Counter-extrapolation method for conjugate interfaces in computational heat and mass transfer

Guigao Le,[1] Othmane Oulaid,[2] and Junfeng Zhang[2,*]

[1]*School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China*

[2]*Bharti School of Engineering, Laurentian University, 935 Ramsey Lake Road, Sudbury, Ontario P3E 2C6, Canada*

(Received 12 January 2015; revised manuscript received 27 February 2015; published 23 March 2015; corrected 23 October 2015)

In this paper a conjugate interface method is developed by performing extrapolations along the normal direction. Compared to other existing conjugate models, our method has several technical advantages, including the simple and straightforward algorithm, accurate representation of the interface geometry, applicability to any interface-lattice relative orientation, and availability of the normal gradient. The model is validated by simulating the steady and unsteady convection-diffusion system with a flat interface and the steady diffusion system with a circular interface, and good agreement is observed when comparing the lattice Boltzmann results with respective analytical solutions. A more general system with unsteady convection-diffusion process and a curved interface, i.e., the cooling process of a hot cylinder in a cold flow, is also simulated as an example to illustrate the practical usefulness of our model, and the effects of the cylinder heat capacity and thermal diffusivity on the cooling process are examined. Results show that the cylinder with a larger heat capacity can release more heat energy into the fluid and the cylinder temperature cools down slower, while the enhanced heat conduction inside the cylinder can facilitate the cooling process of the system. Although these findings appear obvious from physical principles, the confirming results demonstrates the application potential of our method in more complex systems. In addition, the basic idea and algorithm of the counter-extrapolation procedure presented here can be readily extended to other lattice Boltzmann models and even other computational technologies for heat and mass transfer systems.

DOI: 10.1103/PhysRevE.91.033306 PACS number(s): 47.11.−j, 44.35.+c, 44.05.+e, 47.55.N−

I. INTRODUCTION

Conjugate heat and mass transfer between two materials (typically one solid and one flowing fluid) with different transport properties is a common phenomenon in many natural processes and industrial systems, including heat exchangers, jet engines, nuclear reactors, combustion chambers, and microfuidic devices [1–3]. In addition to the continuity requirement for the transport scalar (temperature for heat transfer or concentration for mass transfer) at the conjugate interface, the heat or mass transport flux across the interface must be balanced according to the energy or mass conservation principle. In recent years, the lattice Boltzmann method (LBM) has become a useful choice for computational simulations of various flow and transport systems [4,5]. Compared to other conventional numerical methods, LBM is advantageous in parallel computation and treating complex boundary geometry. Although various numerical schemes for conjugate interfaces have been developed for traditional numerical methods [6], these schemes were designed for their particular differential equation solvers (including numerical algorithms and mesh structures), and it is difficult to adopt them in LBM directly. In addition, iterative calculations are usually required in these conventional methods, and such iterative schemes may become difficult to implement and computational costly for complex interface geometry [7].

The first attempt of using LBM to simulate conjugate heat transfer was conducted by Wang *et al.* [8], where a half-lattice-division scheme (i.e., the conjugate interface locates on the middle point between two lattice nodes in different domains) was proposed. On-node schemes have also been developed where the conjugate interface is placed on lattice nodes [9,10]. Alternatively, Zhao *et al.* [11] simulated the conjugate heat transfer process in porous wick by simply assigning different thermal diffusivity values to lattice nodes on each side of the conjugate interface. This approach is easy to implement and no particular treatments at the interface is required; however, the volume thermal capacities (product of density and specific heat capacity) in different domains must be the same [11]. In addition, Yoshida *et al.* [12] employed a variable weight coefficient in the equilibrium distribution expression to account different thermal transport coefficients across an interface and scaled the density distributions if crossing the interface. Again, the spatial resolution is limited to the lattice resolution since the exact boundary location is not considered. Furthermore, Karani and Huber [13] mathematically examined the difference in the thermal conduction and mass diffusion equations and added a source term to the diffusion equation to mimic the conduction equation. The source term requires the calculation of the spatial gradient of the reciprocal of the volume thermal capacity, and they have assumed that the interface is located halfway along the intersecting lattice links.

These LBM conjugate models above have not considered the exact interface location, and, therefore, for a curved interface, the interface shape can only be approximated by a series of stairlike segments with a spatial resolution of the lattice grid unit, and technical difficulties could be encountered at corner nodes. Li *et al.* [7] have extended their boundary method for thermal LBM simulations [14] to conjugate interfaces and arbitrary interface geometry can be accurately modeled. Both steady and unsteady, as well as flat and circular, interfaces have been examined, and the

*Corresponding author: Dr. Junfeng Zhang, Bharti School of Engineering, Laurentian University, 935 Ramsey Lake Road, Sudbury, ON P3E 2C6, Canada; jzhang@laurentian.ca

algorithm exhibit a second-order accuracy in transport scalar value and a first-order accuracy in interface flux [7]. However, the mathematical formulation is relatively complicated even for the simplest D2Q5 (two-dimensional, five lattice velocities) lattice model (see Eqs. (19)–(21) and Table 1 in Ref. [7]). Such a complicated treatment could affect the computational efficiency, for example, in particulate or porous systems, where a large amount of solid-fluid interface nodes needs to be treated. Moreover, the general algorithm (Eq. (19) in Ref. [7]) cannot be applied at a location where the interface is approximately parallel or perpendicular to the underlying lattice grid lines, since in such situations the transverse intersection points ($\mathbf{x}'_{ss}$ and $\mathbf{x}'_{ff}$ there) may belong to a same domain. Special treatments (for example, switch to the decoupled scheme Eq. (26) in Ref. [7]) could be adopted to overpass this problem; however, this will make the interface method less efficient and less consistent (e.g., when simulating moving particles in particulate flows). Another concern is that only the gradient (flux) component along the intersecting link is available (Eq. (23) in Ref. [7]) but not that in the transverse direction. As a result, the normal mass or heat flux across the interface cannot be calculated directly, and this poses inconvenience in computational heat and mass transfer. For example, without the normal flux available, we even cannot calculate the local Nusselt number, which is an important parameter in heat transfer [15,16].

In this paper, we propose a counter-extrapolation approach for conjugate problems. Considering the physical principles at the interface, we are able to determine the transport scalar value at the interface by extrapolations from the two separate domains toward the interface along the normal direction. Once the scalar value at the interface is available, the conjugate interface has been transferred to a Dirichlet boundary to each domain, and appropriate Dirichlet boundary methods can be readily applied. Compared to the coupled method in Ref. [7], our method originates from a clear and straightforward mathematical layout and possesses a simple numerical algorithm. The above-mentioned technical difficulty also disappears, and the same interface algorithm can be applied for any interface-lattice relative orientation. The normal interfacial gradient is also directly available for calculations of the transport flux across the interface. Validation simulations are performed to examine the correctness and accuracy of our model by comparing LBM results to analytical solutions for flat and curved interfaces, and at last the cooling process of a hot cylinder in a cold flow is presented as a demonstration example for the potential usefulness of our model. Although in this paper the model description and simulations are presented using the D2Q5 lattice structure for simplicity, the interface treatment developed here can be readily applied to other lattice structures and even other numerical methods (such as the finite-difference method and the finite-volume method) of computational heat and mass transfer.

II. THEORY AND METHODS

A. Governing equations

Considering an interface Γ separating two domains Ω_1 and Ω_2 with different transport parameters. The spatial distribution and temporal evolution of a transport scalar property ϕ (ϕ can be taken as the temperature T for heat transfer or the concentration C for mass transfer) is governed by the following differential convection-diffusion equation:

$$\begin{aligned} \partial\phi/\partial t + \mathbf{u}\cdot\nabla\phi &= D_1\nabla^2\phi + G, \quad \text{in } \Omega_1, \\ \partial\phi/\partial t + \mathbf{u}\cdot\nabla\phi &= D_2\nabla^2\phi + G, \quad \text{in } \Omega_2. \end{aligned} \tag{1}$$

Here t is time, $\mathbf{u}$ is the velocity, G is the source term, and D_1 and D_2 are the thermal or mass diffusivities in domains Ω_1 and Ω_2, respectively. At the conjugate interface Γ, the transport scalar ϕ needs to be continuous (i.e., not jump across the interface) and the heat or mass flux has to be balanced according to the conservation law of energy and mass:

$$\phi_{1,\text{int}} = \phi_{2,\text{int}}, \tag{2}$$

$$D_1(\partial\phi/\partial n)_{1,\text{int}} = \sigma D_2(\partial\phi/\partial n)_{2,\text{int}}. \tag{3}$$

Here subscript "int" is used to indicate that the variable or its gradient is measured at the interface, and subscripts "1" and "2" denote on which side of the interface these properties are evaluated. $\partial/\partial n$ represents the derivative in the local normal direction $\mathbf{n}$, pointing from domain Ω_1 toward domain Ω_2. The parameter σ is constant 1 for mass transfer and is the volume heat capacity ratio, i.e., $\sigma = (\rho c_p)_2/(\rho c_p)_1$, with ρ the density and c_p the specific heat.

B. LBM model for transport phenomenon

To solve the convection-diffusion equation Eq. (1), we employ the multiple-relaxation-time (MRT) LBM model as in Ref. [7]. The distribution function $g_i(\mathbf{x},t)$ ($\mathbf{x}$ is the lattice node location and the subscript i denotes the lattice direction) evolves according to the following lattice Boltzmann equation [7]:

$$g_i(\mathbf{x}+\mathbf{c}_i\delta t, t+\delta t) - g_i(\mathbf{x},t) = \Lambda_i + \omega_i G(\mathbf{x},t)\delta t, \tag{4}$$

where δt is the time step and Λ_i is the collision operator. For the D2Q5 lattice structure [Fig. 1(a)] employed in this study, the lattice velocity $\mathbf{c}_i$ and lattice weight ω_i are given as: $\mathbf{c}_0 = [0,0]$, $\mathbf{c}_1 = [\delta x/\delta t, 0]$, $\mathbf{c}_2 = [-\delta x/\delta t, 0]$, $\mathbf{c}_3 = [0, \delta x/\delta t]$, $\mathbf{c}_4 = [0, -\delta x/\delta t]$, $\omega_0 = 1/3$, and $\omega_{1-4} = 1/6$. In MRT LBM models, the collision operator Λ_i is usually written in matrix expressions as:

$$\Lambda = [\Lambda_0, \Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4]^T = -\mathbf{M}^{-1}\mathbf{S}[\mathbf{m}(\mathbf{x},t) - \mathbf{m}^{\text{eq}}(\mathbf{x},t)], \tag{5}$$

where the transformation matrix $\mathbf{M}$ and the relaxation matrix $\mathbf{S}$ for the D2Q5 lattice structure are given as:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 4 & -1 & -1 & -1 & -1 \\ 0 & 1 & 1 & -1 & -1 \end{bmatrix}, \tag{6}$$

$$S = \begin{bmatrix} \tau_0 & 0 & 0 & 0 & 0 \\ 0 & \tau_1 & 0 & 0 & 0 \\ 0 & 0 & \tau_2 & 0 & 0 \\ 0 & 0 & 0 & \tau_3 & 0 \\ 0 & 0 & 0 & 0 & \tau_4 \end{bmatrix}^{-1}. \tag{7}$$

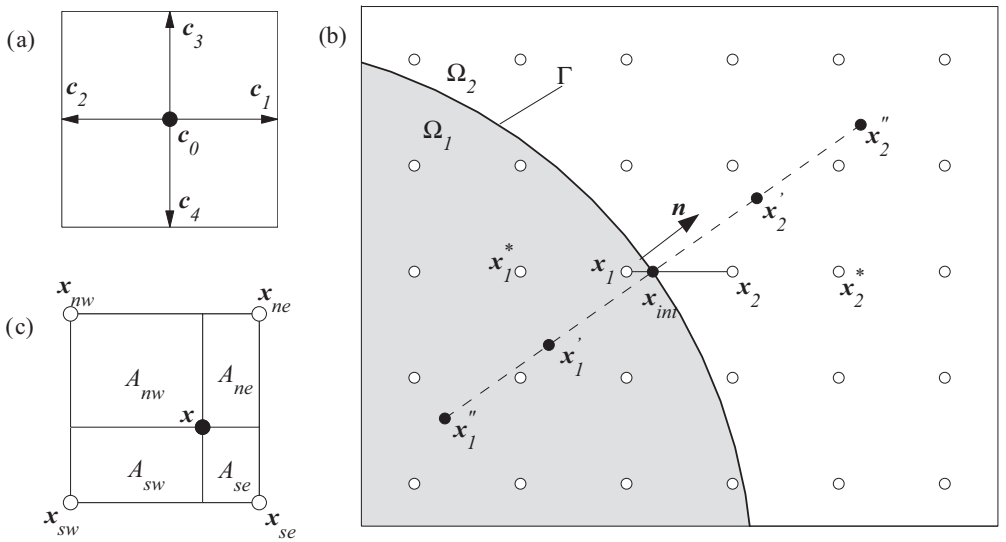

FIG. 1. Schematic illustrations for (a) the D2Q5 lattice structure, (b) the counter-extrapolation method for conjugate interfaces, and (c) the bilinear interpolation operation for estimating the property at an arbitrary position **x** from its four neighboring lattice nodes.

