

This article was downloaded by:[Dhir, Vijay K.]
[Dhir, Vijay K.]

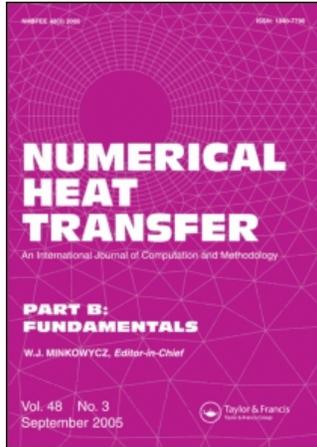
On: 24 April 2007

Access Details: [subscription number 777601808]

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954

Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Numerical Heat Transfer, Part B: Fundamentals

An International Journal of Computation and Methodology

Publication details, including instructions for authors and subscription information:
<http://www.informaworld.com/smpp/title-content=t713723316>

Numerical Simulation of Subcooled Nucleate Boiling by Coupling Level-Set Method with Moving-Mesh Method

To cite this Article: , 'Numerical Simulation of Subcooled Nucleate Boiling by
Coupling Level-Set Method with Moving-Mesh Method', Numerical Heat Transfer,
Part B: Fundamentals, 51:6, 535 - 563

To link to this article: DOI: 10.1080/10407790601177763

URL: <http://dx.doi.org/10.1080/10407790601177763>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article maybe used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

© Taylor and Francis 2007

NUMERICAL SIMULATION OF SUBCOOLED NUCLEATE BOILING BY COUPLING LEVEL-SET METHOD WITH MOVING-MESH METHOD

Jinfeng Wu and Vijay K. Dhir

Henry Samueli School of Engineering and Applied Science, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, Los Angeles, California, USA

Jianliang Qian

Department of Mathematics and Statistics, Wichita State University, Wichita, Kansas, USA

A new numerical procedure coupling the level-set method with the moving-mesh method to simulate subcooled nucleate pool boiling is proposed. Numerical test problems have validated this new method. The simulation of bubble dynamics during nucleate boiling under liquid subcooling shows that this novel adaptive method is more accurate in determining interfacial heat transfer than a computational method based on uniform grids with the same number of mesh points.

1. INTRODUCTION

Nucleate boiling is a liquid–vapor phase-change process associated with bubble formation. Subcooled nucleate pool boiling exists when the bulk temperature in the liquid pool is less than the saturation temperature of the liquid at the given system pressure, and the temperature of the heating surface exceeds a nucleation temperature which is higher than the saturation temperature. As it is a very efficient mode of heat transfer, during the last few decades major efforts have been made to develop numerical techniques to model this phase-change process. Lee and Nydahl [1] modeled a bubble as consisting of a wedge-shaped microlayer and a hemisphere. Mei et al. [2] adopted two empirically determined constants to numerically model the growth rate of vapor bubbles in saturated heterogeneous boiling. Welch [3] presented a finite-volume method to capture the interface and simulate the hydrodynamics, in which transient conduction in the solid was also included. Yoon et al. [4] presented a mesh-free numerical method to solve the Navier-Stokes equations and the energy

Received 14 August 2006; accepted 22 November 2006.

We are grateful to Gihun Son and Gopinath Warriar for helpful remarks on this work. This work is supported by NASA under the Microgravity Fluid Physics Program. Jianliang Qian is supported by NSF Grant 0542174.

Address correspondence to Vijay K. Dhir, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, Engineering IV Building, Rm 38-137H, Los Angeles, CA 90095, USA. E-mail: vdhir@seas.ucla.edu

