Scientia Iranica B (2011) 18 (2), 231–240

Sharif University of Technology

Scientia Iranica

Transactions B: Mechanical Engineering

www.sciencedirect.com

Simulation of buoyant bubble motion in viscous flows employing lattice Boltzmann and level set methods

M. Mehravaran [a], S. Kazemzadeh Hannani [b,*]

[a] *Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 48824-1226, USA*

[b] *Center of Excellence in Energy Conversion, School of Mechanical Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-9567, Iran*

Received 12 June 2010; revised 23 November 2010; accepted 13 February 2011

KEYWORDS

Lattice Boltzmann method;
Level set method;
Two-phase flow;
Bubble dynamics.

Abstract Recently, a hybrid Lattice Boltzmann Level Set Method (LBLSM) for two-phase incompressible fluids with large density differences, in cases of negligible or a priori known pressure gradients, has been proposed. In the present work, the mentioned LBLSM method is extended to take into account pressure gradient effects. The lattice Boltzmann method is used for calculating velocities, the interface is captured by the level set function, and the surface tension is replaced by an equivalent body force. The method can be applied to simulate two-phase fluid flows with density ratios up to 1000 and viscosity ratios up to 100. In order to validate the method, the evolution and merging of rising bubbles were investigated, and the results are in agreement with other numerical or experimental results.

1. Introduction

The flow of two immiscible fluids separated by a sharp interface is encountered in various physical and industrial problems, such as petroleum refining, sprays, wave mechanics, bio-engineering, chemical reactors and combustion. Due to the practical importance of these types of problem, as well as academic interest, a large body of literature has been devoted to this subject over the years [1,2]. In cases of large density differences across the interface (such as the rising of a buoyant bubble), due to the significant role of forces between the components, the simulation of the problem becomes more sophisticated and challenging.

The dynamics of rising bubbles in viscous flow is crucial for many industrial applications, so it has become an important benchmark problem for immiscible multi-component flows in the area of fluid mechanics. Both large density changes and various deformations must be handled in this case, so many experimental and numerical studies have been performed to simulate this problem, such as the series of experiments conducted by Hartunian and Sears [3], Walters and Davidson [4,5], Grace [6], and Bhaga and Weber [7], or the numerical simulations of Hua and Lou [8] and Nagrath et al. [9].

Numerical methods used for modeling multiphase flows are divided into two general categories:

1. "front/interface tracking" methods, such as boundary integral [10] and Arbitrary Lagrangian–Eulerian (ALE) methods [11],
2. "front/interface capturing" methods, such as the Volume Of Fluid methods (VOF) [12], phase field methods and level set methods [13,14].

Each method has its own advantages and disadvantages, which are discussed in [10–14]. Because of the rapid topology changes in bubble dynamics phenomena, such as break down or coalescence, the level set method seems to be a good candidate for treatment of sharp geometrical changes. However, level set methods are weaker than VOF methods in conservation properties, and they need some kind of modification. The interested reader is referred to books by Sethian [13], and Osher and Fedkiw [15] for more information on the details of level set methods.

Sussman et al. [16] suggested several efficient techniques for the simulation of bubble dynamics as an incompressible two-component flow based on level set schemes. In these methods,

* Corresponding author.
E-mail address: hannani@sharif.edu (S.K. Hannani).

1026-3098 © 2011 Sharif University of Technology. Production and hosting by Elsevier B.V. Open access under CC BY-NC-ND license.
Peer review under responsibility of Sharif University of Technology.

doi:10.1016/j.scient.2011.03.018

Production and hosting by Elsevier

Nomenclature

f	mass distribution function
σ	surface tension coefficient
ξ	particle velocity vector
C_{drag}	drag coefficient
e_α	discrete particle velocity in LBM
$\tilde{V}$	volume of bubble
U	fluid velocity
Re	$\rho_l g^{1/2} D^{3/2}/\mu_l$, Reynolds number
p	fluid pressure
Bo	$\rho_l g D^2/\sigma$, Bond number
$f^{(\text{eq})}$	equilibrium distribution function
M	$g\mu_l^4/(\rho_l \sigma^3)$, Morton number
C_s	speed of sound
κ	curvature
λ	relaxation time
ν	kinematic viscosity
g	acceleration due to gravity
ϕ	level set function
$H_\varepsilon(\phi)$	smoothed Heaviside function
μ_l	viscosity of liquid

Abbreviations

BGK	Bhatnagar–Gross–Krook
LBGK	Lattice BGK
PDE	Partial Differential Equation
LBM	Lattice Boltzmann Method
D2Q9	2D 9-velocity

the velocity field is obtained by the solution of incompressible Navier–Stokes equations. This has its own difficulties as we have to encounter numerical instabilities, while simultaneously solving a system of second-order coupled nonlinear PDEs (see [17,18]).

The Lattice Boltzmann Method (LBM) has been developed as an alternative numerical scheme for the solution of incompressible Navier–Stokes equations in fluid mechanics. Although classically evolved from Lattice-Gas cellular Automata (LGA), the lattice Boltzmann equation may also be derived directly from the Boltzmann kinetic equation. Contrary to conventional Navier–Stokes solvers, LBM solvers circumvent treatment of the nonlinear convection term, because convection is substituted with a simple advection term. Besides, in LBM, data communication is always local and pressure may be calculated easily, making use of an appropriate equation of state, which is another advantage compared to the conventional finite volume/finite element Navier–Stokes solvers that need to solve the Poisson equation for pressure, and correct the magnitude of velocities by pressure. There are some other benefits in using LBM, such as the simple handling of boundary conditions and easy implementation. More information on LBM is available in the comprehensive paper of Yu et al. [19].

In addition to the methods outlined above, other hybrid methods, based on a combination of the LB equation with explicit interface-tracking/capturing schemes, have been development. Lallemand et al. [20] use the Peskin distribution function [21] for the interface force used in the Immersed-Boundary (IB) method. Mehravaran and Hannani [22] coupled the basic LBM and level-set into a simple scheme that can handle large property ratios properly, but has some limitations on the pressure field. Yu and Fan [23] also proposed a new scheme for high density ratios, via the species' potential, which however needed mesh refinement for capturing sharp geometries. Recently, Thommes et al. [24] obtained good results by using the level-set approach for tracking the interface, and the LB equation for the velocity field. In Thommes' method, the LBE is solved for each component and the coupling of the two components is handled via imposing boundary conditions on the interface at the kinetic level. Besides, when the interface moves, the population must be refilled, which adds to computational cost. However, advanced versions of LBM that are capable of simulating multiphase systems with high density ratios are available, but they usually need very fine mesh or have high computational cost [25–27].

