that this is also valid for our current pressure-driven model, we would expect that the order of the difference between the simulations should be Mach number squared or higher. In Fig. 10, this is exactly what we see. This is consistent with Λ_p being controlled by the compressibility effects present in the simulations.

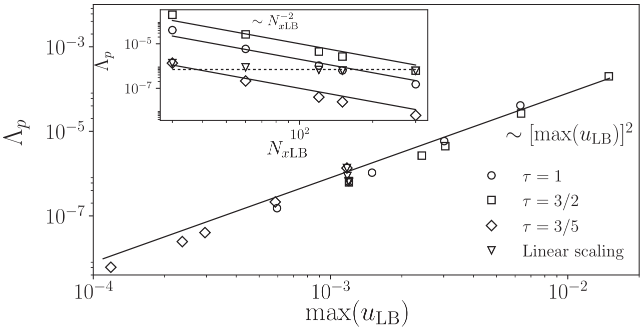

FIG. 10. Plots of the average relative pressure difference Λ_p, defined in Eq. (25), between the stationary pressures in the pressure-driven simulations and the force-driven ones. The inset shows Λ_p as a function of the system length $N_{x\mathrm{LB}}$ given in lattice units. The solid lines there show slopes $\sim N_{\mathrm{LB}}^{-2}$. The main plot shows Λ_p as a function of the maximum velocity $\max(u_{\mathrm{LB}})$ given in lattice units. The solid line shows a slope ~ $[\max(u_{\mathrm{LB}})]^2$. All the parameter sets used in the plot are presented in Table V.

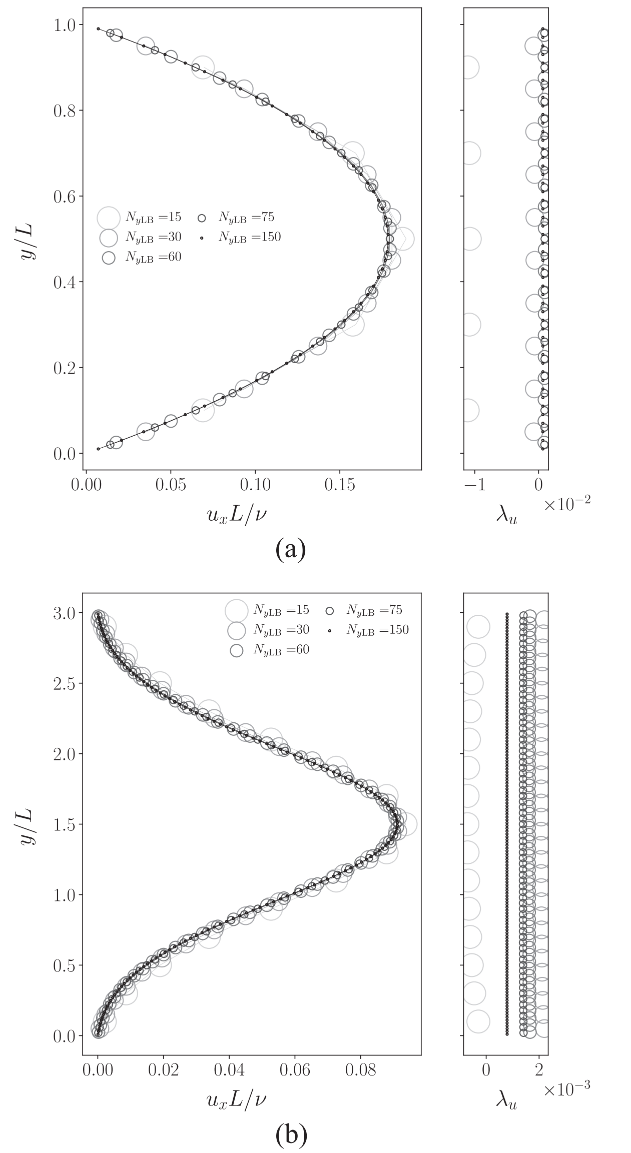

FIG. 11. Plots comparing velocity profiles obtained from the pressure-driven and force-driven simulations with relaxation parameter $\tau = 1$. (a) shows the inlet cross section at $x = 0$, while (b) shows the cross section half way between the inlet and the outlet of the system at $x = 3L$. The figures on the left-hand side show a dimensionless representation of the x-component of the velocities obtained for the different lattice resolutions. The data from the pressure-driven simulations are represented by open circles, while the results from the force-driven simulations are represented by solid lines in the same shade of gray as the corresponding pressure-driven plots. The figures on the right-hand side show the corresponding local relative differences λ_u between the pressure-driven and force-driven simulations.