NOMENCLATURE

<p>A dispersion constant</p> <p>c_p specific heat at constant pressure</p> <p>\vec{g} gravity vector</p> <p>G monitor function</p> <p>h grid spacing for the macro region</p> <p>h_{ev} evaporative heat transfer coefficient</p> <p>h_{fg} latent heat of evaporation</p> <p>H step function</p> <p>J Jacobian of transformation</p> <p>Ja Jacob number ($=c_{pl} \Delta T / h_{fg}$)</p> <p>$k$ thermal conductivity</p> <p>l_0 characteristic length</p> <p>\vec{m} mass flux vector</p> <p>M molecular weight</p> <p>Nu Nusselt number ($=ql_0/k_l \Delta T$)</p> <p>p pressure</p> <p>Pe Peclet number ($=\rho_l c_{pl} u_0 l_0 / k_l$)</p> <p>$q$ heat flux</p> <p>r radial coordinate</p> <p>\dot{r} mesh velocity in r direction</p> <p>R radius of computational domain or bubble</p> <p>\bar{R} universal gas constant</p> <p>R_0 radius of dry region beneath a bubble</p> <p>R_1 radial location of the interface at $y = h/2$</p> <p>Re Reynolds number ($=\rho_l u_0 l_0 / \mu_l$)</p> <p>t time</p> <p>t_0 characteristic time</p> <p>T temperature</p> <p>ΔT temperature difference</p> <p>\vec{u} velocity vector, (u, v)</p> <p>\vec{u}_{int} interfacial velocity vector</p> <p>\hat{u} contravariant velocity normal to η line</p> <p>u_0 characteristic velocity</p> <p>U flow rate normal to η line</p> <p>\hat{v} contravariant velocity normal to ξ line</p> <p>V flow rate normal to ξ line</p> <p>V_{micro} rate of vapor volume production from the microlayer</p> <p>ΔV_{micro} vapor-side control volume</p> <p>w weight function in monitor function</p> <p>We Weber number ($(\rho_l u_0^2 l_0 / \sigma)$)</p>	<p>y vertical coordinate</p> <p>\dot{y} mesh velocity in y direction</p> <p>Y height of computational domain</p> <p>α thermal diffusivity</p> <p>β_T coefficient of thermal expansion</p> <p>δ liquid-film thickness</p> <p>δ_0 nonevaporating liquid-film thickness</p> <p>δ_t thermal-layer thickness</p> <p>ζ similarity variable</p> <p>η curvilinear coordinate</p> <p>η' coordinate in computational domain</p> <p>θ dimensionless temperature</p> <p>θ'_1 a constant</p> <p>κ interfacial curvature</p> <p>μ dynamic viscosity</p> <p>ν_l liquid kinematic viscosity</p> <p>ξ curvilinear coordinate</p> <p>ξ' coordinate in computational domain</p> <p>ρ density</p> <p>σ surface tension</p> <p>τ artificial time</p> <p>ϕ level-set function</p> <p>φ contact angle</p> <p>ψ general dependent variable</p> <p>Ω_p physical domain</p> <p>Subscripts</p> <p>c at the center of control volume</p> <p>e at the edge of control volume</p> <p>int interface</p> <p>l liquid</p> <p>sat saturation</p> <p>sub subcooling</p> <p>sup superheat</p> <p>v vapor</p> <p>w wall</p> <p>η partial differentiation with respect to η</p> <p>ξ partial differentiation with respect to ξ</p> <p>Superscripts</p> <p>n time step n</p> <p>$n+1$ time step $n+1$</p> <p>$*$ intermediate value</p> <p>$**$ intermediate value</p>
---	---

equation; in their work, the interface was accurately traced by following the motion of computational points. In the studies cited above, several assumptions were made, such as bubbles having a hemispherical or spherical shape, uniform temperature around bubbles, or neglect of heat transfer from microlayer. Son et al. [5] presented a bubble-growth model for an isolated bubble based on the level-set method. This

method was previously used by Sussman et al. [6] to solve for an adiabatic incompressible two-phase flow. The level-set model as further developed by Son et al. [5] offers many advantages over previously published models: it takes care of the initial conditions automatically, it can be used for both saturated and subcooled boiling, and it gives the distribution of the wall heat flux, the microlayer contribution, and the interface heat transfer. In this work we extend this model to numerical simulation of subcooled nucleate boiling.

Since the introduction of the level-set method (Osher and Sethian [7]), it has become a popular method for tracking interfaces, owing to its automatic handling of topological changes and easy implementation in comparison to other methods such as surface-fitted grid methods and volume-of-fluid (VOF) methods.

Numerical investigation for two-phase problems requires extremely fine meshes over the interface region to capture the process dynamics accurately. One major goal of this study is to adopt a fine mesh to resolve the temperature gradient around the bubble interface during nucleate boiling in subcooled liquid and adopt coarser grids in regions away from the vapor–liquid interface. In contrast, using a uniformly refined mesh in the entire domain becomes computationally expensive. Thus an adaptive computational mesh is expected to increase the accuracy of interfacial heat transfer and decrease the cost of computation in comparison to using a uniform mesh.

Depending on the mesh geometry, in general, there are two types of meshes: Cartesian and nonorthogonal grids. The advantage of Cartesian grid methods with local refinement is in the simplification of the grid-generation process so that a new mesh is obtained by adding or removing points to achieve a desired level of accuracy. Rectangle-based adaptive mesh refinement methods (Berger and Oliger [8], Sussman [9], Qian [10]) and quad/octree-based methods (Khokhlov [11], Street [12], Cecil, Osher and Qian [13]) belong to the class of Cartesian-mesh-based adaptive methods. However, these methods require sophisticated data structures to efficiently manage the set of locally refined Cartesian cells and their associated data and neighbor connections. As a result, the complicated data structure can pose difficulties in implementation.

The moving-mesh methods (MMM) and adaptive mesh redistribution (AMR) methods are usually based on nonorthogonal grids. Moving-mesh methods relocate a fixed number of nodes with fixed connectivity within the computational domain so as to increase numerical accuracy. During the redistribution process, grid orthogonality is not enforced. One of the frequently used procedures in generating meshes in moving-mesh methods is the variational method. This method generates meshes by minimizing a functional that involves various properties of physical solutions and computational meshes, such as grid smoothness, orthogonality and volume variation, etc., and transforming the solutions between the physical domain and the computational domain. As a result, the mesh evolves with the physical solution and maintains high adaptivity as the solution evolves. The key to moving-mesh adaptivity is using a proper monitor function. By using an appropriate monitor function Tang et al. [14] produced satisfactory mesh concentration for solving Hamilton-Jacobi equations. Cenicerros and Hou [15] improved and simplified the mesh-generation procedure proposed by Huang and Russell [16], and successfully applied it to the numerical simulation of Boussinesq flows.