In the current work, LBM was employed for solving velocities instead of Navier–Stokes equations. The Single-Relaxation-Time (SRT) model, known as the Bhatnagar–Gross–Krook (BGK) model version of LBM [28], is implemented. Consequently, the proposed hybrid algorithm makes use of the advantages of both LBM and level set methods. In the LBM framework, the pressure effects are also included and there is no need to solve for pressure in order to correct velocities as in conventional projection methods [17]. A solution for the pressure is required only if Dirichlet pressure boundary conditions exist (see Section 2.3.1 for more details on the process). This kind of convenience in handling pressure effects originates from local data communication in LBM [19,29], and from imposing pressure jumps as separate force terms near the interface. The governing equations are arranged in such a way that it is required to solve the basic single component LB equation for a virtual fluid, with virtual properties and certain force terms. Continuity or the jump condition on the interface is handled automatically, so no special treatment or boundary condition enforcement on the interface is needed (compared to Thommes et al. [24]).

From the numerical point of view, the method is preliminary. For example, it solves the LB equation in an easy explicit procedure, so there is no need to tackle troublesome iterative methods. However, LBLSM is robust enough to simulate typical benchmark problems in coarse or moderate grids with low computational cost, and it may be improved extensively by utilizing advanced numerical schemes. Compared to Thommes's method [24], the LBLSM needs neither very fine mesh for capturing complex geometries, nor a refilling step.

In this paper, first we present the governing equations in Section 2. Then, we will elaborate on the level set method and how it is used for the solution of two-component incompressible flows. In the sequel, the LBM will be explained. In Section 3, the new hybrid scheme for solving multiphase incompressible flows will be discussed in detail. Results and discussion will be included in Sections 4 and 5 consists of the conclusion and suggestions for future research.

2. Governing equations

We use the lattice Boltzmann equation, coupled with the level set equation, as a mathematical model for an incompressible two-phase flow problem. The velocity field is solved by employing LBM (BGK).

The velocities predicted by LBM are employed as input into the level set method, which can track the evolution of the interface, in time, in a known velocity field.

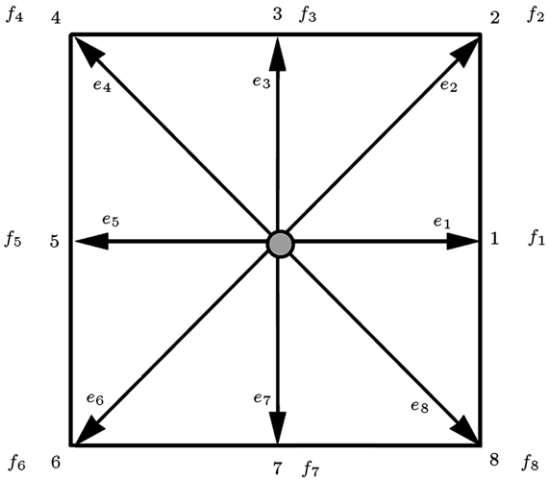

Figure 1: 2D 9-velocity model.

2.1. Lattice Boltzmann method

In LBM, the Boltzmann equation in which the collision operator is substituted by the well-known Bhatnagar–Gross–Krook (BGK) approximation [28] is as the following:

$$\frac{\partial f}{\partial t} + \xi.\nabla f = -\frac{1}{\lambda}\left(f - f^{(\mathrm{eq})}\right) + \frac{1}{N_\alpha e^2} e_{\alpha i} F_i(x,t), \tag{1}$$

where λ is relaxation time, and viscosity is $\upsilon = \lambda RT$ in which R is gas constant and T is temperature. In order to solve for f, Eq. (1) is first discretized in the velocity space using a finite set of velocity vectors, $\{\xi_\alpha\}$. The pattern used in the present work is the 2D 9-velocity (2D9Q) model [30]. N_α is a constant which is defined by the lattice pattern as:

$$N_\alpha = \frac{1}{e^2}\sum_\alpha e_{\alpha i} e_{\alpha i}. \tag{2}$$

In the 2D9Q model, N_α is equal to 6, and the finite set of velocity vectors consists of nine e_α vectors, which is shown in Figure 1. In this model, the speed of the sound is equal to $c_s = e/\sqrt{3}$, and the equation of state is the one used for ideal gas, $p = \rho c_s^2$.

The equilibrium distribution may be estimated by a polynomial of macroscopic properties, which is linear relative to fluid density:

$$f_\alpha^{(\mathrm{eq})} = \rho w_\alpha \left[1 + \frac{3}{c^2} e_\alpha.u + \frac{9}{2c^4}(e_\alpha.u)^2 - \frac{3}{2c^2}u.u\right], \tag{3}$$

where $c = \Delta x/\Delta r$ and w_α are the weighting factor (see [19] for more details).

By using 2D9Q corresponding weight factors, it will lead to obtaining the desired Navier–Stokes equations on the macroscopic level.

The macroscopic parameters, such as density and velocity, may be calculated using f [19,30].

The lattice Boltzmann equation (Eq. (4)) may be derived by the explicit discretization of the LBGK equation (Eq. (1)) in time and space:

$$f_\alpha(x_i + e_\alpha \Delta t, t + \Delta t) - f_\alpha(x_i, t) = -\frac{1}{\tau}\left[f_\alpha(x_i,t) - f_\alpha^{(\mathrm{eq})}(x_i,t)\right] + \frac{\Delta t}{6e^2} e_{\alpha i} F_{\alpha i}, \tag{4}$$

where $\tau = \lambda/\Delta t$. Using Chapmann–Enskog expansion [31], the following two equations can be recovered from the lattice Boltzmann equation:

$$\partial_t \rho + \nabla.(\rho \vec{u}) = 0, \tag{5}$$

and:

$$\partial_t(\rho u_j) + \partial_i\left(\frac{e^2}{3}\rho\delta_{ij} + \rho u_i u_j\right) - \tau\left(1 - \frac{1}{2\tau}\right)\frac{1}{3}\partial_i\left[\partial_j(\rho u_i) + \partial_i(\rho u_j)\right] = 0. \tag{6}$$

It is obvious that if we define pressure as $P = \rho e^2/3$, similar to the ideal gas equation of state, and the viscosity equal to $(1 - 2\tau)/6$, the incompressible Navier–Stokes equations can be derived from the above equations. Employing this modified definition for viscosity (compared to Eq. (1)), the truncation error in the discretization of Eq. (4) will be corrected (see [19] for details).

It may also be shown that in low Mach flows in which $|u|$ is much smaller than the speed of sound, c_s, the lattice Boltzmann equation will satisfy macroscopic continuity and momentum equations up to first-order of accuracy, and can be extended up to second-order using a modification for the force term [30]. The standard LBM cannot handle very large force terms, because strong forces will deviate from the Chapman–Enskog procedure about Maxwellian states. In cases of very large forces, the velocities will increase significantly and we will get far from the incompressible assumption of LBM.

As the basic LBM cannot tackle large forces properly, it is recommended to switch to modified versions of LBM [32] that can handle large forces to make use of the advantages of LBM compared to regular Navier–Stokes-based methods. As Navier–Stokes-based methods are more versatile and can handle large forces as well, they may be considered as a substitute in cases of large forces.