Figure 11 shows a comparison of velocities resulting from the pressure-driven and force-driven simulations with relaxation parameter $\tau = 1$. The plots on the left show the velocity profiles obtained in the different simulations, while the narrow panels on the right show the local relative difference $\lambda_u(y) = u_x^{\mathrm{P}}(y)/u_x^{\mathrm{F}}(y) - 1$ between the pressure-driven and the force-driven simulations. From Fig. 11(a), we see that the velocity profiles obtained at the inlet ($x = 0$) are parabolic in shape. It can be shown that these velocity profiles agree with a plane Poiseuille flow having a mid-channel velocity $-L^2\partial_x p/(8\rho\nu)$, where $\partial_x p$ is the local pressure gradient measured at the inlet. In Fig. 11, both pressure-driven and force-driven simulations generally yield higher local velocities for lower lattice resolutions compared to higher resolutions. The results obtained by the two types of simulations agree well overall. The right-hand panels in Fig. 8 show in general a decrease in the absolute value of the local relative difference

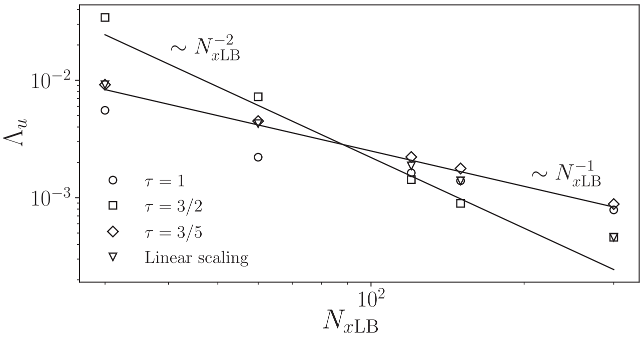

FIG. 12. Plot of the average relative velocity difference Λ_u between the stationary velocities in pressure-driven and force-driven simulations. The difference Λ_u, defined in Eq. (26), is here plotted as a function of the system length $N_{x\mathrm{LB}}$ given in lattice units. All the parameter sets used in these simulations are presented in Table V.

λ_u with increasing lattice resolution. Comparing the top and bottom rightmost panels, it is also observed that the local relative differences λ_u are smaller in the wide part of the channel ($x = 3L$), where the velocities are lower and the pressure gradients are smaller than in the narrow part of the channel. This indicates that compressibility effects also influence the velocity differences observed when comparing the pressure-driven and force-driven simulations.

As a global stationary state measure of the velocity difference between the two types of simulations, we define an average relative velocity difference

$$\Lambda_u = \frac{\sum_j |\mathbf{u}^{\mathrm{P}}(\mathbf{x}_j) - \mathbf{u}^{\mathrm{F}}(\mathbf{x}_j)|}{\sum_j |\mathbf{u}^{\mathrm{F}}(\mathbf{x}_j)|}. \tag{26}$$

Figure 12 shows this relative velocity difference for all the simulation sets presented in Table V as a function of numerical resolution. From this figure we see that the average relative velocity difference Λ_u decreases with increasing lattice resolution. But contrary to Λ_p shown in Fig. 10, Λ_u is not observed to exhibit a simple dependence on the system size $N_{x\mathrm{LB}}$. The results may indicate, to a certain extent, that the behavior of Λ_u is dependent of the relaxation parameter τ.

Overall, using the mass source term as a way to implement constant pressure boundaries has shown that it performs well compared to the equivalent force-driven flow.

V. CONCLUSION

We present in this paper a lattice Boltzmann algorithm for incorporating a mass source term in the fluid dynamical continuum equations that conserves the overall accuracy of the method. This is shown through a Chapman-Enskog expansion of the method (found in the Appendix). We discuss how including a mass source alters the fluid dynamical continuum equation and point out that these equations are Galilean invariant. We have in this paper explicitly shown how to include such a mass source term in a LBGK model.

The performance of the algorithm is demonstrated by four numerical examples which all are in excellent agreement with the expected results.