In numerical simulations, the velocity components normal to the control-volume faces are usually used as the dependent variables. However, nonorthogonal meshes used in moving-mesh methods make it necessary to choose Cartesian, covariant, and contravariant velocity components as dependent variables. Nevertheless, the use of nonorthogonal curvilinear grids provides more flexibility with regard to the distribution of grid points. The use of Cartesian velocity components as primary dependent variables has the advantage that the governing equations remain in a relatively simple form. Moreover, covariant and contravariant components appear to be the perfect choices because of their relationship with grid lines. The discretizations for the covariant and contravariant velocity components are obtained by an algebraic manipulation of the corresponding equations for the Cartesian velocity components. The use of covariant or contravariant components results in extra source terms and more complicated equations. Karki and Patankar [17] presented a method using a staggered grid and covariant velocity components as primary variables. Yue et al. [18] coupled the incompressible Navier-Stokes system with a level-set function in curvilinear coordinates on a fixed grid which is nonuniform but orthogonal. Son and Dhir [19] adopted contravariant velocity components as dependent variables in curvilinear coordinates on a moving grid for film boiling. Their study provides the essential outline for the present project. In order to capture the temperature gradient accurately around the evolving bubble interface, we couple the level-set method with the moving-mesh method to carry out numerical simulations of bubble growth in subcooled nucleate boiling.

2. NUMERICAL FORMULATION

2.1. Model Description

To analyze the growth of a single bubble in subcooled nucleate boiling, we extend the numerical model originally developed by Son et al. [5]. In that model, the computational domain is divided into two parts, a micro region and a macro region, as shown in Figure 1. The micro region is a thin film that lies underneath the bubble, whereas the macro region consists of the bubble and the liquid surrounding the bubble. Numerical simulations of fluid flow and heat transfer are carried out for a time-dependent system in both the micro and macro regions. The computed shapes of the interface in the micro region and the macro region are matched at the outer edge of the micro layer for a given contact angle. In the numerical analysis, a level-set function is solved to represent the macro region.

2.2. Assumptions

The assumptions in this study are the following:

The process is two-dimensional and axisymmetric.

The flows are laminar.

The wall temperature remains constant.

The thermodynamic properties of the individual phases are assumed to be insensitive to small changes in temperature and pressure, except for surface tension.

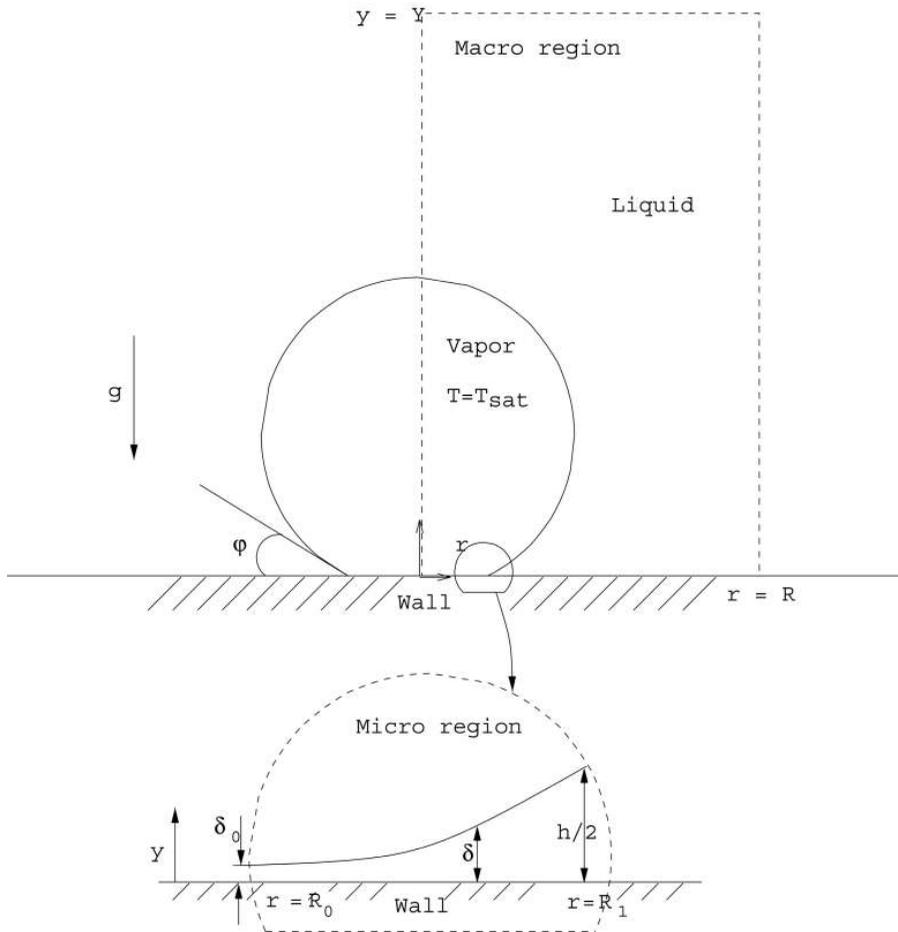


Figure 1. Macro and micro regions in numerical simulation.