2.2. Level set method

In a simulation of bubble dynamics, we may face phenomena with large topological changes, i.e. coalescence or breaking-up, which cannot be simply implemented by front tracking methods. Considering that Level Set Methods (LSM) are capable of modeling complex geometries, as they capture the interface implicitly by means of a scalar function, they are used in the current simulation. The level set function, $\phi(x,t)$, is in general a smooth function. Interface Γ is the zero iso-surface of the level set function, ϕ, across which the sign of the level set function and the type of component we are dealing with, change, so we have:

$$\phi(x,t) = \begin{cases} > 0, & x \in \text{ liquid} \\ = 0, & x \in \Gamma \\ < 0, & x \in \text{ gas} \end{cases} \tag{7}$$

The properties of the fluid are functions of ϕ, i.e. density is defined as:

$$\rho(\phi) = \rho_1 H(\phi) + \rho_2(1 - H(\phi)), \tag{8}$$

and similarly:

$$\mu(\phi) = \mu_1 H(\phi) + \mu_2(1 - H(\phi)), \tag{9}$$

where ρ_1, ρ_2, μ_1 and μ_2 are densities of liquid and gas, and viscosities of liquid and gas, respectively, and $H(\phi)$ is the Heaviside function as:

$$H(\phi) = \begin{cases} 0, & \phi < 0 \\ \frac{1}{2}, & \phi = 0 \\ 1, & \phi > 0. \end{cases} \tag{10}$$

It is convenient to use the smoothed Heaviside function, $H_\varepsilon(\phi)$, instead of $H(\phi)$ as below:

$$H_\varepsilon(\phi) = \begin{cases} 0, & \phi < -\varepsilon \\ \frac{1}{2}\left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right)\right], & |\phi| \le \varepsilon \\ 1, & \phi > \varepsilon \end{cases} \tag{11}$$

and the resultant smoothed delta function, which assumes a thickness of $\frac{2\varepsilon}{|\nabla\phi|}$ for the interface is:

$$\delta_\varepsilon(\phi) = \frac{dH_\varepsilon(\phi)}{d\phi}, \tag{12}$$

that will lead to better numerical results (see [22] for details).

As long as the level set function is not far from a signed distance function, the interface will have a constant thickness, and the unit normal of the interface, from gas into liquid, and its curvature can easily be expressed in terms of $\phi(x, t)$ as below:

$$n = \left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi=0}, \qquad \kappa = \nabla.\left.\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right|_{\phi=0}. \tag{13}$$

The interface evolves in time according to the level set equation, which is similar to the convection equation, and is written as:

$$\frac{\partial\phi}{\partial t} + u.\nabla\phi = 0. \tag{14}$$

2.2.1. Re-initialization

Evolving in time, the level set function, ϕ, gradually distorts away from the signed distance function property, and we need to maintain it as a distance function by a process known as re-initialization or re-establishment. Various techniques have been proposed for the re-initialization process. According to Sussman et al. [14], re-initialization is implemented by solving a PDE up to steady state conditions in fictitious time, τ^d, as the following:

$$\frac{\partial d}{\partial \tau^d} + w.\nabla d = S(\phi), \tag{15}$$

where $d(x, 0) = \phi(x, t)$, $S(\phi)$ is the sign function and:

$$w = S(\phi)\frac{\nabla d}{|\nabla d|}. \tag{16}$$

It is clear that under steady state conditions, the time derivative term of Eq. (15) would be zero and $|\nabla d|$ will be equal to 1. For better numerical results, some form of smooth sign function may be used as:

$$S_\varepsilon(\phi) = 2\left(H_\varepsilon(\phi) - \frac{1}{2}\right). \tag{17}$$

Eq. (15) is a nonlinear hyperbolic equation and its characteristic velocities point outwards normal to the interface. Therefore, d is re-initialized by a sign distance function in the vicinity of the interface. Considering that the signed distance property is required just near the interface, we should solve Eq. (15) until gaining the mentioned property up to the desired distance from the interface, where $|d| \le \varepsilon$. This can be implemented by a finite number of iterations. For example, if the iteration step size is $\Delta\tau$, and the total interfacial thickness is 2ε, a maximum of $\varepsilon/\Delta\tau$ iterations will be required. As we are already close to the distance function, only two or three iterations would be satisfactory.

One of the main drawbacks of re-initialization in level set methods is the difficulty in maintaining the original position of the interface, which will cause spurious area/volume loss. Sussman et al. [16] proposed an improvement to the standard re-initialization process, which focused on preserving the amount of material in each cell, i.e. preserving the area (volume) in two(three) dimensions. We use the fact that:

$$\partial_\tau \int_{\Omega_{ijk}} H(d) = 0, \tag{18}$$

in every cell;

$$\Omega_{ij} = \left\{(x, y)|x_{i-1/2} < x < x_{i+1/2}, y_{j-1/2} < y < y_{j-1/2}\right\},$$

as the volume will not change due to the stationary interface. This volume constraint method was employed successfully by Sussman and Fatemi [16] in the same physical problem of multiphase flow, so we made use of that in the present work. There are some other efficient ways of implementing level set methods with better conservation properties, i.e. methods presented by Adalsteinsson and Sethian [33] which use velocity extensions constructed with the fast marching method that can drastically reduce or even eliminate the need for re-initialization in level set method applications, or the scheme proposed by Chopp [34], using bicubic interpolation for high order construction of re-initialization fields.

2.2.2. Surface tension modeling

Surface tension forces play an important role in many fluid mechanics problems. The surface tension force is a result of unbalanced forces exerted onto the molecules near the boundary by different fluids. In the present work, surface tension is replaced by a body force in the vicinity of the interface. This approach was presented by Brackbill et al. [35] and has also been implemented by Unverdi and Tryggvason [36] and Chang et al. [37] in a similar way. The magnitude of the force is proportional to the curvature of the interface, $\kappa(\phi)$, and is estimated by the following equation:

$$F_{s.\text{tension}} = -\sigma\kappa(\phi)\nabla H_\varepsilon(\phi) = -\sigma\kappa(\phi)\delta_\varepsilon(\phi)\nabla\phi. \tag{19}$$

As long as the level set function is a signed distance function, the curvature of the interface, $\kappa(\phi)$, can be found by solving ϕ from the following relation:

$$\kappa(\phi) = \nabla.\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{20}$$

The mentioned force exists just near the interface and vanishes far from it.

2.3. The new/proposed LBLS method

In the present implementation of the level set method, as mentioned above, the interface is captured implicitly by making use of a scalar parameter, so the method seems to be robust in handling complex topological and geometrical evolutions. The velocities of the domain must be solved and used as the input

of the level set method. Pressure effects are included, using the LBM approach as described below.