For the isotropic diffusion systems concerned in the present study, the elements τ_1 and τ_2 in matrix **S** are related to the diffusion coefficients D_1 and D_2 in Eq. (1) as:

$$\tau_1 = \tau_2 = \frac{1}{2} + \frac{3D_1\delta t}{(\delta x)^2}, \quad \text{in } \Omega_1, \tag{8}$$

$$\tau_1 = \tau_2 = \frac{1}{2} + \frac{3D_2\delta t}{(\delta x)^2}, \quad \text{in } \Omega_2. \tag{9}$$

The relaxation coefficient τ_0 does not affect the numerical solution, and the components τ_3 and τ_4 have no effect on the leading-order terms in the resulted convection-diffusion equation from the LBM algorithm. The off-diagonal components in the relaxation matrix **S** can also be set to nonzero values for anisotropic diffusion systems. In this study, we only consider isotropic situations and use $\tau_0 = \tau_3 = \tau_4 = 1$ [17].

The vector **m** in Eq. (5) is related to the distribution functions g_i via the transformation matrix **M**:

$$\mathbf{m} = [m_0, m_1, m_2, m_3, m_4]^T = \mathbf{M}[g_0, g_1, g_2, g_3, g_4]^T. \tag{10}$$

With the equilibrium distribution functions given as

$$g_i^{\text{eq}} = \omega_i \phi (1 + 3\mathbf{c}_i \cdot \mathbf{u}), \tag{11}$$

and the transport scalar

$$\phi = \sum_i g_i, \tag{12}$$

it is ready to obtain the following expression for the equilibrium counterpart for vector **m**:

$$\begin{aligned} \mathbf{m}^{\text{eq}} &= \left[m_0^{\text{eq}}, m_1^{\text{eq}}, m_2^{\text{eq}}, m_3^{\text{eq}}, m_4^{\text{eq}}\right]^T \\ &= \mathbf{M}\left[g_0^{\text{eq}}, g_1^{\text{eq}}, g_2^{\text{eq}}, g_3^{\text{eq}}, g_4^{\text{eq}}\right]^T \\ &= [T, uT, vT, 2T/3, 0]^T, \end{aligned} \tag{13}$$

where u and v are the components of velocity **u** in the x and y directions, respectively. The superscript T denote the transpose of a vector.

C. The counter-extrapolation method for conjugate interface

For our convenience, the lattice Boltzmann equation Eq. (A1) is split into two steps: the *collision* step:

$$\bar{g}_i(\mathbf{x}, t) = g_i(\mathbf{x}, t) + \Lambda_i + \omega_i G(\mathbf{x}, t)\delta t; \tag{14}$$

and the *propagation* step:

$$g_i(\mathbf{x} + \mathbf{c}_i\delta t, t + \delta t) = \bar{g}_i(\mathbf{x}, t). \tag{15}$$

The intermediate distribution function $\bar{g}_i$ is called the post-collision distribution function. We now consider a general curved interface Γ separating domains Ω_1 and Ω_2 as shown in Fig. 1(b). At a certain time step t, we have all the distribution functions g_i at each lattice node, and the postcollision distribution functions $\bar{g}_i$ can be calculated from Eq. (14). However, the propagation process cannot be performed as usual for lattice links crossing the interface, such as that connecting node $\mathbf{x}_1$ in domain Ω_1 and node $\mathbf{x}_2$ in domain Ω_2 [Fig. 1(b)]. We label the intersecting position as $\mathbf{x}_{\text{int}}$. Inspired by our recent work on Neumann and mixed Robin boundary conditions [18,19], we draw a line at point $\mathbf{x}_{\text{int}}$ perpendicular to the interface. On this line we find points $\mathbf{x}'_1$ and $\mathbf{x}''_1$ in domain Ω_1 and points $\mathbf{x}'_2$ and $\mathbf{x}''_2$ in domain Ω_2. The distance between two adjacent points along the perpendicular line is kept constant δ:

$$|\mathbf{x}''_1 - \mathbf{x}'_1| = |\mathbf{x}'_1 - \mathbf{x}_{\text{int}}| = |\mathbf{x}_{\text{int}} - \mathbf{x}'_2| = |\mathbf{x}'_2 - \mathbf{x}''_2| = \delta. \tag{16}$$

The ϕ values at points $\mathbf{x}''_1$, $\mathbf{x}'_1$, $\mathbf{x}'_2$, and $\mathbf{x}''_2$ can be obtained from neighboring lattice nodes in their individual domain via, for example, bilinear interpolations:

$$\phi(\mathbf{x}) \approx \frac{A_{\text{nw}}\phi(\mathbf{x}_{\text{se}}) + A_{\text{ne}}\phi(\mathbf{x}_{\text{sw}}) + A_{\text{sw}}\phi(\mathbf{x}_{\text{ne}}) + A_{\text{se}}\phi(\mathbf{x}_{\text{nw}})}{(\delta x)^2}, \tag{17}$$

where $\mathbf{x}_{\text{nw}}$, $\mathbf{x}_{\text{ne}}$, $\mathbf{x}_{\text{sw}}$, and $\mathbf{x}_{\text{se}}$ are the four nearest lattice nodes and A_{nw}, A_{ne}, A_{sw}, and A_{se} are the four fractional areas in the lattice square [Fig. 1(c)]. By assuming second-order polynomial variations along the perpendicular line in both

domains, the normal gradient on each domain side at the intersecting point $\mathbf{x}_{\text{int}}$ can be expressed as:

$$(\partial\phi/\partial n)_{1,\text{int}} \approx \frac{3\phi(\mathbf{x}_{\text{int}}) - 4\phi(\mathbf{x}_1') + \phi(\mathbf{x}_1'')}{2\delta}, \tag{18}$$

$$(\partial\phi/\partial n)_{2,\text{int}} \approx -\frac{3\phi(\mathbf{x}_{\text{int}}) - 4\phi(\mathbf{x}_2') + \phi(\mathbf{x}_2'')}{2\delta}. \tag{19}$$

The conjugate boundary condition Eq. (3) is then used in combination with Eqs. (18) and (19) to solve the interface value $\phi(\mathbf{x}_{\text{int}})$:

$$\begin{aligned}\phi_{\text{int}} &= \phi(\mathbf{x}_{\text{int}}) \\ &= \frac{D_1[4\phi(\mathbf{x}_1') - \phi(\mathbf{x}_1'')] + \sigma D_2[4\phi(\mathbf{x}_2') - \phi(\mathbf{x}_2'')]}{3(D_1 + \sigma D_2)}.\end{aligned} \tag{20}$$

Once the interface value ϕ_{int} is available, the original conjugate interface problem has been converted to a Dirichlet boundary problem for each domain, and an appropriate boundary method for the Dirichlet condition can be used. Here we adopt the midpoint bounce-back scheme for its simple algorithm and good numerical accuracy [18]:

$$g_{\bar{i}}(\mathbf{x}_1, t+\delta t) = -\bar{g}_i(\mathbf{x}_1, t) + 2\omega_i\phi_1^m, \tag{21}$$

$$g_i(\mathbf{x}_2, t+\delta t) = -\bar{g}_{\bar{i}}(\mathbf{x}_2, t) + 2\omega_{\bar{i}}\phi_2^m. \tag{22}$$

Here the lattice direction i is from node $\mathbf{x}_1$ to node $\mathbf{x}_2$ and the lattice direction $\bar{i}$ is the reverse direction. For the particular situation in Fig. 1(b), we have $i = 1$ and $\bar{i} = 2$. Parameters ϕ_1^m and ϕ_2^m are the estimated ϕ values at the midpoint $\mathbf{x}^m = (\mathbf{x}_1 + \mathbf{x}_2)/2$ via interpolation or extrapolation from each domain side, respectively [18,20]:

$$\phi_1^m = \frac{\phi_{\text{int}} + (1-2\Delta)\phi(\mathbf{x}_1)}{2(1-\Delta)}, \quad \Delta \leqslant 1/2, \tag{23}$$

$$\phi_1^m = \frac{3\phi_{\text{int}} - (2\Delta-1)\phi(\mathbf{x}_1^*)}{2(2-\Delta)}, \quad \Delta > 1/2, \tag{24}$$

$$\phi_2^m = \frac{\phi_{\text{int}} - (1-2\Delta)\phi(\mathbf{x}_2)}{2\Delta}, \quad \Delta \leqslant 1/2, \tag{25}$$

$$\phi_2^m = \frac{3\phi_{\text{int}} - (1-2\Delta)\phi(\mathbf{x}_2^*)}{2(1-\Delta)}, \quad \Delta > 1/2. \tag{26}$$

In these equations, $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ are the second lattice nodes along the intersecting lattice link from the interface into each domain, and Δ is the fraction in domain Ω_1 of the lattice link connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ [Fig. 1(b)]:

$$\Delta = \frac{|\mathbf{x}_{\text{int}} - \mathbf{x}_1|}{|\mathbf{x}_2 - \mathbf{x}_1|}. \tag{27}$$

The counter-extrapolation method described above performs extrapolations of ϕ along the local normal direction of the conjugate interface. This efficiently avoids the technical difficulty in Ref. [7] at locations where the interface is parallel or perpendicular to the underlying lattice grid lines. With a properly selected extrapolation interval δ ($\delta = 1.5\delta x$ in this study) [19], all the extrapolation control points $\mathbf{x}_1''$, $\mathbf{x}_1'$, $\mathbf{x}_2'$, and $\mathbf{x}_2''$ are well defined in their respective domains; unless the local curvature radius is very small (for example, a circular interface with a radius $R < 2\delta x$), and such a situation should be avoided anyway for the low spatial resolution. With the scalar value ϕ_{int} at the interface from Eq. (20), the normal gradients on both sides of the interface or the transport flux across the interface can be readily obtained via Eqs. (18) and (19).

III. VALIDATION AND DEMONSTRATION SIMULATIONS

In this section, we perform validation simulations to examine the correctness and accuracy of the counter-extrapolation method by comparing calculation results to analytical solutions. The systems considered are taken from Ref. [7] for the good coverage of steady and unsteady situations as well as flat and curved interfaces. We also simulate the cooling process of a hot cylinder in a cold flow as a demonstration example to illustrate potential applications of our conjugate interface treatment.