2.3. Thermal and Physical Properties

The thermal and physical properties used in the computation are listed in Table 1. All properties are evaluated of water at atmospheric pressure and 100°C .

2.4. Governing Equations in the Macro Region

To numerically analyze the macro region, we use the level-set formulation developed by Son et al. [5] for nucleate boiling of pure liquid. The interface separating the two phases is captured by solving the following equation for the level-set function ϕ :

$$\frac{\partial \phi}{\partial t} = -\mathbf{u}_{\text{int}} \cdot \nabla \phi \quad (1)$$

Table 1. Thermal and physical properties

Property	Unit	Liquid	Vapor
ρ	kg/m ³	958	0.598
c_p	kJ/kg K	4.212	2.02
k	W/m K	0.68	0.0248
μ	m ² /s	2.85×10^{-4}	1.2×10^{-5}
h_{fg}	kJ/kg	2,257	
T_{sat}	K		373.15
β	K ⁻¹	7.5×10^{-4}	
σ	N/m	0.0589	

where

$$\mathbf{u}_{\text{int}} = \mathbf{u} + \frac{\mathbf{m}}{\rho} \quad (2)$$

A reinitialization equation is solved until steady state to ensure that $|\nabla\phi| = 1$:

$$\frac{\partial\phi}{\partial t} = \text{sign}(\phi_0)(1 - |\nabla\phi|) \quad (3)$$

In the above equation, ϕ_0 is the solution of Eq. (1). The material properties are assumed to be constant in the individual phases, except near the interface and in a thin region around the interface. To describe such an interface we define the Heaviside function, H , as follows:

$$\begin{aligned} H &= 1 && \text{if } \phi \geq +1.5h \\ &= 0 && \text{if } \phi \leq -1.5h \\ &= 0.5 + \phi/(3h) + \sin[2\pi\phi/(3h)]/(2\pi) && \text{if } |\phi| \leq 1.5h \end{aligned} \quad (4)$$

where h is equal to the grid spacing on a uniform grid, and H remains 1 in the liquid phase and 0 in the vapor phase. However, it is smoothed over an interval of $3h$, so that the material properties change continuously at the interface. The properties are defined as follows:

$$\rho = \rho_v + (\rho_l - \rho_v)H \quad (5)$$

$$\mu^{-1} = \mu_v^{-1} + (\mu_l^{-1} - \mu_v^{-1})H \quad (6)$$

$$k^{-1} = k_l^{-1}H \quad (7)$$

where ρ , μ , and k denote density, shear viscosity, and the thermal conductivity, respectively. The subscripts v and l represent the vapor and fluid states, respectively. Equation (7) is consistent with the assumption that the vapor temperature remains constant at T_{sat} .

The interfacial curvature is expressed in terms of the level-set function as follows:

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \quad (8)$$

Next, we present the governing equations of continuity, momentum, and energy for the macro region.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (9)$$

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) &= -\nabla p + \rho \mathbf{g} - \rho \beta_T (T - T_{\text{sat}}) \mathbf{g} - \sigma \kappa \nabla H \\ &+ \nabla \cdot \mu \nabla \mathbf{u} + \nabla \cdot \mu \nabla \mathbf{u}^T \end{aligned} \quad (10)$$

$$\begin{aligned} \rho c_{pl} \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) &= \nabla \cdot k \nabla T \quad \text{for } H > 0 \\ T &= T_{\text{sat}}(p_v) \quad \text{for } H = 0 \end{aligned} \quad (11)$$

The vapor in the bubble was assumed to remain at the saturation temperature. As such, the energy equation in the vapor is not solved. Since there is no temperature gradient along the interface, no Marangoni flow is considered in this work.

The mass conservation equation, Eq. (9), can be rewritten as

$$\nabla \cdot \mathbf{u} = \frac{\mathbf{m}}{\rho^2} \cdot \nabla \rho + \dot{V}_{\text{micro}} = \frac{k \nabla T}{h_{fg} \rho^2} \cdot \nabla \rho + \dot{V}_{\text{micro}} \quad (12)$$

where \dot{V}_{micro} is the volume expansion attributed to the heat transfer from the micro-layer, which is

$$\dot{V}_{\text{micro}} = \int_{R_0}^{R_1} \frac{k_l (T_w - T_{\text{int}})}{\rho_v h_{fg} \delta \Delta V_{\text{micro}}} r dr \quad (13)$$