The first three numerical examples demonstrate one of the main strengths of the method, namely, fluid injection and withdrawal at a given rate. Here, we have simulated potential flows generated by a set of spatially inhomogeneous sources and sinks. These simulations have been directly compared with analytical results. The observed errors in the static source simulations are attributed the inherent compressibility of the model, a feature shared by all the standard LBGK models. In a set of simulations, we let the source and sink move at a constant speed. These simulations were compared to a set of simulations where Galilean invariance was not maintained. We observed a clear erroneous shift in the densities due to the lack of Galilean invariance. Such a discrepancy was not observed in the simulations conducted using the mass source term introduced in this paper.

The fourth and final numerical example demonstrates the mass source term used as a means to implement fixed pressures at preferred locations. Here, the source term fixes the inlet and outlet pressures of a channel with spatially varying width. An extrapolation scheme for handling the LB distribution functions arriving from outside the simulation domain is in this case necessary. In this sense, our algorithm is as complex as other boundary condition schemes. The results obtained have been compared to simulations of the equivalent fluid flow driven by a constant homogeneous body force. The comparison shows an excellent agreement between these simulations.

The final numerical example investigated here shows that the algorithm is capable of fixing the pressure at a preferred lattice site. And, in general, this does not implicate an extrapolation scheme. However, caution should be exercised since the operation of fixing the pressure relies on a continuous adjustment of the source term to balance the distribution function behavior at the site. Such a feedback interaction between the source term and the LB distribution functions is not present when using the source term to inject or withdraw fluid at a given rate.

As a final remark, preliminary results indicate that the proposed mass source term successfully handles both time dependent injection and withdrawal rates and multi-phase flows in porous structures.

ACKNOWLEDGMENTS

The presented work was financially supported by the Research Council of Norway, through CLIMIT Project No. 255572/E20, and by the National IOR Centre of Norway. The authors thank the National IOR Centre of Norway industry partners: ConocoPhillips Skandinavia AS, Aker BP ASA, Eni Norge AS, Maersk Oil Norway AS, DONG Energy A/S, Denmark, Statoil Petroleum AS, ENGIE E&P NORGE AS, Lundin Norway AS, Halliburton AS, Schlumberger Norge AS, and Wintershall Norge AS, for their support.

APPENDIX: CHAPMAN-ENSKOG EXPANSION

Having a lattice Boltzmann equation (LBE)

$$f_\alpha(\mathbf{x} + \mathbf{c}_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{\mathrm{eq}}\right) + \Delta\Omega_\alpha, \tag{A1}$$

where $\Delta\Omega_\alpha(\mathbf{x}, t)$ is given by Eqs. (9) and (10), performing a Chapman-Enskog expansion will recover the desired Navier-Stokes equations presented in Eqs. (1) and (2).

This may be done by introducing an expansion parameter ε such that

$$f_\alpha = \sum_{n=0}^{\infty} \varepsilon^n f_\alpha^{(n)}, \tag{A2}$$

$$\partial_t = \varepsilon\partial_{t1} + \varepsilon^2\partial_{t2}, \quad \text{and} \quad \partial_i = \varepsilon\partial_{i1}. \tag{A3}$$

The parameter ε is often taken to be proportional to the Knudsen number which is here the ratio of the lattice spacing to a characteristic macroscopic length scale.

The Taylor series

$$f_\alpha(\mathbf{x} + \mathbf{c}_\alpha\Delta t, t + \Delta t) = \sum_{m=0}^{\infty} \frac{(\Delta t)^m}{m!} D_\alpha^m f_\alpha(\mathbf{x}, t), \tag{A4}$$

where we have defined

$$D_\alpha = \varepsilon D_{\alpha 1} + \varepsilon^2 D_{\alpha 2} = \varepsilon(\partial_{t1} + c_{\alpha i}\partial_{i1}) + \varepsilon^2\partial_{t2}. \tag{A5}$$

This derivative should not to be confused with the macroscopic material derivative $D_t \equiv (\partial_t + u_i\partial_i)$. This gives, through second order in Δt, the following hierarchy of LBEs at increasing orders of ε:

$$\mathcal{O}(\varepsilon^0) : f_\alpha^{(0)} = f_\alpha^{\text{eq}}, \tag{A6}$$

$$\mathcal{O}(\varepsilon^1) : \Delta t D_{\alpha 1} f_\alpha^{(0)} = -\frac{1}{\tau} f_\alpha^{(1)} + \Delta\Omega_\alpha^{(1)}, \tag{A7}$$

$$\mathcal{O}(\varepsilon^2) : \Delta t D_{\alpha 2} f_\alpha^{(0)} = -\frac{1}{\tau} f_\alpha^{(2)} + \Delta\Omega_\alpha^{(2)} - \Delta t D_{\alpha 1}\left[\left(1 - \frac{1}{2\tau}\right) f_\alpha^{(1)} + \frac{1}{2}\Delta\Omega_\alpha^{(1)}\right]. \tag{A8}$$