A. Steady and unsteady convection-diffusion processes with a flat interface

1. Model description and analytical solution

First we consider the system shown in Fig. 2 with a flat, horizontal interface. In this study, we limit ourselves to square systems with the interface in the middle, i.e., $L = H$ and $h = H/2$. The relative position of the interface to the underlying lattice mesh lines is represented by the interface-lattice offset Δ. The variable ϕ values at the top and bottom walls are specified by:

$$\phi(x, y=0) = \phi(x, y=H) = \cos(kx + \omega t), \tag{28}$$

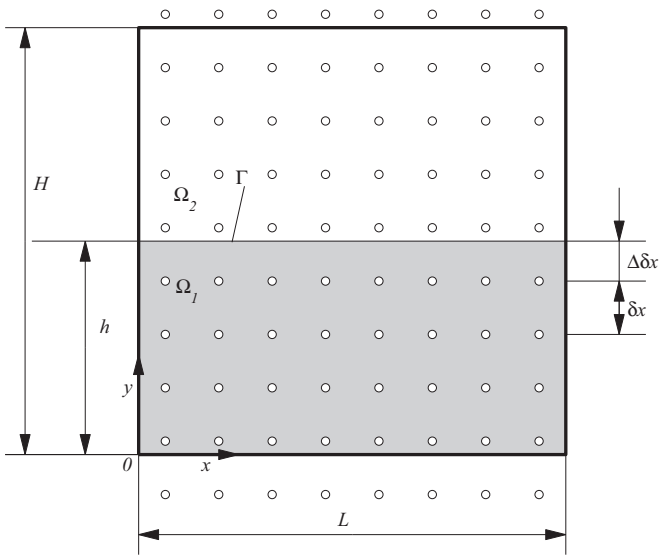

FIG. 2. The schematic for the flat interface system for steady and unsteady convection-diffusion simulations. The horizontal interface Γ at $y = h$ separates the two rectangular domains (Ω_1 for $0 < y < h$ and Ω_2 for $h < y < H$), and different transport coefficients can be assigned to them. The relative position of the interface to the underlying lattice grid is denoted by the distance from the interface to the first row of lattice nodes in domain Ω_1. In addition, a uniform velocity U_0 exists over the entire simulation domain, and the boundary condition at $y = 0$ and $y = H$ is given by Eq. (28). In this study we keep $L = H$ and $h = H/2$ as in Ref. [7].

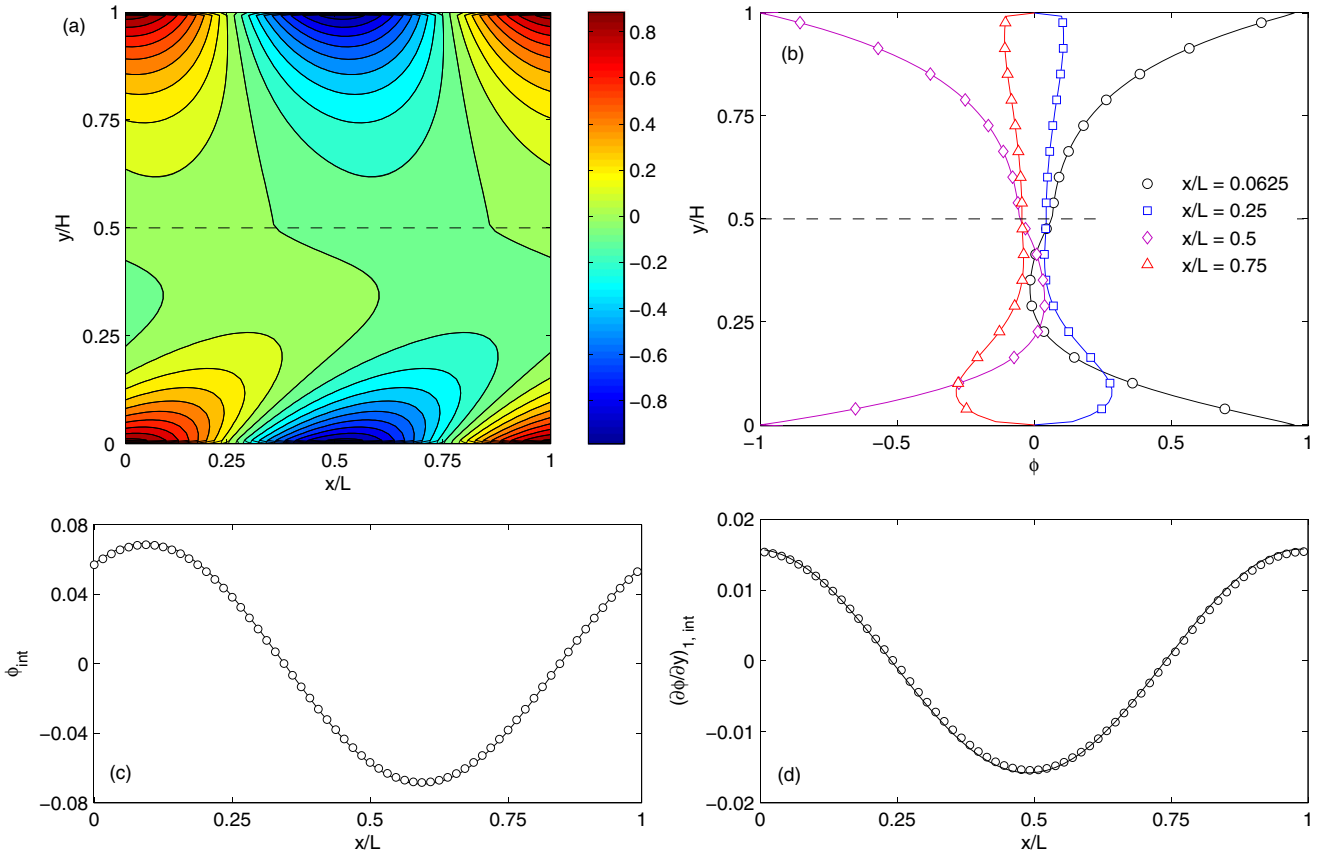

FIG. 3. (Color online) The ϕ distributions (a), the ϕ profiles along $x/L = 0.0625$, 0.25, 0.5, and 0.75 (b), the ϕ_{int} variation along the interface (c), and the $(\partial\phi/\partial y)_{1,\text{int}}$ variation along the interface (d) for the steady convection-diffusion system with a flat interface. In (a) the background color patches are from our LBM calculation and the black contour lines are from the analytical solution. In (b)–(d) the symbols are LBM results and the underlying curves are theoretical predictions. The dashed lines in (a) and (b) indicate the interface location.

where $k = 2\pi/L$ is the wave number and ω is the temporal frequency. The midpoint bounce-back boundary method [18] is utilized for this Dirichlet boundary problem for each domain, and periodic boundary conditions [4] are implemented at the left and right boundaries. In addition, a uniform flow $u(x,y) = (U_0,0)$ (U_0 is a constant) is applied over the entire computational domain. The analytical solution of this system is [7]

$$\phi_{\text{ex}}(x,y,t) = \text{Re}\{e^{i(kx+\omega t)}[\gamma_1 e^{-\lambda_1 y} + (1-\gamma_1)e^{\lambda_1 y}]\}, \quad 0 \leqslant y \leqslant h, \tag{29}$$

$$\phi_{\text{ex}}(x,y,t) = \text{Re}\{e^{i(kx+\omega t)}[\gamma_2 e^{-\lambda_2 y} + (1-\gamma_2 e^{-\lambda_2 H})e^{-\lambda_2(H-y)}]\}, \quad h \leqslant y \leqslant H,$$

with relevant parameters defined as below:

$$\gamma_1 = \frac{\lambda_1\left(a_3^2 - a_2^2\right) + \kappa\sigma\lambda_2\left(2a_1a_2a_3 - a_2^2 - a_3^2\right)}{(\lambda_1 + \kappa\sigma\lambda_2)\left(a_1^2a_3^2 - a_2^2\right) - (\lambda_1 - \kappa\sigma\lambda_2)\left(a_1^2a_2^2 - a_3^2\right)}, \tag{30}$$

$$\gamma_2 = \frac{\lambda_1\left(a_1^2a_3 + a_3 - 2a_1a_2\right) + \kappa\sigma\lambda_2\left(a_1^2 - 1\right)a_3}{(\lambda_1 + \kappa\sigma\lambda_2)\left(a_1^2a_3^2 - a_2^2\right) - (\lambda_1 - \kappa\sigma\lambda_2)\left(a_1^2a_2^2 - a_3^2\right)}, \tag{31}$$

$$a_1 = e^{-\lambda_1 h}, \quad a_2 = e^{-\lambda_2 h}, \quad a_3 = e^{-\lambda_2 H}, \tag{32}$$

$$\lambda_1 = k\sqrt{1 + i\frac{\omega + kU_0}{k^2 D_1}}, \quad \lambda_2 = k\sqrt{1 + i\frac{\omega + kU_0}{k^2 D_2}}, \tag{33}$$

$$\kappa = D_2/D_1. \tag{34}$$

2. Steady simulation

By setting $\omega = 0$, the temporal variational effect disappears, and the system becomes steady. Here, following Ref. [7], we use $D_1 = 1/60$, $D_2 = 1/6$, $\sigma = 1$, and the Peclet number $\text{Pe} = U_0 H/D_1 = 20$ for the steady simulations. Figure 3 compares the LBM results to those from Eq. (29) for the case with $H = 64$ and the interface-lattice offset $\Delta = 0.5\delta x$. In Fig. 3(a), we plot the iso-ϕ contours from our LBM calculation (the background color patches) and the analytical solution Eq. (29) (the black lines). Here we have a visually perfect match in the contours. Actually, if we plot the LBM and theoretical results both in contour lines, we will not be able to see any difference between them. For a more quantitative comparison, Fig. 3(b) displays the ϕ distribution profiles along four constant-x lattice lines, with the symbols from our LBM simulation and the curves from Eq. (29). Again excellent agreement is observed here. As the main purpose of this study

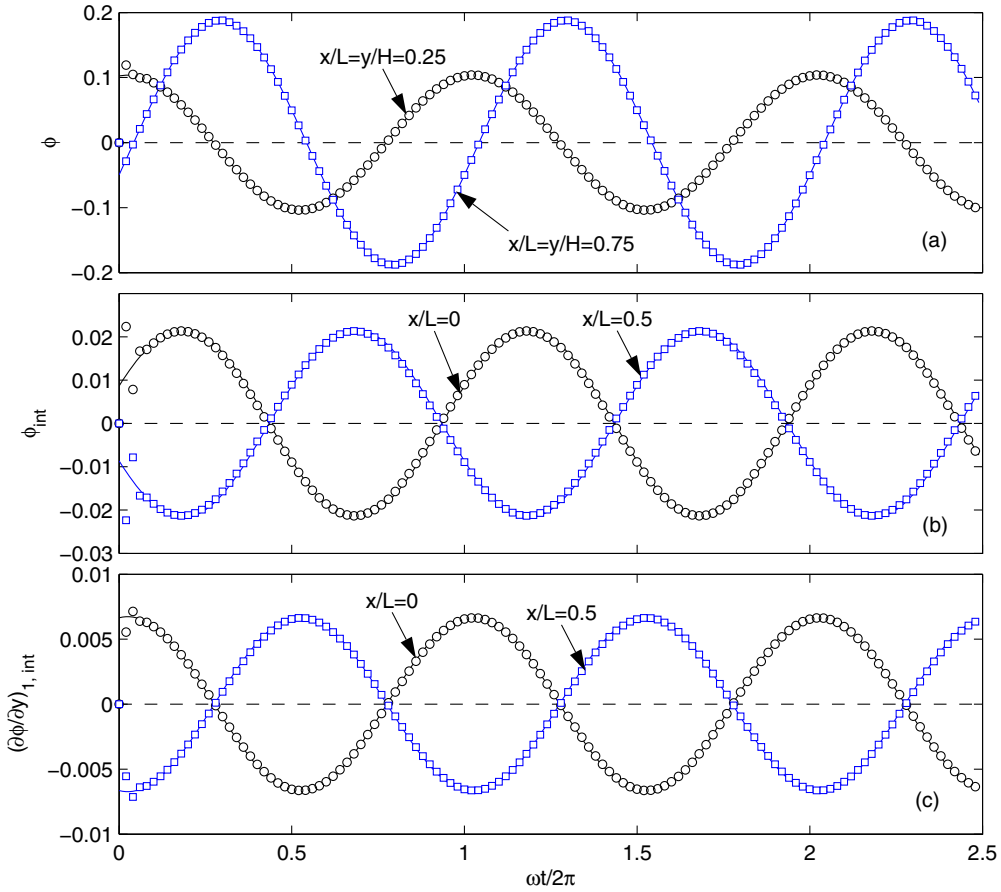

FIG. 4. (Color online) The variations of the ϕ values in separate domains (a) and the interfacial scalar (b) and gradient (c) values with simulation time for the unsteady convection-diffusion system with a flat interface. The symbols are LBM results and the underlying curves are from analytical solutions.