ΔV_{micro} is a vapor-side control volume near the micro region. \mathbf{m} is the water-evaporation-rate vector, which is defined as (see Son et al. [5] for details)

$$\mathbf{m} = \rho(\mathbf{u}_{\text{int}} - \mathbf{u}) = \rho_l(\mathbf{u}_{\text{int}} - \mathbf{u}_l) = \rho_v(\mathbf{u}_{\text{int}} - \mathbf{u}_v) \quad (14)$$

Also,

$$\mathbf{m} = \frac{k \nabla T}{h_{fg}} \quad (15)$$

Equations (12), (10), and (11) are nondimensionalized using the characteristic length, time, and velocity scales l_0 , t_0 , and u_0 , respectively.

$$l_0 = \sqrt{\frac{\sigma}{g(\rho_l - \rho_v)}} \quad (16)$$

$$u_0 = \sqrt{gl_0} = \left[\frac{\sigma g}{(\rho_l - \rho_v)} \right]^{1/4} \quad (17)$$

$$t_0 = \frac{l_0}{u_0} = \left[\frac{\sigma}{g^3(\rho_l - \rho_v)} \right]^{1/4} \quad (18)$$

The temperature is nondimensionalized such that the wall temperature is 1 and the subcooled liquid temperature is 0, i.e.,

$$\theta = \frac{T - T_l}{T_w - T_l} \quad (19)$$

The governing equations (12), (10), and (11) are solved throughout the domain to obtain the velocity, temperature, and pressure in each cell. The detailed computational framework is discussed in a later section.

2.5. Governing Equations in the Micro Region

The micro region is illustrated in Figure 1. The thickness of the liquid layer varies from a couple of molecules to a few micrometers near the end, where it joins with the macro region. Lay and Dhir [20] have modeled and solved numerically for the shape of the microlayer underneath a bubble using the lubrication theory. In carrying out the analysis, δ is taken to be the thickness of the microlayer measured from the wall, and r is the radial coordinate. The mass conservation, momentum, and energy equations in the microlayer are given as

$$\frac{q}{\rho_l h_{fg}} = -\frac{1}{r} \frac{\partial}{\partial r} \int_0^\delta r u_l dy \quad (20)$$

$$\frac{\partial p_l}{\partial r} = \mu_l \frac{\partial^2 u_l}{\partial y^2} \quad (21)$$

$$q = \frac{k_l(T_w - T_{\text{int}})}{\delta} = h_{\text{ev}} \left[T_{\text{int}} - T_v + \frac{(p_l - p_v)T_v}{\rho_l h_{fg}} \right] \quad (22)$$

where T_w is the wall temperature, T_{int} is the interface temperature, T_v is the vapor temperature, p_v is the vapor pressure, and h_{ev} is the evaporation heat transfer coefficient. The evaporation heat transfer coefficient is obtained from kinetic theory as

$$h_{\text{ev}} = \frac{(2M/\pi \bar{R} T_v)^{0.5} \rho_v h_{fg}^2}{T_v} \quad T_v = T_{\text{sat}}(p_v) \quad (23)$$

The pressures in the vapor and liquid phases satisfy the following relation (Lay and Dhir [20]):

$$p_l = p_v - \sigma \kappa - \frac{A}{\delta^3} + \frac{q^2}{\rho_v h_{fg}^2} \quad (24)$$

where σ is a function of temperature and A is the dispersion constant in the disjoining pressure. In Eq. (24), the second term on the right-hand side accounts for the capillary pressure, the third term for the disjoining pressure, and the last term for the recoil pressure. The curvature of the interface is defined as

$$\kappa = \frac{1}{r} \frac{\partial}{\partial r} \left[r \frac{\partial \delta}{\partial r} / \sqrt{1 + \left(\frac{\partial \delta}{\partial r} \right)^2} \right] \quad (25)$$

The combination of the mass, momentum, and energy equations for the micro-layer yields

$$\delta'''' = f(\delta, \delta', \delta'', \delta''') \quad (26)$$

where $'$ denotes $\partial/\partial r$.

The boundary conditions for the above equation are posed as follows:

$$\begin{aligned} \text{At } r = R_0 (\text{inner radius of dry region}), \\ \delta = \delta_o \quad \delta' = \delta'' = 0 \end{aligned} \quad (27)$$

where δ_o is of the order of molecular size and can be obtained from Wayner [21] at the junction of the evaporating and nonevaporating regions.

$$\begin{aligned} \text{At } r = R_1 (\text{outer radius of dry region}), \\ \delta = \frac{h}{2} \quad \delta'' = 0 \end{aligned} \quad (28)$$

where $h/2$ is the vertical distance to the first computational node for the level-set function ϕ , on uniform grids from the wall. In implementing the above boundary conditions, the radius R_1 is determined from the solution of the macro region. For a given dispersion constant, the microlayer formulation, Eq. (26), and R_0 are solved with the five boundary conditions [Eqs. (27) and (28)]. In this work an apparent contact angle is defined as

$$\tan \varphi = \frac{0.5h}{R_1 - R_0} \quad (29)$$

φ is measurable experimentally and used as boundary condition in the level-set function. Equation (26) is integrated numerically using a Runge-Kutta method with a separate code. An expression for the rate at which vapor is produced from the micro-layer is shown in Eq. (13). Within moving meshes, the distance to the first computational node from the wall is no longer equal to $h/2$ on uniform grids. It also may vary slightly from time to time. The above equations are still valid with the varying distance, just as in the case of refined uniform grid sizes.