In the following we will also need that

$$\sum_\alpha w_\alpha = 1, \tag{A9}$$

$$\sum_\alpha w_\alpha c_{\alpha i} c_{\alpha j} = c_s^2 \delta_{ij}, \tag{A10}$$

$$\sum_\alpha w_\alpha c_{\alpha i} c_{\alpha j} c_{\alpha k} c_{\alpha l} = c_s^4(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \tag{A11}$$

Sums over odd numbers of the velocity vectors $\mathbf{c}_\alpha$ are equal to zero.

A direct sum of Eq. (A6) over all lattice directions combined with Eq. (7) yields

$$\sum_\alpha f_\alpha^{(0)} = \rho \tag{A12}$$

and comparing to Eq. (11)

$$\sum_\alpha f_\alpha^{(n)} = -\frac{1}{2} q^{(n)}\Delta t, \quad \text{for} \quad n \geq 1. \tag{A13}$$

In the same manner, a weighted sum of Eq. (A6) using $c_{\alpha i}$ as weights leads to

$$\sum_\alpha f_\alpha^{(0)} c_{\alpha i} = \rho u_i, \tag{A14}$$

and comparing to Eq. (12)

$$\sum_\alpha f_\alpha^{(n)} c_{\alpha i} = -\frac{1}{2}(F_i^{(n)} + q^{(n)} u_i)\Delta t, \quad \text{for} \quad n \geq 1. \tag{A15}$$

Consistent with Eq. (A6), $q^{(0)} = 0$ and $F_i^{(0)} = 0$. We will further need the two additional results,

$$\sum_\alpha f_\alpha^{(0)} c_{\alpha i} c_{\alpha j} = c_s^2 \rho\delta_{ij} + \rho u_i u_j, \tag{A16}$$

$$\sum_\alpha f_\alpha^{(0)} c_{\alpha i} c_{\alpha j} c_{\alpha k} = c_s^2 \rho(\delta_{ij} u_k + \delta_{ik} u_j + \delta_{jk} u_i). \tag{A17}$$

And, from Eqs. (9) and (10), combined with Eq. (8), we have that

$$\sum_\alpha \Delta\Omega_\alpha^{(n)} = \left(1 - \frac{1}{2\tau}\right) q^{(n)}\Delta t, \tag{A18}$$

$$\sum_\alpha \Delta\Omega_\alpha^{(n)} c_{\alpha i} = \left(1 - \frac{1}{2\tau}\right)\left(F_i^{(n)} + q^{(n)} u_i\right)\Delta t, \tag{A19}$$

$$\sum_\alpha \Delta\Omega_\alpha^{(n)} c_{\alpha i} c_{\alpha j} = \left(1 - \frac{1}{2\tau}\right)\left(u_i F_j^{(n)} + u_j F_i^{(n)}\right)\Delta t + \left(1 - \frac{1}{2\tau}\right)(c_s^2\delta_{ij} + u_i u_j) q^{(n)}\Delta t. \tag{A20}$$

A direct sum of Eq. (A7) over all the lattice directions, combined with Eqs. (A18) and (A19), yields

$$\partial_{t1}\rho + \partial_{j1}(\rho u_j) = q^{(1)}. \tag{A21}$$

A weighted sum of Eq. (A7) using $c_{\alpha i}$ as weight leads to

$$\partial_{t1}(\rho u_i) + \partial_{j1}(c_s^2 \rho\delta_{ij} + \rho u_i u_j) = F_i^{(1)} + q^{(1)} u_i. \tag{A22}$$

Identifying $p = \rho c_s^2$ as the pressure, which is consistent with c_s^2 being the lattice speed of sound squared, shows that the previous equation may be considered the momentum conservation equation for an inviscid fluid, where $c_s^2\rho\delta_{ij} + \rho u_i u_j$ is identified as the component form of the inviscid momentum flux tensor $\Pi_{ij}^{(0)}$.