is the conjugate interface treatment, one may be interested in the performance of our method along the interface. It should be noticed that the value and flux continuity requirements in Eqs. (2) and (3) have been already considered in the derivation of Eq. (20) for ϕ_{int}. Thus, in Figs. 3(c) and 3(d), we only plot ϕ_{int} and $(\partial\phi/\partial y)_1$ along the interface, with symbols from our simulation [using Eqs. (20) and (18)] on top of the curves from Eq. (29). The agreement in ϕ_{int} is excellent in Fig. 3(c), while some small deviations can be noticed for the interfacial gradient in Fig. 3(d).

3. *Unsteady simulation*

Following Ref. [7], we next examine the performance of our counter-extrapolation method in dealing with unsteady problems. The system in Fig. 2 is employed again, however, with a dynamic boundary condition. The frequency ω in Eq. (28) is set from the Stokes number $\text{St} = \sqrt{\frac{H^2\omega}{2\pi D_1}} = 1$ [7]. The ratio σ is set to 10 to represent more general cases, for example, the heat transfer between two media with different volume heat capacities. All other parameters are kept the same as in the previous steady case. For simplicity, we start the simulation with a uniform $g_i(x,y,t=0) = 0$ initial condition. We monitor the ϕ values at two locations away from the interface and boundaries (one at $x/L = y/H = 0.25$ in domain 1 and another at $x/L = y/H = 0.75$ in domain 2) and the ϕ values and its gradients on two interfacial locations ($x/L = 0, y/H = 0.5$, and $x/L = y/H = 0.5$). Their revolutions with simulation time (normalized by the variation period $2\pi/\omega$) are displayed in Fig. 4. The figure shows that all recorded variables (symbols) quickly start to follow their individual analytical solutions (curves), and within 1/10 of the imposed variation period, the simulated results have already matched the theoretical predictions with no evident deviations. Also, as for the steady case, we compare the overall ϕ distributions [Fig. 5(a)], the ϕ profiles along constant-x lattice lines [Fig. 5(b)], and the ϕ values and its gradient variations along the interface [Figs. 5(c) and 5(d)] at time instant $t = 4\pi/\omega$ (i.e., two periods after the simulation starts). In general, good agreement between the LBM results and the exact solutions is observed.

4. *Accuracy analysis*

The accuracy of our method for flat conjugate interfaces is examined by varying the plate-plate distance H from $16\delta x$ to $128\delta x$ with keeping $L = H$ and $h = H/2$. To measure the numerical accuracy of our counter-extrapolation algorithm in the entire computation domain ($L \times H$) and along the interface, the following relative errors are

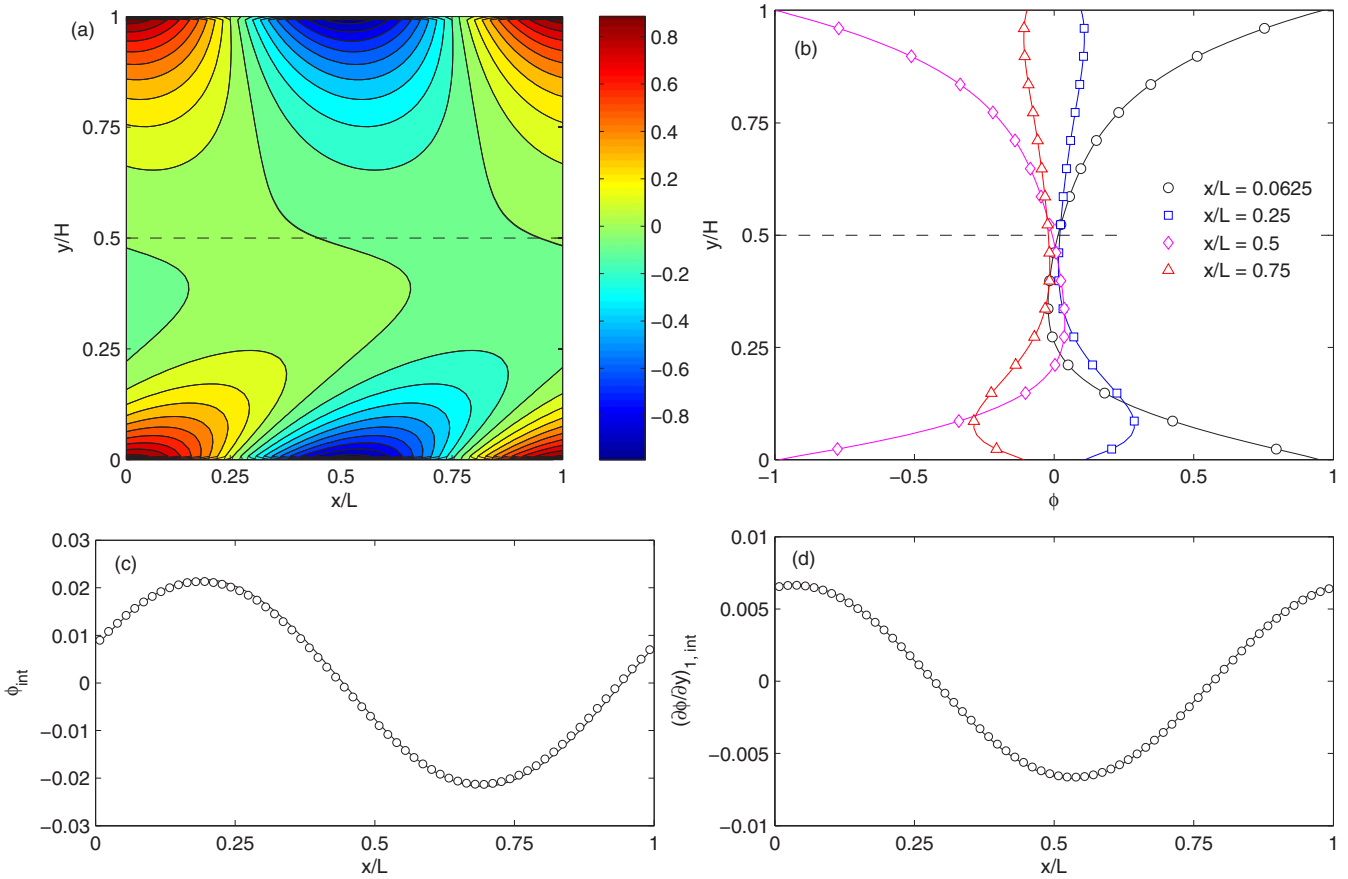

FIG. 5. (Color online) The ϕ distributions (a); the ϕ profiles along $x/L = 0.0625$, 0.25, 0.5, and 0.75 (b); the ϕ_{int} variation along the interface (c); and the $(\partial\phi/\partial y)_{1,\text{int}}$ variation along the interface (d) for the unsteady convection-diffusion system with a flat interface. In (a) the background color patches are from our LBM calculation and the black contour lines are from the analytical solution. In (b)–(d) the symbols are LBM results and the underlying curves are theoretical predictions. The dashed lines in (a) and (b) indicate the interface location.

calculated [7]:

$$E_2 = \frac{\sum_{\Omega_1+\Omega_2}(\phi_{\text{LBM}} - \phi_{\text{ex}})^2}{\sum_{\Omega_1+\Omega_2}\phi_{\text{ex}}^2}, \tag{35}$$

$$E_{2,\text{int}} = \frac{\sum_{\Gamma}(\phi_{\text{int,LBM}} - \phi_{\text{int,ex}})^2}{\sum_{\Gamma}\phi_{\text{int,ex}}^2}, \tag{36}$$

$$E_{2,\text{qint}} = \frac{\sum_{\Gamma}[(\partial\phi/\partial y)_{1,\text{int,LBM}} - (\partial\phi/\partial y)_{1,\text{int,ex}}]^2}{\sum_{\Gamma}[(\partial\phi/\partial y)_{1,\text{int,ex}}]^2}. \tag{37}$$

Here subscripts "LBM" and "ex" are used to denote the properties obtained either from LBM simulations or the exact analytical solution Eq. (29). For the unsteady simulations, the above defined errors also vary with time, and periodicity can be observed shortly after a simulation starts. The periodical variations of the errors have much shorter periods (compared to the imposed boundary value variation period $2\pi/\omega$) and small amplitudes. For these situations, we use the averaged error values over a complete error variation period for our following analysis.

In Fig. 6 we collect all the error data for E_2 (top panels), $E_{2,\text{int}}$ (middle panels), and $E_{2,\text{qint}}$ (low panels) from our steady (left panels) and unsteady (right panels) simulations and plot them versus the reciprocal of the domain height H with logarithm scales for both axes. For steady cases, different interface-lattice offset $\Delta = 0.01\delta x$ (black circles), $0.5\delta x$ (blue squares), and $0.75\delta x$ (red diamonds) are considered; and for unsteady situations, the effect of the ratio σ is examined using $\sigma = 0.1$ (black circles), 1 (blue squares), and 10 (red diamonds). Overall, the interface offset Δ and the ratio σ do not have significant influence on the accuracy performance. Based on the cases studied, the slopes in Figs. 6(c) and 6(c′) appear sensitive to the Δ and σ values, with a higher accuracy order from a lower Δ or a larger σ value. For all relative errors, the accuracy orders are in the range between 1 and 2. Similar accuracy orders have been reported in previous LBM simulations when finite-difference approximations are involved [18,19,21]; however, the exact origin of the decrease from the second-order accuracy of the LBM algorithm is not clear, and our recent study has indicated that the overall numerical accuracy order and magnitude could be affected by several factors, including the system geometry [18]. We have tested the nine-node Lagrange quadratic interpolation [22] in place of the bilinear interpolation scheme in Eq. (17), and not much of an improvement is observed in the relative errors and the results are presented. Also there is no significant difference noticed either in error magnitude or in accuracy order between the steady and unsteady cases.