We shall assume that both fluids are governed by the incompressible Navier–Stokes equation, therefore:

$$\begin{aligned}
&\rho_1 u_{i,t} + \rho_1 u_j . u_{i,j} = -P_{,i} + \mu_1(\phi)(u_{i,j} + u_{j,i})_{,j} + \rho_1 g,\\
&\nabla . u = 0, \quad x \in \text{ liquid.}\\
&\rho_2 u_{i,t} + \rho_2 u_j . u_{i,j} = -P_{,i} + \mu_2(\phi)(u_{i,j} + u_{j,i})_{,j} + \rho_2 g,\\
&\nabla . u = 0, \quad x \in \text{ gas.}
\end{aligned} \tag{21}$$

The boundary condition at the interface, Γ, between the two phases is:

$$[P] = \sigma\kappa + 2[\mu](\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n}, \tag{22}$$

where $[*]$ represents a jump in the variable considered across the interface and $\vec{n}$ is the unit vector normal to the interface. Since the flow is viscous, velocities and their tangential derivatives are continuous across the interface, i.e.:

$$[u] = [v] = [w] = 0, \tag{23}$$

which leads to a jump condition:

$$[(\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n}] = 0. \tag{24}$$

According to Sussman and Fatemi [16], the governing equation for each fluid, along with the boundary condition at the interface, may be written as:

$$\begin{aligned}
&\rho(\phi)u_{i,t} + \rho(\phi)u_j . u_{i,j} + P_{,i} - (\mu(\phi)(u_{i,j} + u_{j,i}))_{,j}\\
&\quad = -\sigma\kappa(\phi)\delta_\varepsilon(\phi)\phi_{,i} + \rho(\phi)g, \qquad (25a)\\
&u_{i,i} = 0, \qquad (25b)
\end{aligned}$$

where ρ and μ are density and viscosity, respectively, and δ is the Dirac delta function. We must use Eq. (25a) in all nodes, so after each time step, the velocities will change but the densities will remain the same.

Considering Eq. (25a), we see that the left hand side is the single-phase Navier–Stokes equation and the right hand side can be considered as a body force term. So, Eq. (25a) can be replaced by the lattice Boltzmann equation with a force term. If we implement this algorithm, it is observed (not shown in the current paper) that after each time step, the densities of the two neighbor nodes, especially those about the interface, affect each other, and spurious vortices form near the interface. In order to solve this difficulty, the following procedure is used. Eq. (25a) may be rewritten as:

$$\begin{aligned}
&u_{i,t} + u_j . u_{i,j} + \frac{p_{,i}}{\rho(\phi)} - \frac{(\mu(\phi)(u_{i,j} + u_{j,i}))_{,j}}{\rho(\phi)}\\
&\quad = -\frac{\sigma\kappa(\phi)\delta_\varepsilon(\phi)\phi_{,i}}{\rho(\phi)} + g.
\end{aligned} \tag{26}$$

The above equation is similar to the Navier–Stokes equation for a fluid with the virtual density equal to unity and body force equal to the right hand side. The $\frac{\nabla p}{\rho(\phi)}$ term must be treated properly, which is explained in the following. However, this term disappears in cases of zero pressure gradients, or may be considered as a force field when the pressure gradient is known, i.e. collision of droplets in the atmosphere, Poiseuille flow and shallow water flow.

2.3.1. Handling pressure effects

When pressure is unknown, pressure effects may be included as follows: After differentiating the last term on the left hand side of Eq. (26), it may be rewritten as:

$$\bar{\rho}(\phi)u_{i,t} + \bar{\rho}(\phi)u_j . u_{i,j} + \frac{P_{,i}}{\rho(\phi)} - \bar{\mu}(\phi)(u_{i,j} + u_{j,i})_{,j} = \bar{F}, \tag{27}$$

and:

$$\bar{F} = \frac{\mu(\phi)_{,j}(u_{i,j} + u_{j,i})}{\rho(\phi)} - \frac{\sigma\kappa(\phi)\delta_\varepsilon(\phi)\phi_{,i}}{\rho(\phi)} + g, \tag{28}$$

where $\bar{\rho}(\phi)$ is virtual density approximately equal to unity; $\bar{\mu}(\phi)$ is virtual viscosity and is equal to $\mu(\phi)/\rho(\phi)$, and the right hand side can be considered as a force term called $\bar{F}$. The first term of $\bar{F}$ accounts for the effect of viscosity jump across the interface, while the second and third terms consider surface tension and gravity, respectively. Comparing Eq. (27) with Eq. (6), it is noticed that Eq. (27) may be solved in the frame of the lattice Boltzmann, but the term $\bar{\rho}_{,i}e^2/3$ must be considered equal to $P_{,i}/\rho(\phi)$ instead of $P_{,i}$, so pressure forces are included. Besides, as $[P]$ was excluded and considered as a force term, we may employ Eq. (27) as governing equations, and calculate pressure only by solving the following set of equations as [15]:

$$\nabla P = \rho(\phi)(\nabla\bar{\rho})e^2/3, \tag{29}$$

$$[P] = \sigma\kappa + 2[\mu]\,(\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n}, \quad x \in \Gamma, \tag{30}$$

or the following single smoothed equation instead:

$$\nabla P = \rho(\phi)(\nabla\bar{\rho})e^2/3 + \sigma\kappa(\phi)\delta_\varepsilon(\phi)\nabla\phi + \nabla.(2\mu(\phi)\tilde{D}), \tag{31}$$

where $\tilde{D}$ is the rate of deformation tensor and equal to $\tilde{D} = (u_{i,j} + u_{j,i})/2$. Eq. (31) may be rewritten as:

$$\begin{aligned}
\nabla P \;=\; &\rho(\phi)(\nabla\bar{\rho})e^2/3 + \sigma\kappa(\phi)\delta_\varepsilon(\phi)\nabla\phi\\
&+ 2[\mu]\delta_\varepsilon(\phi)\left((\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n}\right)\nabla\phi.
\end{aligned} \tag{32}$$

Actually, the pressure is smoothly modified and the pressure jump, due to surface tension and viscosity jump, is considered by the second and third terms on the right hand side of Eq. (32), respectively. All terms are functions of scalar ϕ that varies smoothly across the interface, and the pressure jumps are added in the vicinity of the interface via the $\delta_\varepsilon(\phi)$ function. In other words, Eq. (32) applies to both phases, and the interface and pressure jumps are automatically handled via the smeared delta function.

From the mathematical point of view, a corresponding pressure field exists if and only if:

$$\begin{aligned}
\nabla \times \nabla P \;=\; &\nabla \times (\rho(\phi)(\nabla\bar{\rho})e^2/3) = 0\\
&\Leftrightarrow e^2/3.(\nabla\rho(\phi)) \times (\nabla\bar{\rho}) = 0.
\end{aligned} \tag{33}$$

If we use the first choice for solving pressure, the equation is satisfied, because $\nabla\rho(\phi) = 0$ on each fluid, except for the interface where ∇P cannot be defined because of the jump condition. From the mathematical approach, the interface is a line of zero thickness, but we assume an infinitesimal thickness for it, in order to get better numerical results.