2.6. The Moving Mesh Method

In the following we use the mesh-generation procedure given by Cenicerros and Hou [15]. The mesh map from the computational domain to the physical domain is generated by minimizing the following functional:

$$E[\xi', \eta'] = \frac{1}{2} \int_{\Omega_p} (\nabla \xi'^T G^{-1} \nabla \xi' + \nabla \eta'^T G^{-1} \nabla \eta') dr dy \quad (30)$$

In the above, ξ' and η' are two coordinates in the computational domain, r and y are the corresponding ones in the physical domain; G is a given symmetric positive definite matrix, the so-called monitor function; and Ω_p denotes the physical domain. Some terms controlling mesh orthogonality and alignment are omitted in the above equation. After interchanging the dependent and independent variables and making the simplest choice for G (i.e., $G = wI$, I is the identity matrix), we end up with the following equations:

$$\nabla \cdot (w \nabla r) = 0, \nabla \cdot (w \nabla y) = 0 \quad (31)$$

where w is a weight function to be chosen and $\nabla = (\partial/\partial \xi', \partial/\partial \eta')^T$.

A standard procedure to solve Eq. (31) is to use the pseudo-transient method (Huang and Russell [16], Cenicerros and Hou [15]),

$$r_\tau = \nabla \cdot (w \nabla r) \quad (32)$$

$$y_\tau = \nabla \cdot (w \nabla y) \quad (33)$$

where τ is an artificial time. At $t = 0$, using an initial guess we solve Eqs. (32) and (33) by an iterative method up to a steady state and the resulting mesh is fit well with the initial conditions. The details about the effectiveness of this mesh generator can be found in Cenicerros and Hou [15]. However, after $t = 0$, τ is replaced by t , and Eqs. (32) and (33) become the governing equations to generate the adaptive mesh for the evolving interface.

2.7. Transformed Governing Equations in Curvilinear Coordinates

The following transformation formulas are used to obtain the equations in curvilinear coordinates:

$$\begin{aligned} J &= r_\xi y_\eta - r_\eta y_\xi \\ \psi_r &= \frac{1}{J} (y_\eta \psi_\xi - y_\xi \psi_\eta) \\ \psi_y &= \frac{1}{J} (-r_\eta \psi_\xi + r_\xi \psi_\eta) \\ \psi_{rr} &= \frac{1}{J} \left\{ [J^{-1} (y_\eta)^2 \psi_\xi]_\xi - (J^{-1} y_\eta y_\xi \psi_\eta)_\xi - (J^{-1} y_\xi y_\eta \psi_\xi)_\eta + [J^{-1} (y_\xi)^2 \psi_\eta]_\eta \right\} \\ \psi_{yy} &= \frac{1}{J} \left\{ [J^{-1} (r_\eta)^2 \psi_\xi]_\xi - (J^{-1} r_\eta r_\xi \psi_\eta)_\xi - (J^{-1} r_\xi r_\eta \psi_\xi)_\eta + [J^{-1} (r_\xi)^2 \psi_\eta]_\eta \right\} \end{aligned} \quad (34)$$

The contravariant velocity components \hat{u} and \hat{v} can be expressed in the form of Cartesian components u and v as

$$\begin{aligned}\hat{u} &= y_\eta u - r_\eta v \\ \hat{v} &= -y_\xi u + r_\xi v \\ U &= r\hat{u} \frac{d\eta_e}{d\eta_c} \\ V &= r\hat{v} \frac{d\xi_e}{d\xi_c}\end{aligned}\quad (35)$$

where $d\xi_e$ and $d\eta_e$ are the edge sizes of the control volume in the ξ and η directions, while $d\xi_c$ and $d\eta_c$ are the centerline sizes of the control volume in the ξ and η directions. The computation of mesh distribution shows that the difference of $d\xi$ and $d\eta$ between neighboring grids becomes larger when the grids are farther away from the interface. These terms are introduced to correct the effect of nonuniform grids.