The direct sum of Eq. (A8), combined with Eqs. (A18) and (A19), gives

$$\partial_{t2}\rho = q^{(2)}. \tag{A23}$$

The weighted sum of Eq. (A8), using $c_{\alpha i}$ as weight, yields

$$\partial_{t2}(\rho u_i) = F_i^{(2)} + q^{(2)} u_i - D_{\alpha 1}\sum_\alpha\left[\left(1 - \frac{1}{2\tau}\right) f_\alpha^{(1)} + \frac{1}{2}\Delta\Omega_\alpha^{(1)}\right] c_{\alpha i}. \tag{A24}$$

By using Eq. (A7) to express $f_\alpha^{(1)}$, this equation may be rewritten as

$$\partial_{t2}(\rho u_i) = F_i^{(2)} + q^{(2)} u_i - \partial_{j1}\sum_\alpha\left[\left(\frac{1}{2} - \tau\right)\Delta t D_{\alpha 1} f_\alpha^{(0)} + \tau\Delta\Omega_\alpha^{(1)}\right] c_{\alpha i} c_{\alpha j}. \tag{A25}$$

Employing Eq. (A20), combined with the continuity equation from Eq. (A21), the first-order momentum conservation equation from Eq. (A22), and that

$$\partial_{t1}(\rho u_i u_j) = u_i F_j^{(1)} + u_j F_i^{(1)} + u_i u_j q^{(1)} - c_s^2(u_i\partial_{j1}\rho + u_j\partial_{i1}\rho) - \partial_{k1}(\rho u_i u_j u_k), \tag{A26}$$

we have that

$$\partial_{t2}(\rho u_i) = F_i^{(2)} + q^{(2)} u_i + \partial_{j1}\left\{\rho\nu\left[\partial_{i1}u_j + \partial_{j1}u_i - \frac{\partial_{k1}(\rho u_i u_j u_k)}{\rho c_s^2}\right]\right\}, \qquad \text{(A27)}$$

where we also have defined $\nu \equiv c_s^2(\tau - 1/2)\Delta t$. The last unwanted term in Eq. (A27) is $\mathcal{O}(u^3)$ and negligibly small at low velocities compared with the other terms. Observe that the source-and-correction term $\Delta\Omega_\alpha$ results in no unwanted body force or mass source terms in the continuum equations. Bringing the above equations together through $\mathcal{O}(\varepsilon^2)$, using that $\partial_t = \varepsilon\partial_{t1} + \varepsilon^2\partial_{t2}$ and $\partial_i = \varepsilon\partial_{i1}$, gives

$$\partial_t\rho + \partial_i(\rho u_i) = q, \qquad \text{(A28)}$$

$$\rho\Big(\partial_t u_i + u_j\partial_j u_i\Big) = -\partial_i p + F_i + \partial_j\{\rho\nu[\partial_i u_j + \partial_j u_i + \mathcal{O}(u^3)]\}. \qquad \text{(A29)}$$

These are the Navier-Stokes equations we wanted to retrieve. On comparison to Eq. (2) combined with Eq. (3), we see that in this LB model the bulk viscosity would, in a d-dimensional system, be directly related to the shear viscosity as $\xi = 2\rho\nu/d$. It could here be noted that the disregarded $\mathcal{O}(u^3)$ term breaks Galilean invariance. There have been proposed ways to remedy this higher order invariance breakage,[32] but this is outside the scope of the current work.

[1] D. H. Rothman and S. Zaleski, *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics* (Cambridge University Press, 1997).

[2] S. Succi, *The Lattice-Boltzmann Equation* (Oxford University Press, Oxford, 2001).

[3] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The Lattice Boltzmann Method* (Springer, 2017).

[4] C. Pan, M. Hilpert, and C. Miller, "Lattice-Boltzmann simulation of two-phase flow in porous media," Water Resour. Res. **40**, W01501, https://doi.org/10.1029/2003wr002120 (2004).

[5] E. S. Boek and M. Venturoli, "Lattice-Boltzmann studies of fluid flow in porous media with realistic rock geometries," Comput. Math. Appl. **59**, 2305 (2010).

[6] O. Aursjø, G. Løvoll, H. A. Knudsen, E. G. Flekkøy, and K. J. Måløy, "A direct comparison between a slow pore scale drainage experiment and a 2D lattice Boltzmann simulation," Transp. Porous Media **86**, 125 (2011).

[7] O. Aursjø, H. A. Knudsen, E. G. Flekkøy, and K. J. Måløy, "Oscillation-induced displacement patterns in a two-dimensional porous medium: A lattice Boltzmann study," Phys. Rev. E **82**, 026305 (2010).