In cases of using the second choice for calculating pressure and a smoothed jump condition across the interface, it must be shown that Eq. (33) is satisfied as the interface thickness approaches zero. As $\varepsilon \to 0$, ∇P approaches:

$$\begin{aligned}
\nabla P \;\approx\; &\sigma\kappa(\phi)\delta_\varepsilon(\phi)\nabla\phi\\
&+ 2[\mu]\delta_\varepsilon(\phi)((\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n})\nabla\phi.
\end{aligned} \tag{34}$$

According to Eq. (24), there is no jump in the corresponding term across the interface, so $\nabla \times \nabla P$ will approach:

$$\begin{aligned}
\nabla \times \nabla P \approx\; &\nabla\sigma\kappa(\phi)\delta_\varepsilon(\phi) \times \nabla\phi\\
&+ \nabla(2[\mu]\delta_\varepsilon(\phi)((\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n})) \times \nabla\phi.
\end{aligned} \tag{35}$$

The terms $2[\mu]\delta_\varepsilon(\phi)((\nabla u.\vec{n}, \nabla v.\vec{n}, \nabla w.\vec{n}).\vec{n})$ and $\sigma\kappa(\phi)\delta_\varepsilon(\phi)$ are functions of ϕ, and function ϕ is a distance function near the interface, so the gradients of the two mentioned terms are vectors normal to the interface. Moreover, vector $\nabla\phi$ is similarly normal to the interface, so the left hand side of Eq. (35) will approach zero.

We must keep in mind that the solution of the above equations is just required when we need the value of pressure at a point, or when the problem is defined by Dirichlet pressure boundary conditions. In the current work, it was not needed to solve any of these equations for pressure. As we assume infinitesimal variations in the amount of $\bar{\rho}$, the method may encounter difficulties in modeling flows with high pressure gradients. In cases of such a problem, we may break the pressure as:

$$P = P_{\text{approximate}} + P_{\text{residual}}. \tag{36}$$

$P_{\text{approximate}}$ is the approximate amount of pressure that may be predicted, i.e. by the Bernoulli equation or potential flow, and its gradient could be imposed as a force term. We must also consider that large force terms also may violate the incompressibility hypothesis of the LBM. The small residual amount will be considered as P_{res}, which can be modeled by the lattice Boltzmann method. This method is useful in many cases, but is not a general way for modeling flows with large pressure gradients. In the current work, this approach is used for hydrostatic pressure and the pressure jump caused by surface tension forces. As the LBM method used employs an ideal-gas equation of state, it is weak in handling large pressure gradients. We may also use other modified lattice Boltzmann methods, which are able to calculate large pressure gradients [32], or solve Poisson's equation instead, but these techniques will be studied in future work.

Note that Eq. (27) is solved in the frame of LBM, so there is no need to care about the continuity equation, because mass conservation is satisfied up to first or second order, depending on the force term modification. In this context, the velocities are approximately divergence free.

3. Numerical method

In this section, numerical implementation of the governing equations will be discussed. The method proposed by Sussman et al. [16] for the re-initialization process will be explained concisely. At the end, the overall computational algorithm will be summarized.

3.1. Level set equation discretization

The first scheme that may be used for solution of the level set equation is the first-order upwind scheme, as:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + u^n\phi_x^n + v^n\phi_y^n = 0. \tag{37}$$

As this scheme is first-order accurate, it may be inappropriate for handling complex geometries, but it may be improved upon by using more precise approximations for ϕ_x^+ and ϕ_x^-. Velocity u defines whether ϕ_x^+ or ϕ_x^- should be used, and the spatial derivative approximations for ϕ_x^+ and ϕ_x^- are improved extensively. The Essentially Non Oscillatory, (ENO) method and the Weighted Essentially Non Oscillatory (WENO) method were used in the present work. In the ENO method, we use the smoothest possible polynomial interpolation to find ϕ, and then differentiate to obtain ϕ_x, but in the WENO method, the weighted convex approximation of three possible ENO approximations is used [15]. For more details, please refer to [33].

In Section 2.2, the importance of keeping level set function, ϕ, as a signed distance function from the interface was discussed, and several methods were proposed for gaining this objective. We use the PDE-based method, presented by Sussman and Fatemi, with the volume-constraint improvement (see [16,33] for details). In order to calculate the spatial derivatives, a second-order ENO method will be used.

The developed constraint (Eq. (28)) will significantly improve the accuracy of solving Eq. (15). This volume constraint improves results obtained by the ENO scheme, but it sometimes decreases the quality of results obtained with the significantly more accurate WENO scheme. So we must switch between the ENO and WENO schemes to get good results. It really depends on the problem, and it cannot be said that one method is generally better than the other. The algorithm of computation is summarized as below:

Step 1. Calculate the velocities using LBM;
Step 2. Update the level set function and density distribution;
Step 3. Re-initialize the level set function;
Step 4. Calculate force terms including surface tension force;
Step 5. Advance one time step and return to Step 1.

As mentioned before, a solution for pressure is required only if the Dirichlet pressure boundary conditions exists or we are interested in calculating the values of pressure at certain points, which may be implemented via solving Eq. (32) after Step 4.

4. Numerical results and discussion

In this part, several 2D problems regarding the motion of buoyant bubbles has been solved using a hybrid formulation of lattice Boltzmann and level set methods.

4.1. Single rising bubble

According to Hua and Lou [8], the dimensionless parameters that characterize the motion and deformation of a rising bubble are density ratio (ρ_l/ρ_b) and viscosity ratio (μ_l/μ_b), Reynolds number and Bond number, which are defined as:

$$\text{Re} = \frac{\rho_l g^{1/2} D^{3/2}}{\mu_l}, \qquad \text{Bo} = \frac{\rho_l g D^2}{\sigma},$$

where D is the effective diameter of the bubble, g is gravity and σ is surface tension coefficient. Sometimes, another dimensionless number may be used as:

$$\text{M} = \frac{g\mu_l^4}{\rho_l\sigma^3}.$$

The evolution of a rising bubble at various Re and Bo numbers will be simulated; the density and viscosity ratios are assigned to $\rho_l/\rho_b = 1000$ and $\mu_l/\mu_b = 100$, respectively. We impose zero gradient boundary conditions on all walls.

4.1.1. Single buoyant bubble at Re = 20 and Bo = 1.2

In this section, the motion of a single bubble at Re $= 20$, Bo $= 1.2$ is simulated. The transient problem is solved until a steady solution is achieved. The mesh used is 50×175 and $\Delta x = \Delta y = \Delta t = 0.001$. As the Bo number is small, the surface

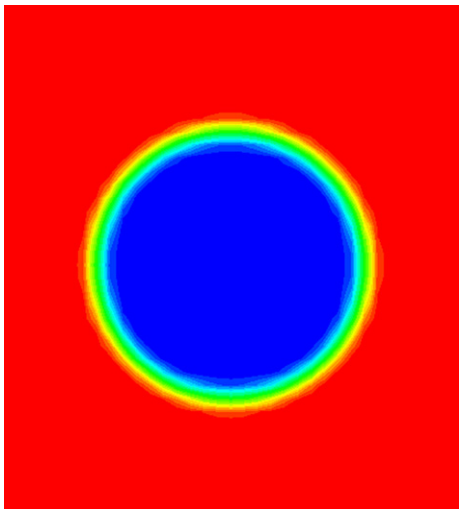

Figure 2: Single rising bubble at Re = 20 and Bo = 1.2.