The error magnitudes are larger and the accuracy orders are lower than those reported in Ref. [7] for the same systems. It

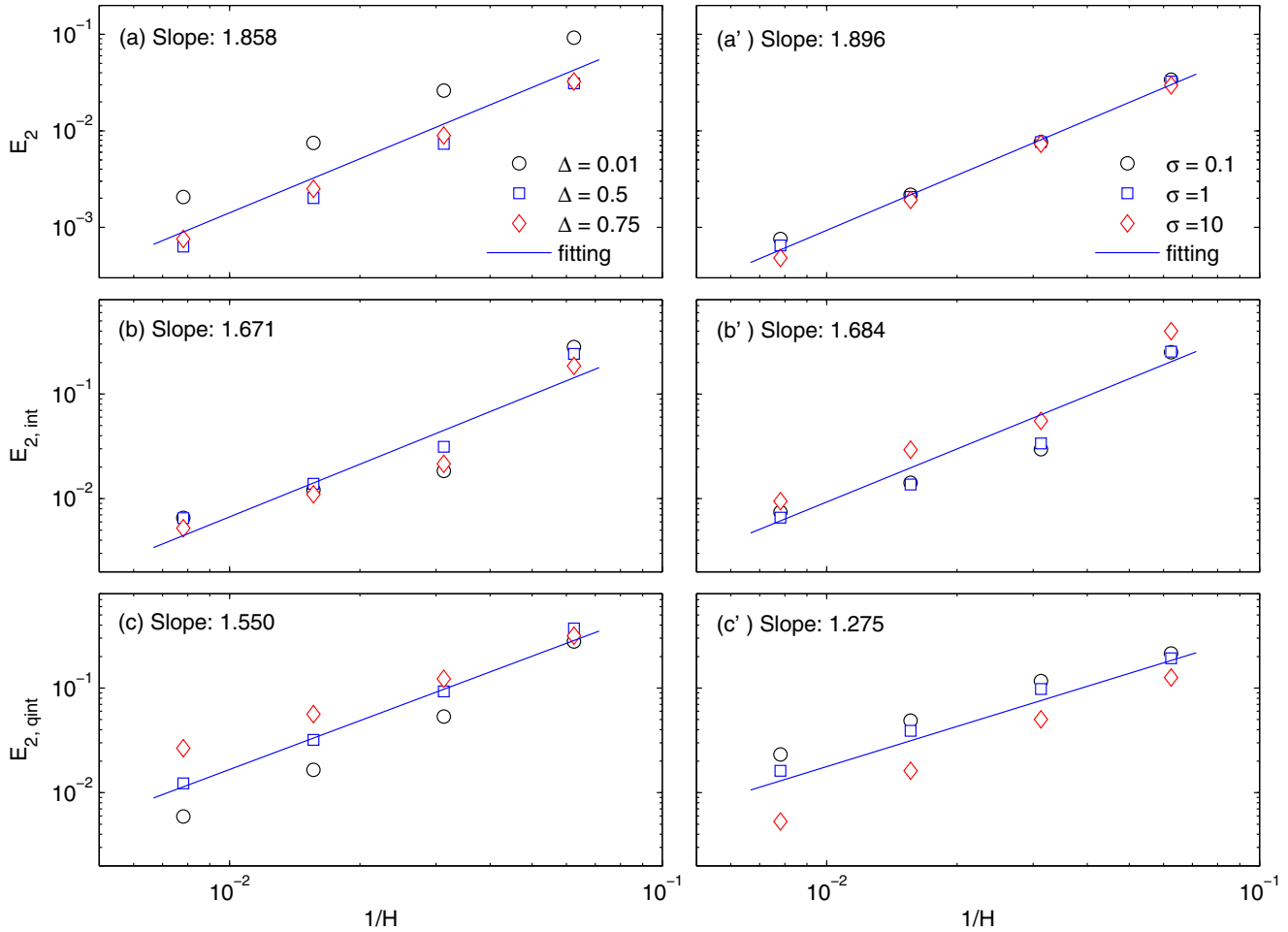

FIG. 6. (Color online) The relative errors E_2 [(a) and (a′)], $E_{2,\text{int}}$ [(b) and (b′)], and $E_{2,\text{qint}}$ [(c) and (c′)] for the steady [(a), (b), and (c)] and unsteady [(a′), (b′), and (c′)] convection-diffusion systems with a flat interface. For the steady systems, different interface-lattice offsets $\Delta = 0.01$, 0.5, and 0.75 are simulated with the volume heat capacity ratio $\sigma = 1$, while for the unsteady systems, different heat capacity ratios $\sigma = 0.1$, 1, and 10 are considered with the interface-lattice offset $\Delta = 0.5$. The straight lines are liner fittings of the LBM data points (symbols) in the log-log plots, and the line slopes are displayed in the figure labels.

should be pointed out that in Ref. [7] a different decoupled algorithm was developed for flat interfaces and utilized for these simulations by taking advantage of the particular interface shape and orientation. Also for the unsteady cases, special attention had been paid on the initial condition. These efforts might be helpful to improve the accuracy characteristics (reducing error magnitudes and increasing accuracy orders); however, these treatments are limited to these special situations (flat interfaces, aligned along lattice lines, and exact solutions available), and they cannot be applied to general systems. The better accuracy performance with these special treatments implemented may not reflect the real feature of the algorithm either. For example, in Ref. [7], $E_{2,\text{qint}}$ behaves similarly to E_2 and $E_{2,\text{int}}$ for flat interface systems, but it has much larger magnitudes and exhibits only a first-order accuracy for circular interface systems. This large difference could be from the different interface algorithms utilized for flat and interfaces there.

B. Steady diffusion system with curved interface

For a more general case, we next consider the diffusion system illustrated in Fig. 7, with domain Ω_1 in the central region of $r < R_1$ and domain Ω_2 in the surrounding ring region of $R_1 < r < R_2$. For this case, only the steady situation is simulated and the boundary ϕ value on the outer edge $r = R_2$ is specified by [7]:

$$\phi(r = R_2) = \cos(n\theta), \tag{38}$$

where θ is the polar angle and n is an integer. The analytical solution of this system is given by [7]:

$$\begin{aligned} &\phi(r,\theta) = b_1 r^n \cos(n\theta), \quad 0 \leqslant r \leqslant R_1, \\ &\phi(r,\theta) = (b_2 r^n + b_3 r^{-n}) \cos(n\theta), \quad R_1 \leqslant r \leqslant R_2, \end{aligned} \tag{39}$$

and the parameters b_1, b_2, and b_3 are given as:

$$\begin{aligned} b_1 &= \frac{2\sigma(D_2/D_1)R_1^{-2n}R_2^{-n}}{(\sigma D_2/D_1+1)R_1^{-2n} + (\sigma D_2/D_1-1)R_2^{-2n}}, \\ b_2 &= \frac{(\sigma D_2/D_1+1)R_1^{-2n}R_2^{-n}}{(\sigma D_2/D_1+1)R_1^{-2n} + (\sigma D_2/D_1-1)R_2^{-2n}}, \\ b_3 &= \frac{(\sigma D_2/D_1-1)R_2^{-n}}{(\sigma D_2/D_1+1)R_1^{-2n} + (\sigma D_2/D_1-1)R_2^{-2n}}. \end{aligned} \tag{40}$$

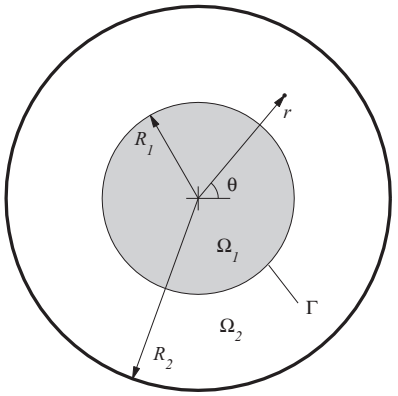

FIG. 7. The schematic of the steady diffusion system with a circular interface at $r = R_1$, which separates the two coaxial domains Ω_1 for $r < R_1$ and Ω_2 for $R_1 < r < R_2$. Different transport parameters are assigned to the two domains, and a steady but nonuniform boundary condition as given by Eq. (38) is imposed at the outer boundary at $r = R_2$.

In our simulations, we maintain n=4, R_2/R_1=2, D_2/D_1=10, and $\sigma = 1$. This set of parameters are taken exactly from Ref. [7] for the convenience of a direct comparison.

In Fig. 8, the LBM results for the case with $R_2 = 2R_1 = 32.5\delta x$ are compared to the exact solution in Eq. (39). Similarly to Figs. 3 and 5, here we display the ϕ distribution contours [Fig. 8(a); only a quarter of the system is shown for system symmetry], the ϕ radial profiles along $\theta = 0$ and $\theta = \pi/4$ [Fig. 8(b)], and the interfacial ϕ value and normal gradient variations along the interface $r = R_1$ [Figs. 8(c) and 8(d)]. Again there is no visual difference observed in the distribution contours, except at the center, where the $\phi = 0$ contour line does not reach the domain center radially but turns back near it. We believe this is from the plotting software, since the ϕ magnitude is very small near the center ($\phi = 0$ at $r = 0$) and the plotting program may not be able to *tell* such tiny difference in this region. Nevertheless, this is for the contour line from the exact solution, and our LBM calculation has captured the local ϕ distribution to a reasonable accuracy [see the color patches in Fig. 8(a)]. The overall agreement between LBM data and theory in Figs. 8(b)–8(d) is excellent. Slight deviations can be seen at the peaks and valleys of ϕ_{int} and $(\partial\phi/\partial r)_1$ in [Figs. 8(c) and 8(d)], and this is reasonable since larger errors may be introduced from the bilinear interpolations and normal extrapolations in such regions. Compared to Ref. [7], the gradient $(\partial\phi/\partial r)_{1,\text{int}}$ from our model matches the theoretical solution much better than the ϕ gradients in lattice directions in Ref. [7], especially at the peak and valley