Using the above equations and some algebraic manipulations, we obtain the following transformed equations in curvilinear coordinates:

$$\begin{aligned}U_\xi + V_\eta &= \frac{rk}{h_{fg}\rho^2 J^2} \left[\frac{d\eta_e}{d\eta_c} T_\xi \rho_\xi - \frac{d\eta_e}{d\eta_c} (r_\xi r_\eta + y_\xi y_\eta) T_\xi \rho_\eta \right. \\ &\quad \left. - \frac{d\xi_e}{d\xi_c} (r_\xi r_\eta + y_\xi y_\eta) T_\eta \rho_\xi + \frac{d\xi_e}{d\xi_c} T_\eta \rho_\eta \right] + \dot{V}_{\text{micro}}\end{aligned}\quad (36)$$

$$\begin{aligned}& \rho r \left\{ J \frac{\partial \hat{u}}{\partial t} + y_\eta u_\xi [(u - \dot{r})y_\eta - (v - \dot{y})r_\eta] \frac{d\eta_e}{d\eta_c} + y_\eta u_\eta [-(u - \dot{r})y_\xi + (v - \dot{y})r_\xi] \frac{d\xi_e}{d\xi_c} \right. \\ & \quad \left. - r_\eta v_\xi [(u - \dot{r})y_\eta - (v - \dot{y})r_\eta] \frac{d\eta_e}{d\eta_c} + r_\eta v_\eta [(u - \dot{r})y_\xi - (v - \dot{y})r_\xi] \frac{d\xi_e}{d\xi_c} \right\} \\ &= -rp_\xi + (r_\xi r_\eta + y_\xi y_\eta) r p_\eta - \sigma k r H_\xi + (r_\xi r_\eta + y_\xi y_\eta) \sigma k r H_\eta \\ & \quad - 2\mu J y_\eta u / r + [1 - \beta_T (T - T_{\text{sat}})] \rho g J r r_\eta \\ & \quad + y_\eta \left\{ \left[\frac{2r\mu d\eta_e}{d\eta_c J} (y_\eta^2 u_\xi - y_\eta y_\xi u_\eta) \right]_\xi + \left[\frac{2r\mu d\xi_e}{d\xi_c J} (y_\xi^2 u_\eta - y_\eta y_\xi u_\xi) \right]_\eta \right\} \\ & \quad + y_\eta \left\{ - \left[\frac{r\mu d\eta_e}{d\eta_c J} (y_\eta r_\eta v_\xi - y_\xi r_\eta v_\eta) \right]_\xi + \left[\frac{r\mu d\xi_e}{d\xi_c J} (r_\xi y_\eta v_\xi - y_\xi r_\xi v_\eta) \right]_\eta \right\} \\ & \quad + y_\eta \left\{ \left[\frac{r\mu d\eta_e}{d\eta_c J} (r_\eta^2 u_\xi - r_\eta r_\xi u_\eta) \right]_\xi + \left[\frac{r\mu d\xi_e}{d\xi_c J} (r_\xi^2 u_\eta - r_\eta r_\xi u_\xi) \right]_\eta \right\} \\ & \quad - r_\eta \left\{ \left[\frac{r\mu d\eta_e}{d\eta_c J} (y_\eta^2 v_\xi - y_\eta y_\xi v_\eta) \right]_\xi + \left[\frac{r\mu d\xi_e}{d\xi_c J} (y_\xi^2 v_\eta - y_\eta y_\xi v_\xi) \right]_\eta \right\} \\ & \quad - r_\eta \left\{ \left[\frac{2r\mu d\eta_e}{d\eta_c J} (r_\eta^2 v_\xi - r_\eta r_\xi v_\eta) \right]_\xi + \left[\frac{2r\mu d\xi_e}{d\xi_c J} (r_\xi^2 v_\eta - r_\eta r_\xi v_\xi) \right]_\eta \right\} \\ & \quad - r_\eta \left\{ - \left[\frac{r\mu d\eta_e}{d\eta_c J} (r_\eta y_\eta u_\xi - y_\eta r_\xi u_\eta) \right]_\xi - \left[\frac{r\mu d\xi_e}{d\xi_c J} (y_\xi r_\xi u_\eta - r_\eta y_\xi u_\xi) \right]_\eta \right\}\end{aligned}\quad (37)$$