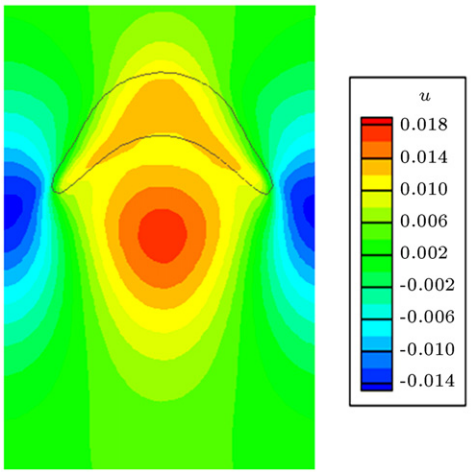

Figure 4: Upward velocity contours of a buoyant bubble; Re = 62.36 and Bo = 116 at steady conditions.

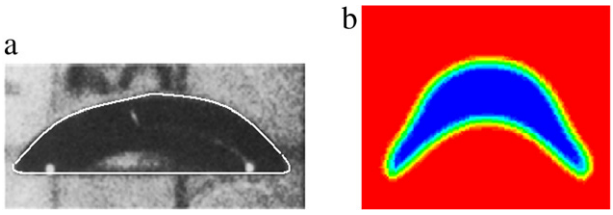

Figure 3: Shape of a rising buoyant bubble and a comparison between (a) experimental photographs [7] (the interface is shown by a line), and (b) the results of numerical simulation for Re = 62.36 and Bo = 116.

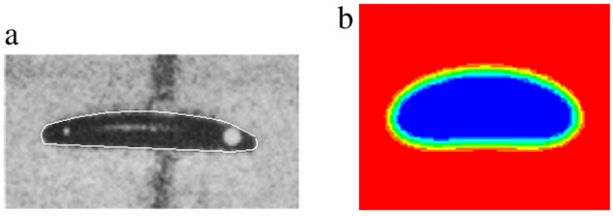

Figure 5: Shape of a rising buoyant bubble and a comparison between (a) experimental photographs [7], and (b) the results of numerical simulation for Re = 79.88 and Bo = 32.2.

tension force is strong enough to prevent great deformation of the bubble from a circular or oval shape. The maximum change in mass (or area) was about 0.066 in our simulation. The numerical result is shown in Figure 2, and is in agreement with numerical and experimental results [16,38]. In some cases of small Bo and Re numbers, the numerical simulation diverges, but the method is robust in modeling large Bo and Re numbers.

4.1.2. Single buoyant bubble at Re = 62.36 *and* Bo = 116

In the second problem Re and Bo numbers are raised to 62.36 and 116, respectively. Considering that the density and viscosity ratios are approximately equal to those of air and water, our numerical simulation (Figure 3) compares well with experimental data [7] and numerical results [8]. The vertical velocity contours are also shown in Figure 4. In our numerical simulation, maximum error in the mass (or area) of the bubble was about 0.095. Average error was less than half this amount, which is due to the fact that we have solved the problem in the framework of the lattice Boltzmann method that satisfies mass and momentum conservation laws simultaneously.

As explained by Nagrath et al. [9], the pressure gradient at the lower surface is greater than the upper one, so a jet of water forms that will push the bubble from below and cause it to curve up. As this phenomenon is captured in our simulation, our method can handle pressure forces as well.

4.1.3. Single buoyant bubble at Re = 79.88 *and* Bo = 32.2

We have simulated the problem for Re = 79.88 and Bo = 32.2 (Figure 5); the vertical velocity contours are also shown in Figure 6. The deviation in mass of the bubble was approximately equal to 0.095 and average error was about half this amount.

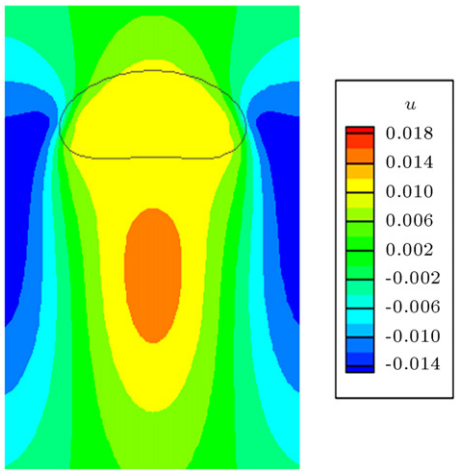

Figure 6: Contours of upward velocity for a buoyant bubble at Re = 79.88 and Bo = 32.2 under steady conditions.

4.1.4. Single buoyant bubble at Re = 100 *and* Bo = 200

Finally, we have solved the same problem for Re = 100 and Re = 200. In our simulation, we captured two vortices behind the bubble, due to different pressure gradients on the upper and lower sides of the bubble [9], which causes pinching off followed by shredding of the satellite bubbles (Figures 7 and 8). The same phenomenon was reported by Walter and Davidson [4] and Nagrath et al. [9]. The final shape of the bubble is also in agreement with the results of Hua et al. [8]. The

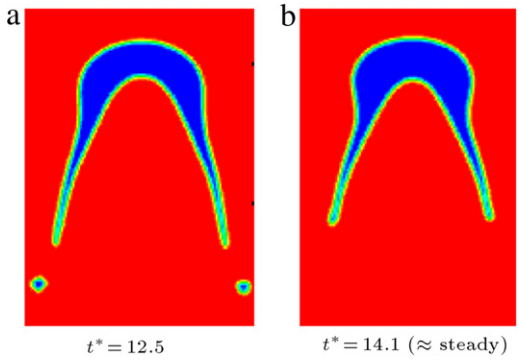

Figure 7: Evolution of a rising buoyant bubble employing LBLSM for Re = 100 and Bo = 200.

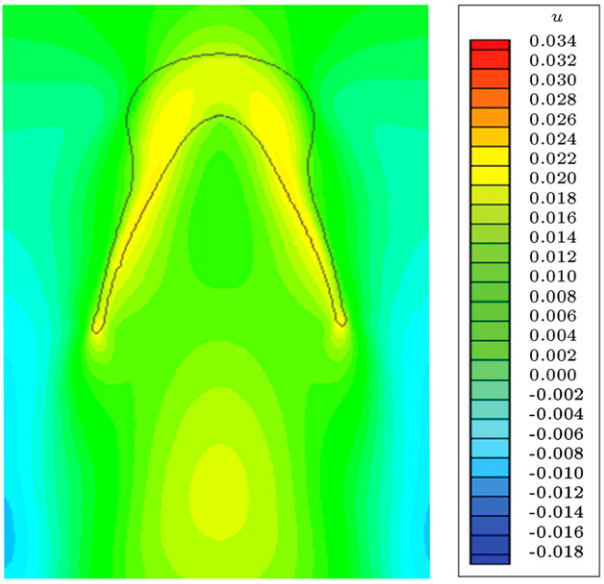

Figure 8: Upward velocity contours of a buoyant bubble; Re = 100 and Bo = 200 at steady conditions.

deviation in error was equal to 0.095. The non-dimensional time is defined as $t^* = t = \sqrt{\frac{g}{D}}$.