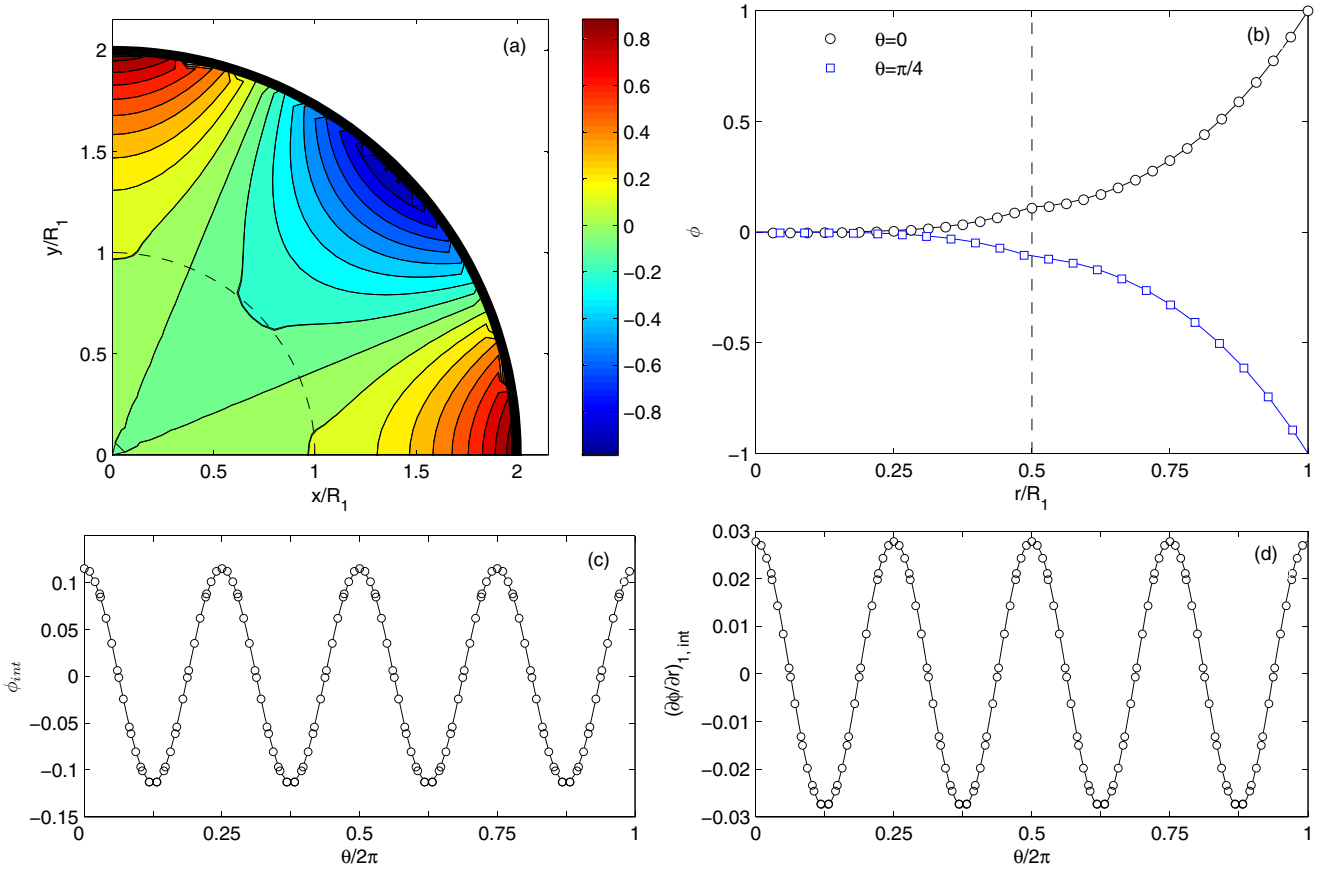

FIG. 8. (Color online) The ϕ distributions (a), the ϕ profiles along $\theta = 0$ and $\pi/4$ (b), the ϕ_{int} variation along the interface (c), and the $(\partial\phi/\partial y)_{1,\text{int}}$ variation along the interface (d) for the steady diffusion system with a circular interface. In (a) the background color patches are from our LBM calculation and the black contour lines are from the analytical solution. In (b)–(d) the symbols are LBM results and the underlying curves are theoretical predictions. The dashed arc in (a) and the dashed vertical line in (b) indicate the interface location.

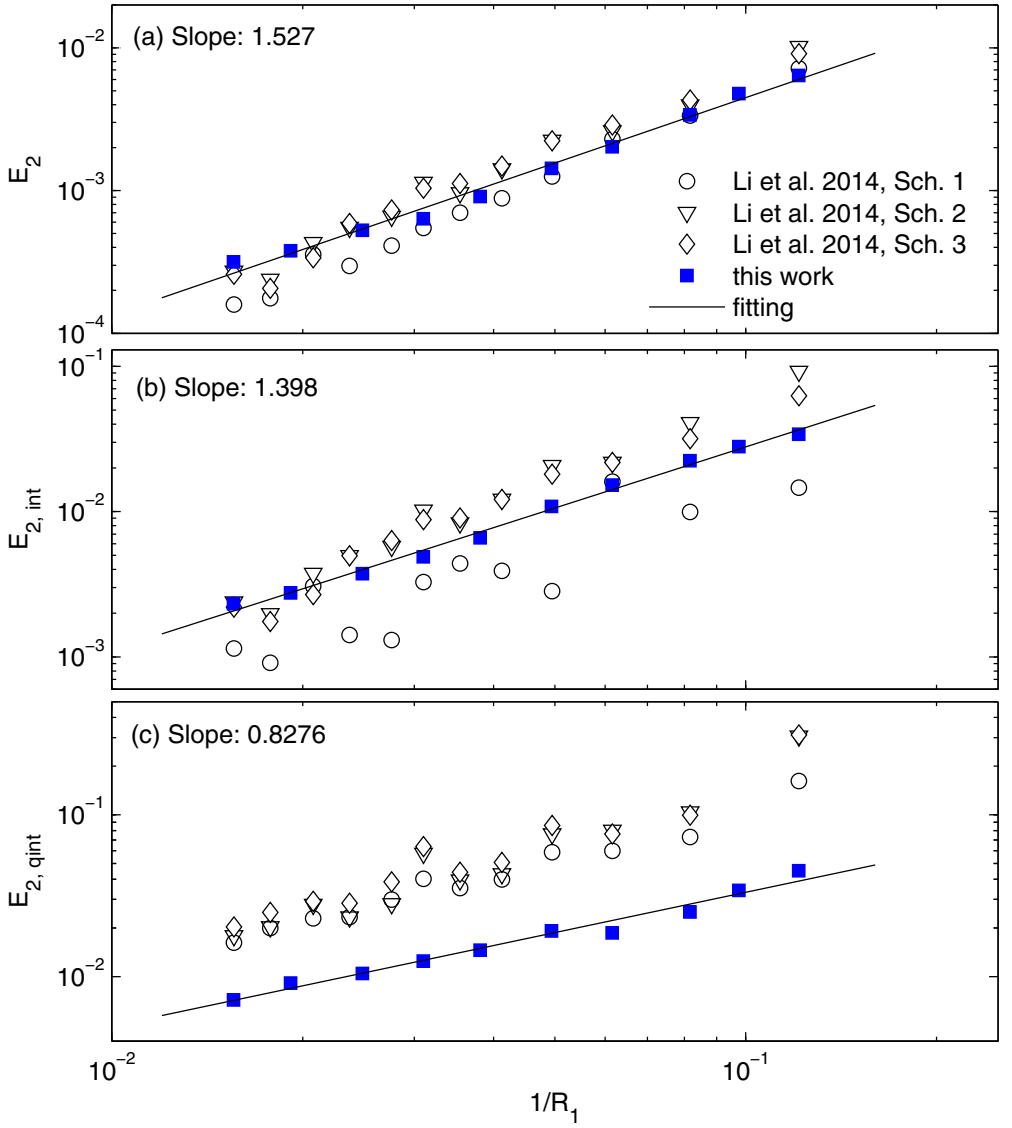

FIG. 9. (Color online) The relative errors E_2 (a), $E_{2,\text{int}}$ (b), and $E_{2,\text{qint}}$ (c) for the steady diffusion system with a circular interface. The straight lines are liner fittings of our LBM data points (filled squares) in the log-log plots, and the line slopes are displayed in the figure labels. The error data read from Ref. [7] are also displayed as open symbols for comparison.

locations [see Fig. 8(d) here and Fig. 18(b) there]. These large deviations there may be due to the technical difficulty we mentioned above, which occurs when the interface is parallel or perpendicular to the lattice directions. In addition, it should be mentioned that only the gradient component along the intersecting lattice link is available in Ref. [7] [Eq. (23) there] but not the transverse gradient component. A direct consequence of this incompleteness is that we cannot calculate the mass or heat flux across the interface and other important parameters such as the local Nusselt number directly without extra effort. For our counter-extrapolation method, the extrapolations are performed along the local normal direction, and the consistent algorithm produces relatively uniform agreement along the interface for both the interfacial value and normal gradient.

The accuracy performance of our model for this circular interface system is also investigated by changing the system size, and the relative errors calculated from Eqs. (35)–(37) are plotted in Fig. 9 as filled symbols in comparison to those reported in Ref. [7] as open symbols. One can see that the errors in ϕ value over the entire domain (E_2) and along the interface ($E_{2,\text{int}}$) from our model are of similar magnitudes as those in Ref. [7]. The respective converging orders, as indicated in the figure labels in Figs. 9(a) and 9(b), are 1.527 and 1.398, which are between those from the half-lattice division scheme [8] and the coupled model by Li *et al.* [7]. The relatively low accuracy orders might be from the finite-different approximations to obtain the ϕ values at the extrapolation control points, as reported in previous studies [18,19,21]. However, this should not be a serious concern for practical applications. As seen in Figs. 9(a) and 9(b), even at a high resolution of $R_1 = 64\delta x$, the E_2 and $E_{2,\text{int}}$ errors are still of similar magnitudes to those from the coupled method in Ref. [7]. If a finer resolution, for example, $R_1 = 128\delta x$ is employed, the errors estimated by following the converging trends are $E_2 \approx 10^{-4}$ and $E_{2,\text{int}} \approx 8 \times 10^{-4}$ for our method, and $E_2 \approx 5 \sim 8 \times 10^{-5}$

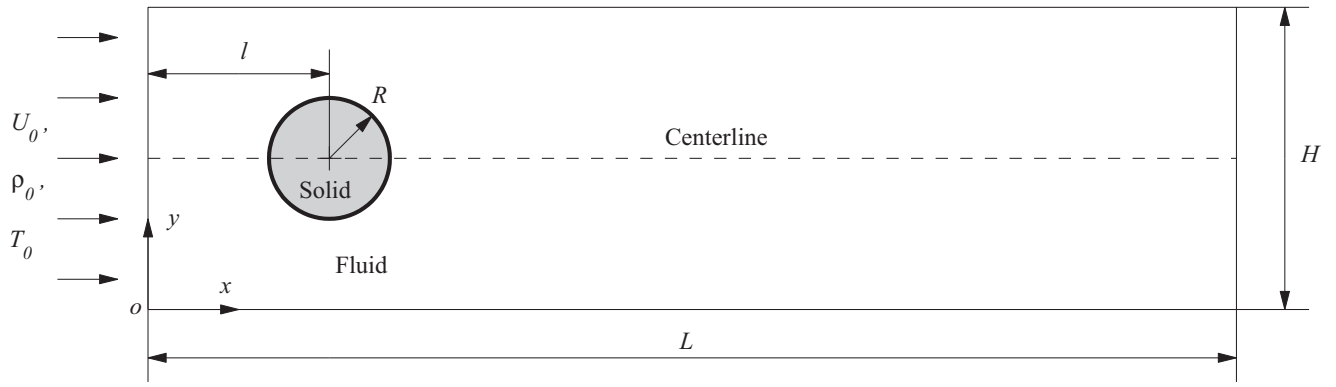

FIG. 10. The schematic for simulations of the cooling processes of a hot cylinder in a cold flow. Initially, the solid cylinder has a uniform temperature T_s, and the fluid is at rest with temperature $T_0 < T_s$. When the simulation starts, fluid enters at the left of the simulation domain with density ρ_0, velocity U_0, and temperature T_0, and the flow and temperature in both the fluid and solid domains are recorded. See text for detailed descriptions of the boundary conditions.

and $E_{2,\mathrm{int}} \approx 4 \sim 8 \times 10^{-4}$ for the method from Ref. [7]. The improvement in accuracy is relatively limited (of the order of $10^{-5} \sim 10^{-4}$), but the increase in computational cost could be significant (quadrupled in two-dimensional and eight-folded in three-dimensional systems). The better performance in interfacial gradient of our method observed before in Fig. 8(d) is reconfirmed in Fig. 9(c). The $E_{2,\mathrm{qint}}$ magnitudes are smaller and the accuracy order (slope) is similar to those from Ref. [7]. The better accuracy in interfacial gradient could be favorable for applications where the transport flux across the interface is important (for example, the calculation of the local Nusselt number [16]). In addition, with a uniform treatment for flat and curved interfaces, we observe a consistent error change trend in Figs. 6 and 9: larger error magnitudes and lower accuracy orders for interfacial errors $E_{2,\mathrm{int}}$ and $E_{2,\mathrm{qint}}$ than those for the global error E_2. This is reasonable since the interfacial treatment, although carefully developed, is still an *artificial disruption* to the regular LBM iteration, and relatively large errors are usually expected in such boundary or interface regions where the artificial disruption is injected [19,23].