[8] T. Ramstad, P.-E. Øren, S. Bakke *et al.*, "Simulation of two-phase flow in reservoir rocks using a lattice Boltzmann method," SPE J. **15**, 917 (2010).

[9] O. Aursjø and S. R. Pride, "Lattice Boltzmann method for diffusion-limited partial dissolution of fluids," Phys. Rev. E **92**, 013306 (2015).

[10] A. Ferrari and I. Lunati, "Direct numerical simulations of interface dynamics to link capillary pressure and total surface energy," Adv. Water Resour. **57**, 19 (2013).

[11] F. Jiang and T. Tsuji, "Estimation of three-phase relative permeability by simulating fluid dynamics directly on rock-microstructure images," Water Resour. Res. **53**, 11, https://doi.org/10.1002/2016wr019098 (2017).

[12] Q. Li, K. Luo, Q. Kang, Y. He, Q. Chen, and Q. Liu, "Lattice Boltzmann methods for multiphase flow and phase-change heat transfer," Prog. Energy Combust. Sci. **52**, 62 (2016).

[13] T. Inamuro, M. Yoshino, and F. Ogino, "A non-slip boundary condition for lattice Boltzmann simulations," Phys. Fluids **7**, 2928 (1995).

[14] Q. Zou and X. He, "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model," Phys. Fluids **9**, 1591 (1997).

[15] B. Shi, B. Deng, R. Du, and X. Chen, "A new scheme for source term in LBGK model for convection-diffusion equation," Comput. Math. Appl. **55**, 1568 (2008).

[16] Z. Chai and T. Zhao, "Nonequilibrium scheme for computing the flux of the convection-diffusion equation in the framework of the lattice Boltzmann method," Phys. Rev. E **90**, 013305 (2013).

[17] O. Aursjø, E. Jettestuen, J. L. Vinningland, and A. Hiorth, "An improved lattice Boltzmann method for simulating advective-diffusive processes in fluids," J. Comput. Phys. **332**, 363 (2017).

[18] J. Zhang, "Lattice Boltzmann method for microfluidics: models and applications," Microfluid. Nanofluid. **10**, 1 (2011).

[19] N.-T. Nguyen, M. Hejazian, C. H. Ooi, and N. Kashaninejad, "Recent advances and future perspectives on microfluidic liquid handling," Micromachines **8**, 186 (2017).

[20] I. Halliday, L. Hammond, C. Care, K. Good, and A. Stevens, "Lattice Boltzmann equation hydrodynamics," Phys. Rev. E **64**, 011208 (2001).

[21] Y. Cheng and J. Li, "Introducing unsteady non-uniform source terms into the lattice Boltzmann model," Int. J. Numer. Methods Fluids **56**, 629 (2008).

[22] A. Kuzmin, Z. Guo, and A. Mohamad, "Simultaneous incorporation of mass and force terms in the multi-relaxation-time framework for lattice Boltzmann schemes," Philos. Trans. R. Soc., A **369**, 2219 (2011).

[23] K. Wang, Y. Yu, L. Yang, G. Hou, S. Xu, S. Xu *et al.*, "Lattice Boltzmann based internal wave-maker," in *The 27th International Ocean and Polar Engineering Conference* (International Society of Offshore and Polar Engineers, 2017).

[24] G. K. Batchelor, *An Introduction to Fluid Dynamics* (Cambridge University Press, 2000).

[25] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, 2nd ed. (Butterworth-Heinemann, 1987).

[26] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," Phys. Rev. E **65**, 046308 (2002).

[27] S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases* (Cambridge University Press, 1970).

[28] M. B. Reider and J. D. Sterling, "Accuracy of discrete-velocity BGK models for the simulation of the incompressible Navier-Stokes equations," Comput. Fluids **24**, 459 (1995).

[29] B. Cabral and L. C. Leedom, in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (ACM, 1993), pp. 263–270.

[30] D. P. Ziegler, "Boundary conditions for lattice Boltzmann simulations," J. Stat. Phys. **71**, 1171 (1993).

[31] X. He, Q. Zou, L.-S. Luo, and M. Dembo, "Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model," J. Stat. Phys. **87**, 115 (1997).

[32] P. J. Dellar, "Lattice Boltzmann algorithms without cubic defects in Galilean invariance on standard lattices," J. Comput. Phys. **259**, 270 (2014).