In spite of the fact that our code is based on a 2D/Cartesian grid, we see that there is good agreement with the experimental results, except for the terminal velocity of bubbles that had an error of about 50% in some cases, which is unacceptable. This deviation is due to the fact that we have used a 2D/Cartesian solver. The terminal velocity of the disk shaped bubble may be calculated, considering the balance of buoyancy and drag force as below:

$$F_{\text{buoyancy}} = (\rho_l - \rho_b)\tilde{V}g = F_{\text{drag}} = \frac{1}{2}C_{\text{drag}}\rho_l U^2 A_{\text{proj}}, \quad (38)$$

where $\tilde{V}$ is the volume of the bubble and consequently;

$$U = \sqrt{\frac{2(\rho_l - \rho_b)\tilde{V}g}{C_{\text{drag}}\rho_l A_{\text{proj}}}} \propto \sqrt{\frac{\tilde{V}}{C_{\text{drag}}A_{\text{proj}}}}. \quad (39)$$

Referring to Fluid Mechanics text books [39], the drag coefficient of a disk of diameter D is about 1.17, but around 2 for a wall

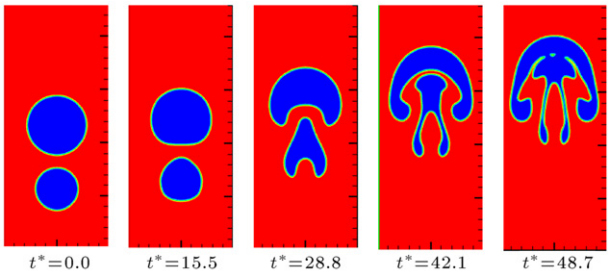

Figure 9: Time evolution of two rising bubbles in non-dimensional time, t^*, for Re = 556 and Bo = 800 using present method (LBLSM).

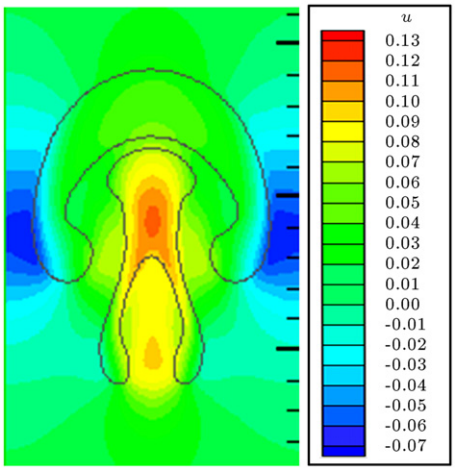

Figure 10: Upward velocity contours of two buoyant bubbles at $t^* = 42.1$.

with a width of D and the same thickness. So the difference between our 2D/Cartesian simulation and experiments in terminal velocity can be justified. To get better results, we may extend our code to 3D. We may also use a projection method and get better conservation properties solving Poisson's equation.

4.2. Coalescence of two rising bubbles

The coalescence of two successive buoyant bubbles is a benchmark problem for validation of an interface capturing code. The ratio of bubble diameters is 1.4, and both are of the same density and viscosity. The dimensionless parameters (based on the diameter of the smaller bubble) are Re = 566 and Bo = 800, but the process is similar for various Re and Bo numbers. The simulation is implemented in a 350 × 100 mesh and $\Delta x = \Delta y = \Delta t = 0.001$. Zero gradient boundary conditions are imposed on all walls. A buoyancy force equal to $0.0981 \times (\rho_l - \rho_b)$ was exerted onto both bubbles. The coalescence process and the upward velocity contours are shown in Figures 9 and 10, respectively. The maximum reduction in mass during the process was 0.02. The non-dimensional time is defined as $t^* = t\sqrt{\frac{g}{D}}$, where D is the diameter of the smaller droplet.

It is detected that the shapes are qualitatively in accordance with the computational work of Nagrath et al. [9], besides which, in their research, it was mentioned that the pressure gradient between the bottom of the larger bubble and the top of the smaller one imposes a strong flow at this distance, so the pressure gradient is an important factor in the merging process.

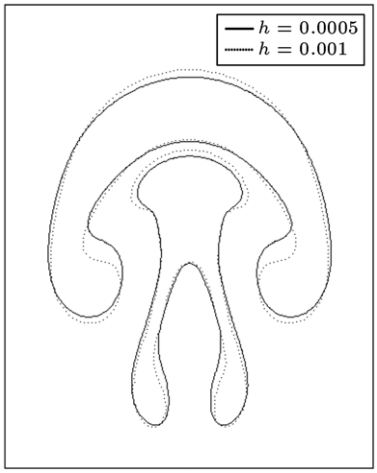

Figure 11: Investigation of the grid size on the shape of two rising bubbles.

As our simulations compare well with the results of Nagrath et al. [9], the LBLSM is able to model pressure effects properly.

To check the correctness of the results, the simulations have been implemented for finer mesh sizes; in cases of simple topological change (Figures 2–6), even a course mesh can handle similar outputs. In cases of more complex bubble shapes (Figures 7–10), if we use a course grid, we may lose some details, such as narrow tails or small satellite bubbles. If we use a finer grid than that referred to in the article, the variations in general shape and topological changes would not be considerable, even in cases of bubble structures as complex as Figure 9. After increasing the number of nodes by four, the shape is almost the same, which shows that the results are independent of the grid (see Figure 11).

5. Conclusion

A hybrid Lattice Boltzmann Level Set Method (LBLSM) for incompressible two-phase fluids with large density differences has been extended to include pressure effects. The method can simulate two-phase flows with the density ratio up to 1000, and viscosity ratio up to 100. The rising of a single buoyant bubble and the merging of two rising bubbles have been simulated by the current method. The simulations compare well with other numerical or experimental results. The method is convenient for capturing complex geometries.

Acknowledgement

This work has been undertaken with the support of the Vice-Chancellor for Research at Sharif University of Technology, Iran.