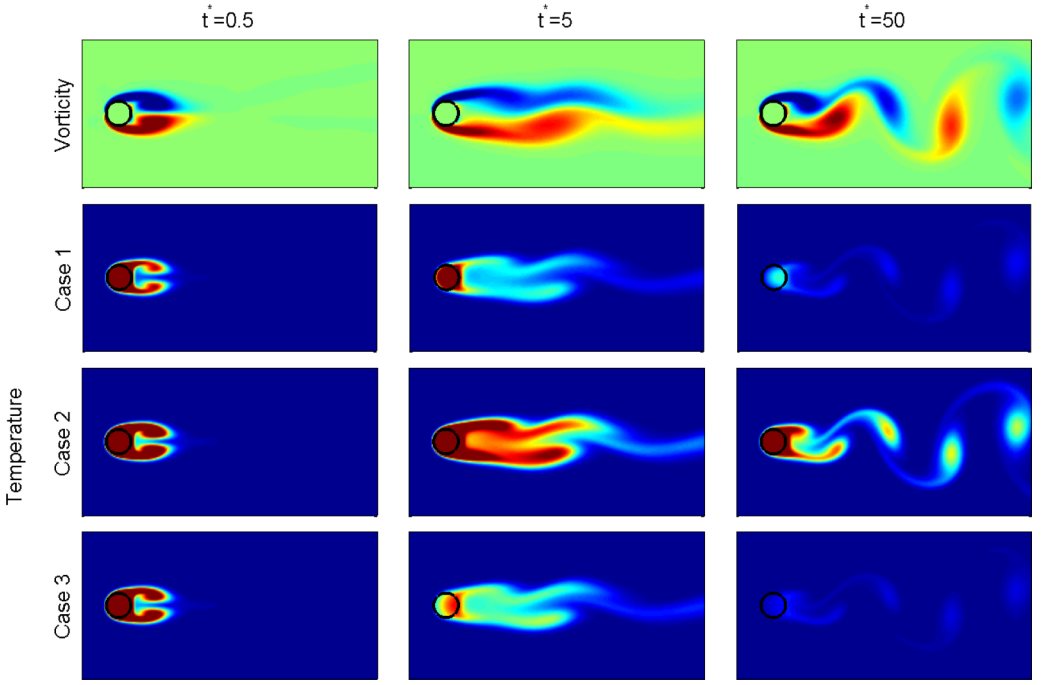

FIG. 11. (Color online) The distributions of flow vorticity (top panel) and temperature (lower three panels, one for each case simulated) at three representative time instants, $t^* = 0.5$ (left panels), 5 (middle panels), and 50 (right panels), during the cooling processes of the system in Fig. 10. All three simulations start from the same initial condition with the fluid at rest and the solid cylinder of a same higher temperature. The fluid and cylinder in Case 1 (second row) have the same heat capacity and heat conduction coefficient and this case serves as the control case for comparison. The solid cylinder in Case 2 (third row) has a larger heat capacity and thus the cylinder has more thermal energy to release. In Case 3 (last row) the solid cylinder has a better thermal conduction, which facilitates the heat conduction inside of the cylinder and therefore speeds up the overall cooling process.

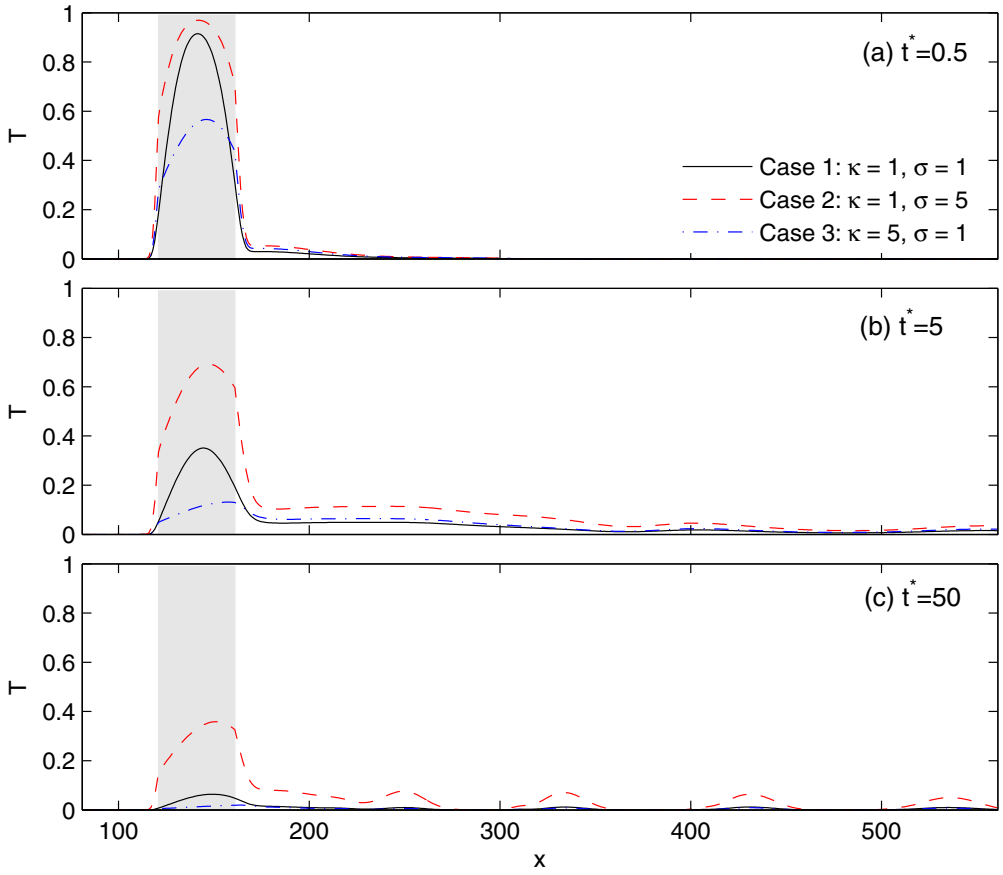

FIG. 12. (Color online) The temperature distributions along the system centerline for the three cases studied. Compared to Case 1 (black solid curves), the high temperature in the cylinder lasts longer and also the downstream fluid temperature is higher since there is more thermal energy in the cylinder in Case 2 (dashed red curves). On the other hand, the cooling process in Case 3 (blue dash-dotted curves) is faster due to the better thermal conduction of the cylinder. The gray area indicates the solid region of the cylinder.

C. Cooling process of a hot cylinder in a cold flow

At last, we simulate the cooling process of a hot cylinder with a cold fluid flowing over. The system involves unsteady flows and curved interface, and it can be considered an example to demonstrate potential usefulness of our model in conjugate heat transfer systems. The transport scalar ϕ is replaced with temperature T in the following discussion. Figure 10 provides a schematic description of the system setup, and we use subscripts f and s to distinguish properties of the fluid and the solid cylinder, respectively. At the left inlet ($x = 0$), we impose a uniform flow with constant velocity U_0, density ρ_0, and temperature T_0. For the top ($y = H$)and bottom ($y = 0$) boundaries and the right exit ($x = L$), we apply the following no-gradient conditions: $(\partial u/\partial y)_{y=0} = (\partial u/\partial y)_{y=H} = 0$, $(\partial \rho/\partial y)_{y=0} = (\partial \rho/\partial y)_{y=H} = 0$ (ρ is the fluid density), $(\partial T/\partial y)_{y=0} = (\partial T/\partial y)_{y=H} = 0$, $(\partial u/\partial x)_{x=L} = (\partial v/\partial x)_{x=L} = 0$, $(\partial \rho/\partial x)_{x=L} = 0$, and $(\partial T/\partial x)_{x=L} = 0$. Also the transverse velocity v is set to zero at $y = 0$ and $y = H$: $v(y = 0) = v(y = H) = 0$. The flow field is solved using the standard single-relaxation-time LBM model with the D2Q9 (two-dimensional, nine lattice velocities) lattice structure (see the appendix), and the no-slip boundary condition over the cylinder surface is achieved with the midpoint boundary method by Yin and Zhang [20]. The flow velocity from the flow calculation is then utilized in the calculation of temperature field, in particular, via Eq. (13). For simplicity, we assume that the fluid properties (density, viscosity, and heat capacity) are not changing with the fluid temperature, and the pressure compression work and viscous heat dissipation are neglected for the incompressible flow considered here [24]. Before starting the simulation, we set the fluid with a constant density $\rho(t = 0) = \rho_0$ and no flow velocity $\mathbf{u}(t = 0) = (0,0)$ and $T(t = 0) = T_s$ in the cylinder and $T(t = 0) = T_f$ in fluid. These boundary and initial conditions are typical in computational studies of flows over objects, and a more detailed description of the flow calculation can be found in the literature [16]. In our following simulations, we use $R = 20\delta x$, $L = 1600\delta x$, $H = 640\delta x$, $l = 300\delta x$, $\rho_0 = 1$, $U_0 = 0.05$, $T_0 = T_f = 0$, and $T_s = 1$. The fluid kinematic viscosity ν (and then the relaxation parameter in the D2Q9 LBM flow calculation, see the appendix) is obtained from the Reynolds number $\mathrm{Re} = 2U_0R/\nu = 100$, and, similarly, the fluid heat diffusivity D_f is calculated from the Prandlt number $\mathrm{Pr} = \nu/D_f = 2$. For the heat transfer properties of the cylinder, we consider the following three cases to examine their effects on the cooling process:

Case 1. $\kappa = D_s/D_f = 1$ and $\sigma = (\rho c_p)_s/(\rho c_p)_f = 1$ (control case);

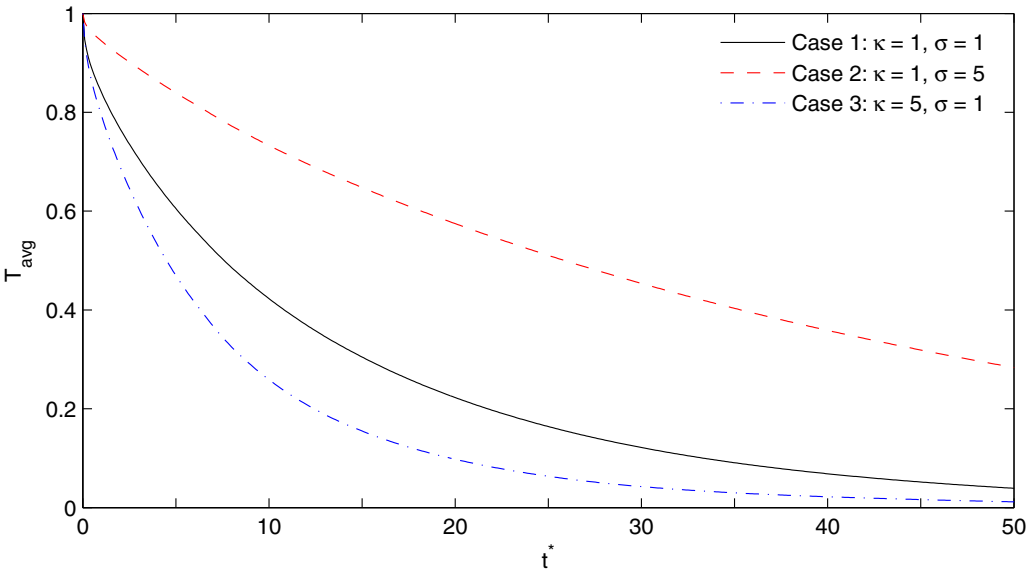

FIG. 13. (Color online) The changes of the average temperature of the cylinder during the cooling process for the three cases studied. Starting with a same value $T_0 = 1$, the cylinder temperature decreases slower in Case 2 (red dashed curve) due to the higher heat capacity and it drops faster in Case 3 (blue dash-dotted curve) due to the better thermal conduction of the solid.

Case 2. $\kappa = D_s/D_f = 1$ and $\sigma = (\rho c_p)_s/(\rho c_p)_f = 5$ (larger solid heat capacity);

Case 3. $\kappa = D_s/D_f = 5$ (better solid heat conduction) and $\sigma = (\rho c_p)_s/(\rho c_p)_f = 1$.

Figure 11 shows the distributions of flow vorticity (Row 1) and temperature (Row 2 for Case 1, Row 3 for Case 2, and Row 4 for Case 3) at normalized time $t^* = tU_0/2R = 0.5$ (Column 1), 5 (column 2), and 50 (Column 3). The flow field is the same for all three cases under our assumptions described above. At $t^* = 0.5$, the flow is still in the early development stage and an approximately symmetric vorticity pattern is observed. The symmetric structure is then quickly destroyed as time processes, and the unsteady nature of this moderate Reynolds number flow becomes dominant at $t^* = 5$. At $t^* = 50$, the well-known Karman vortex street pattern has been established behind the cylinder. These flow patterns are consistent to results reported in previous studies [16]. The temperature field is clearly associated with the flow structure for this convection-dominant system with the Peclet number $\mathrm{Pe} = \mathrm{RePr} = 200$.

The heat transfer parameters of the solid cylinder also play important roles. The cooling of the cylinder is much slower and the downstream fluid is much hotter in Case 2 compared to Case 1, since a larger solid heat capacity in Case 2 means more heat energy is stored in the hot cylinder. On the other hand, the better heat conduction in Case 3 facilitates the heat transfer inside of the cylinder and thus speeds up the heat release from the cylinder surface to fluid. For a more quantitative comparison, the temperature distributions along the system centerline $y = H/2$ are plotted in Fig. 12. Here we also notice that the hottest spot in the cylinder is shifting from the cylinder center toward the rear edge during the cooling process. This is understandable by considering the following fact: The front edge of the cylinder is directly exposed to the incoming cold stream, while the rear part surface is in contact with somewhat warmed fluid. As a result, the heat release flux is greater and the temperature drops faster in the front part than in the rear part. We also plot the average cylinder temperature over simulation time in Fig. 13, which clearly shows the effects of solid heat capacities and conduction coefficient on the overall cooling process in a same fluid flow.

IV. SUMMARY AND CONCLUDING REMARKS

We have developed a conjugate interface method for LBM simulations of convection-diffusion systems between two materials with different transport properties. The novelty of this method lies in that we look at the variation of the transport scalar along the normal direction of the local interface, and the scalar and flux continuity requirements at the conjugate interface can be directly utilized to determine the interfacial scalar value. With the interfacial scalar value obtained, the original conjugate problem can be solved as two boundary value systems with an appropriate method for the Dirichlet boundary conditions. Our counter-extrapolation method is advantageous over other existing LBM conjugate models in several aspects, including the simple and straightforward algorithm, accurate representation of the interface geometry, applicability to any interface-lattice relative orientation, and availability of the normal gradient. Validation simulations of steady and unsteady convection-diffusion systems with flat interfaces and steady diffusion systems with circular interfaces have been conducted, and the LBM results show good agreement with theoretical predictions. In addition, the cooling process of a hot cylinder in a cold flow, which involves unsteady convection and diffusion and a curved interface, has been studied to demonstrate the potential applications of our model in complex systems. Although all the model description and simulations in this paper are based on the MRT LBM model with the D2Q5 lattice structure, there should be no technical difficulty to extend this method to other LBM models or lattice structures. The basic idea of our counter-extrapolation

method can also be useful in other computational methods such as the finite-element or finite-volume methods for heat and mass transfer simulations.

We have not considered the spatial motion of the interface in this study for simplicity. When the interface moves in the domain, a lattice node near the interface may switch its status from an Ω_1 node to an Ω_2 node or vice versa. At such a node, the distribution functions for the flow and temperature (concentration) fields in the new domain do not exist, since this node has just entered into this domain. One possible technique is to approximate the missing distribution values at such nodes via extrapolations from existing neighboring nodes in this domain. This method has been used in our recent study for the diffusion-convection process of a cylinder moving in a straight channel and satisfactory results have been observed there [18]. The interface shapes in this study are either planar or circular, and therefore the local normal direction is readily available. For general interfaces with arbitrary shapes, extra efforts are necessary: For example, one can discretize the interface into small surface elements [line segments in two-dimensional (2D) systems and triangular elements in 3D systems] and use the normal direction of the surface element (which can be efficiently calculated from the element node coordinates) as an approximation of the local normal direction. Furthermore, for systems with large interface areas such as porous media, in particular in 3D simulations, the interfacial treatment could become demanding in computational resources and less efficient. Also, the extrapolation operation in this method also poses a requirement for the gap distance between two close interfaces, while such close gaps are frequently encountered in situations like particulate flows. Possible remedies include replacing the second-order extrapolation scheme in Eqs. (18)–(20) with their first-order counterparts, local lattice refinement, or adopting a simpler method with less spatial accuracy. These concerns are common for inter- or extrapolation-based boundary methods, and users should balance the calculation accuracy and efficiency according to their particular applications when choosing numerical models.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No.51303081), the Natural Science Foundation of Jiangsu Province (Grant No. BK20130761), the Natural Science and Engineering Research Council of Canada (NSERC), and the Laurentian University Research Fund (LURF). J.Z. acknowledges the helpful discussion with Dr. Like Li and Professor Renwei Mei at the University of Florida.

APPENDIX: THE D2Q9 SINGLE-RELAXATION-TIME LATTICE BOLTZMANN MODEL FOR FLUID FLOW

In LBM models for fluid flows, the density distribution $f_i(\mathbf{x},t)$ is governed by the following lattice Boltzmann equation:

$$f_i(\mathbf{x}+\mathbf{c}_i\delta t, t+\delta t) = f_i(\mathbf{x},t) + \Omega_i(f), \tag{A1}$$

which can be considered as a discrete version of the Boltzmann equation in classical statistical physics [25]. Here δt is the time step, and Ω_i is the collision operator incorporating the change in f_i due to the particle collisions. The collision operator can be further expressed as following with the single-relaxation-time (SRT) approximation [25],

$$\Omega_i(f) = -\frac{f_i(\mathbf{x},t) - f_i^{\mathrm{eq}}(\mathbf{x},t)}{\tau_f}, \tag{A2}$$

and here τ_f is a relaxation parameter. The equilibrium distribution f_i^{eq} is typically given as

$$f_i^{\mathrm{eq}} = \rho\alpha_i\left[1 + \frac{\mathbf{u}\cdot\mathbf{c}_i}{c_s^2} + \frac{1}{2}\left(\frac{\mathbf{u}\cdot\mathbf{c}_i}{c_s^2}\right)^2 - \frac{u^2}{2c_s^2}\right]. \tag{A3}$$

Here $\rho = \sum_i f_i$ is the fluid density and $\mathbf{u} = \sum_i f_i\mathbf{c}_i/\rho$ is the fluid velocity. Other parameters, including the lattice sound speed c_s and weight factors α_i, depend on the lattice structure employed. Through the Chapman-Enskog expansion [25], one can recover the macroscopic continuity and momentum (Navier-Stokes) equations from the above-defined LBM algorithm:

$$\begin{aligned}\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) &= 0,\\ \frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} &= -\frac{1}{\rho}\nabla P + \frac{\mu}{\rho}\nabla^2\mathbf{u},\end{aligned} \tag{A4}$$

where $\mu = (2\tau_f - 1)c_s^2\delta t/2\rho$ is the fluid viscosity and $P = c_s^2\rho$ is the fluid pressure. For the D2Q9 lattice model utilized in our current study, the nine lattice velocities are

$$\begin{aligned}\mathbf{c}_0 &= [0,0],\\ \mathbf{c}_i &= \left[\cos\frac{(i-1)\pi}{2}, \sin\frac{(i-1)\pi}{2}\right]\frac{\delta x}{\delta t}, \quad i = 1-4,\\ \mathbf{c}_i &= \left[\cos\frac{(2i-9)\pi}{4}, \sin\frac{(2i-9)\pi}{4}\right]\frac{\delta x}{\delta t}, \quad i = 5-8,\end{aligned} \tag{A5}$$

with δx as the lattice grid spacing. The lattice weight factors for the D2Q9 model are $\alpha_0 = 4/9$, $\alpha_{1-4} = 1/9$, and $\alpha_{5-8} = 1/36$; and the lattice sound speed is $c_s = (1/\sqrt{3})\delta x/\delta t$.

[1] A. Tarokh, A. A. Mohamad, and L. Jiang, Numer. Heat Tr. A-Appl. **63**, 159 (2013).
[2] W. D. Henshaw and K. K. Chand, J. Comput. Phys. **228**, 3708 (2009).
[3] A. A. Mohamad, in *Proceedings of the ASME International Mechanical Engineering Congress and Exposition 2007* (American Society of Mechanical Engineers, Seattle, Washington, 2008), pp. 1341–1346.
[4] J. Zhang, Microfluid. Nanofluid. **10**, 1 (2011).
[5] Z. Guo and C. Shu, *Lattice Boltzmann Method and Its Applications in Engineering* (World Scientific, Singapore, 2013).
[6] A. Dorfman and Z. Renner, Math. Probl. Eng. **2009**, 927350 (2009).
[7] L. Li, C. Chen, R. Mei, and J. F. Klausner, Phys. Rev. E **89**, 043308 (2014).
[8] J. Wang, M. Wang, and Z. Li, Int. J. Therm. Sci. **46**, 228 (2007).
[9] P. Meng, M. Wang, and Z. Li, Int. J. Heat Fluid Flow **29**, 1203 (2008).

[10] M. Seddiq, M. Maerefat, and M. Mirzaei, Int. J. Therm. Sci. **75**, 28 (2014).
[11] K. Zhao, Q. Li, and Y. Xuan, Sci. China Ser. E **52**, 2973 (2009).
[12] H. Yoshida, T. Kobayashi, H. Hayashi, T. Kinjo, H. Washizu, and K. Fukuzawa, Phys. Rev. E **90**, 013303 (2014).
[13] H. Karani and C. Huber, Phys. Rev. E **91**, 023304 (2015).
[14] L. Li, R. Mei, and J. F. Klausner, J. Comput. Phys. **237**, 366 (2013).
[15] Y. Q. Zu, Y. Y. Yan, W. P. Shi, and L. Q. Ren, Int. J. Numer. Methods Heat Fluid Flow **18**, 766 (2008).
[16] Q. Chen, X. Zhang, and J. Zhang, Commun. Comput. Phys., doi: 10.4208/cicp.2014.m314.
[17] H. Yoshida and M. Nagaoka, J. Comput. Phys. **229**, 7774 (2010).
[18] Q. Chen, X. Zhang, and J. Zhang, Phys. Rev. E **88**, 033304 (2013).
[19] O. Oulaid, Q. Chen, and J. Zhang, J. Phys. A: Math. Theor. **46**, 475501 (2013).
[20] X. Yin and J. Zhang, J. Comput. Phys. **231**, 4295 (2012).
[21] D. Hlushkou, D. Kandhai, and U. Tallarek, Int. J. Numer. Methods Fluids **46**, 507 (2004).
[22] J. N. Reddy, *An Introduction to the Finite Element Method* (McGraw-Hill, New York, 2004).
[23] W. Xiong and J. Zhang, Comput. Math. Appl. **61**, 3453 (2011).
[24] Y. Peng, C. Shu, and Y. T. Chew, Phys. Rev. E **68**, 026701 (2003).
[25] S. Succi, *The Lattice Boltzmann Equation* (Oxford University Press, Oxford, 2001).