References

[1] Scardovelli, R. and Zaleski, S. "Direct numerical simulation of free-surface and interfacial flow", *Ann. Rev. Fluid Mech.*, 31, pp. 567–603 (1999).
[2] Shyy, W., Udaykumar, H.S., Rao, M.M. and Smith, R.W., *Computational Fluid Dynamics with Moving Boundaries*, Taylor & Francis (1996).
[3] Hartunian, R.A. and Sears, W.R. "On the instability of small gas bubbles moving uniformly in various liquids", *J. Fluid Mech.*, 3, pp. 27–47 (1957).
[4] Walters, J.K. and Davidson, J.F. "The initial motion of a gas bubble formed in an inviscid liquid. Part 1. The two-dimensional bubble", *J. Fluid Mech.*, 12, pp. 408–417 (1962).
[5] Walters, J.K. and Davidson, J.F. "The initial motion of a gas bubble formed in an inviscid liquid. Part 2. The three dimensional bubble and the toroidal bubble", *J. Fluid Mech.*, 17, pp. 321–336 (1963).
[6] Grace, J.R. "Shapes and velocities of bubbles rising in infinite liquids", *Trans. Inst. Chem. Engrg.*, 51, pp. 116–120 (1973).
[7] Bhaga, D. and Weber, M.E. "Bubbles in viscous liquids: shapes, wakes, and velocities", *J. Fluid Mech.*, 105, pp. 61–85 (1981).
[8] Hua, J. and Lou, J. "Numerical simulation of bubble rising in viscous liquid", *J. Comput. Phys.*, 222, p. 769 (2007).
[9] Nagrath, S., Jansen, K.E. and Lahey, R.T. "Computation of incompressible bubble dynamics with a stabilized finite element level set method", *Comput. Methods Appl. Mech. Engrg.*, 194, pp. 45–65 (2005).
[10] Best, J.P. "The formation of toroidal bubbles upon the collapse of transient cavities", *J. Fluid Mech.*, 251, p. 79 (1993).
[11] Hughes, T.J.R., Liu, W.K. and Zimmermann, T.K. "Lagrangian–Eulerian finite element formulation for incompressible viscous flows", *Comput. Methods Appl. Mech. Engrg.*, 29, p. 329 (1981).
[12] Hirt, C.W. and Nichols, B.D. "Volume of fluid (VOF) method for the dynamics of free boundaries", *J. Comput. Phys.*, 39, p. 201 (1981).
[13] Sethian, J.A., *Level Set Methods and Fast Marching Methods*, Cambridge University Press (1999).
[14] Sussman, M., Smereka, P. and Osher, S.J. "A level set approach for computing solutions to incompressible two-phase flows", *J. Comput. Phys.*, 114, p. 146 (1994).
[15] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag (2004).
[16] Sussman, M., Fatemi, E., Smereka, P. and Osher, S. "An improved level set method for incompressible two-fluid flows", *Comput. & Fluids*, 27, p. 663 (1998).
[17] Gresho, P.M. and Sani, R.L., *Incompressible Flow & Finite Element V 1*, John Wiley & Sons (2000).
[18] Franca, L.P., Frey, S.L. and Hughes, T.J.R. "Stabilized finite element methods: I. application to the advective-diffusive model", *Comput. Methods Appl. Mech. Engrg.*, 95, p. 253 (1992).
[19] Yu, D., Mei, R., Luo, L.S. and Shyy, W. "Viscous flow computations with the method of lattice Boltzmann equation", *Prog. Aerosp. Sci.*, 39, p. 329 (2003).
[20] Lallemand, P., Luo, L.S. and Peng, Y. "A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two dimensions", *J. Comput. Phys.*, 226, pp. 1367–1384 (2007).
[21] Peskin, C.S. "The immersed boundary method", *Acta Numer.*, 11, pp. 479–517 (2002).
[22] Mehravaran, M. and Hannani, S.K. "Simulation of incompressible two-phase flows with large density differences employing lattice Boltzmann and level set methods", *Comput. Methods Appl. Mech. Engrg.*, 198, p. 223 (2008).
[23] Yu, Z. and Fan, L.S. "An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation", *J. Comput. Phys.*, 228, pp. 6456–6478 (2009).
[24] Thommes, G., Becker, J., Junk, M., Vaikuntam, A.K., Kehrwald, D., Klar, A., Steiner, K. and Wiegmann, A. "A lattice Boltzmann method for immiscible multiphase flow simulations using the level set method", *J. Comput. Phys.*, 228, pp. 1139–1156 (2009).
[25] Servan-Camas, B. and Tsai, F.T.-C. "Non-negativity and stability analysis of lattice Boltzmann method for advection-diffusion equation", *J. Comput. Phys.*, 228, pp. 236–256 (2009).
[26] Lee, T. and Lin, C.L. "A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio", *J. Comput. Phys.*, 206, pp. 16–47 (2005).
[27] Zheng, H.W., Shu, C. and Chew, Y.T. "A lattice Boltzmann model for multiphase flows with large density ratio", *J. Comput. Phys.*, 218, pp. 353–317 (2006).
[28] Bhatnagar, P.L., Gross, E.P. and Krook, M. "A model for collision process in gases. I. Small amplitude process in charged and neural one-component system", *Phys. Rev.*, 94, p. 511 (1954).
[29] Qian, Y.H., d'Humieres, D. and Lallemand, P. "Lattice BGK models for Navier Stokes equation", *Europhys. Lett.*, 17, p. 479 (1992).
[30] Zhou, G.G., *Lattice Boltzmann Method for Shallow Water Flows*, Springer-Verlag (2004).
[31] Chapman, S. and Cowling, T.G., *The Mathematical Theory of Nonuniform Gases*, Cambridge University Press, Cambridge (1970).
[32] Shi, Y., Zhao, T.S. and Guo, Z.L. "Lattice Boltzmann method for incompressible flows with large pressure gradients", *Phys. Rev. E*, 73, pp. 260–704 (2006).
[33] Adalsteinsson, D. and Sethian, J. "The fast construction of extension velocities in level set methods", *J. Comput. Phys.*, 48, pp. 2–22 (1999).
[34] Chopp, D.L. "Some improvements of the fast marching methods", *SIAM J. Sci. Comput.*, 23, pp. 230–244 (2001).
[35] Brackbill, J.U., Kothe, D.B. and Zemach, C. "A continuum method for modeling surface tension", *J. Comput. Phys.*, 100, p. 335 (1992).
[36] Unverdi, S.O. and Tryggvason, G. "A front-tracking method for viscous, incompressible, multifluid flows", *J. Comput. Phys.*, 100, p. 25 (1992).
[37] Chang, Y.C., Hou, T.Y., Merriman, B. and Osher, S. "Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces", *J. Comput. Phys.*, 124, p. 449 (1996).

[38] Smolianski, A., Haario, H. and Luukka, P. "Vortex shedding behind a rising bubble and two-bubble coalescence: a numerical approach", *Appl. Math. Model.*, 29, p. 615 (2005).
[39] Shames, I.H., *Mechanics of Fluids*, McGraw-Hill (1992).

Meisam Mehravaran obtained his B.S. degree in Solid Mechanics from Amir Kabir University of Technology. Subsequently, he continued towards an M.S. degree in Energy Conversion at Sharif University of Technology, and focused on combining lattice Boltzmann and level set methods. Currently, he is a Ph.D. student at Michigan State University, working on Transport and Sedimentation of Sub-Micron Particles due to Thermophoresis.

Siamak Kazemzadeh Hannani is currently Professor of Mechanical Engineering at Sharif University of Technology. He obtained his Ph.D. in Turbulence Simulation, using the Finite Element Method, from Lille University in France. His research interests include Turbulence Modeling and Energy System Simulation.