$$\begin{aligned}
\rho r \left\{ J \frac{\partial \hat{v}}{\partial t} - y_\xi u_\xi [(u - \dot{r})y_\eta - (v - \dot{y})r_\eta] \frac{d\eta_e}{d\eta_c} + y_\xi u_\eta [(u - \dot{r})y_\xi - (v - \dot{y})r_\xi] \frac{d\xi_e}{d\xi_c} \right. \\
\left. + r_\xi v_\xi [(u - \dot{r})y_\eta - (v - \dot{y})r_\eta] \frac{d\eta_e}{d\eta_c} + r_\xi u_\eta [-(u - \dot{r})y_\xi + (v - \dot{y})r_\xi] \frac{d\xi_e}{d\xi_c} \right\} \\
= -rp_\eta + (r_\xi r_\eta + y_\xi y_\eta)rp_\xi - \sigma\kappa r H_\eta + (r_\xi r_\eta + y_\xi y_\eta)\sigma\kappa r H_\xi \\
+ 2\mu J y_\xi u / r - [1 - \beta_T(T - T_{\text{sat}})]\rho g J r r_\xi \\
- y_\xi \left\{ \left[\frac{2r\mu d\eta_e}{d\eta_c J} (y_\eta^2 u_\xi - y_\eta y_\xi u_\eta) \right]_\xi + \left[\frac{2r\mu d\xi_e}{d\xi_c J} (y_\xi^2 u_\eta - y_\eta y_\xi u_\xi) \right]_\eta \right\} \\
- y_\xi \left\{ - \left[\frac{r\mu d\eta_e}{d\eta_c J} (y_\eta r_\eta v_\xi - y_\xi r_\eta v_\eta) \right]_\xi + \left[\frac{r\mu d\xi_e}{d\xi_c J} (r_\xi y_\eta v_\xi - y_\xi r_\xi v_\eta) \right]_\eta \right\} \\
- y_\xi \left\{ \left[\frac{r\mu d\eta_e}{d\eta_c J} (r_\eta^2 u_\xi - r_\eta r_\xi u_\eta) \right]_\xi + \left[\frac{r\mu d\xi_e}{d\xi_c J} (r_\xi^2 u_\eta - r_\eta r_\xi u_\xi) \right]_\eta \right\} \\
+ r_\xi \left\{ \left[\frac{r\mu d\eta_e}{d\eta_c J} (y_\eta^2 v_\xi - y_\eta y_\xi v_\eta) \right]_\xi + \left[\frac{r\mu d\xi_e}{d\xi_c J} (y_\xi^2 v_\eta - y_\eta y_\xi v_\xi) \right]_\eta \right\} \\
+ r_\xi \left\{ \left[\frac{2r\mu d\eta_e}{d\eta_c J} (r_\eta^2 v_\xi - r_\eta r_\xi v_\eta) \right]_\xi + \left[\frac{2r\mu d\xi_e}{d\xi_c J} (r_\xi^2 v_\eta - r_\eta r_\xi v_\xi) \right]_\eta \right\} \\
+ r_\xi \left\{ - \left[\frac{r\mu d\eta_e}{d\eta_c J} (r_\eta y_\eta u_\xi - y_\eta r_\xi u_\eta) \right]_\xi - \left[\frac{r\mu d\xi_e}{d\xi_c J} (y_\xi r_\xi u_\eta - r_\eta y_\xi u_\xi) \right]_\eta \right\} \quad (38)
\end{aligned}$$

$$\begin{aligned}
\rho c_p r \left\{ J \frac{\partial T}{\partial t} + T_\xi [(u - \dot{r})y_\eta - (v - \dot{y})r_\eta] \frac{d\eta_e}{d\eta_c} + T_\eta [-(u - \dot{r})y_\xi + (v - \dot{y})r_\xi] \frac{d\xi_e}{d\xi_c} \right\} \\
= \left\{ \frac{rk d\eta_e}{J d\eta_c} [T_\xi - (y_\eta y_\xi + r_\eta r_\xi) T_\eta] \right\}_\xi + \left\{ \frac{rk d\xi_e}{J d\xi_c} [T_\eta - (y_\eta y_\xi + r_\eta r_\xi) T_\xi] \right\}_\eta \quad (39)
\end{aligned}$$

$$\begin{aligned}
J \frac{\partial \phi}{\partial t} + \phi_\xi [(u_{\text{int}} - \dot{r})y_\eta - (v_{\text{int}} - \dot{y})r_\eta] + \phi_\eta [-(u_{\text{int}} - \dot{r})y_\xi + (v_{\text{int}} - \dot{y})r_\xi] = 0 \\
u_{\text{int}} = u + \frac{k}{J\rho h_{fg}} (y_\eta T_\xi - y_\xi T_\eta) \\
v_{\text{int}} = v + \frac{k}{J\rho h_{fg}} (-x_\eta T_\xi + x_\xi T_\eta) \quad (40)
\end{aligned}$$

$$\frac{\partial \phi}{\partial \tau} = \text{sign}(\phi_0) \left[1 - J^{-1} \sqrt{\phi_\xi^2 + \phi_\eta^2 - 2(y_\xi y_\eta + r_\xi r_\eta) \phi_\xi \phi_\eta} \right] \quad (41)$$

$$\text{sign}(\phi_0) = \frac{2}{\pi} \tan^{-1} \left[\left(\frac{\phi_0}{1.5h} \right)^3 64 \right] \quad (42)$$

Here $\text{sign}(\phi_0)$ is motivated by Qian and Leung [22], and it works well in our application.

3. THE COMPUTATIONAL FRAMEWORK

Taking advantage of the bubble symmetry, we need to compute only half of the bubble. We use a staggered-grid finite-difference scheme. The scalar parameters are defined at the centers of cells (i.e., centers of $d\xi_c$ and $d\eta_c$) and velocity components are stored at the edges of cells (i.e., \hat{u} at the center of $d\eta_e$ and \hat{v} at the center of $d\xi_e$). To easily obtain the discretized forms of various quantities, both Cartesian velocity components and contravariant ones are stored in memory. We use upwind differencing for advection terms and central differencing for diffusion terms. A projection method is used to solve velocities and pressure. Because the pressure computation takes most of the computational time, we combine multigrid and conjugate gradient methods so that the numerical solver for pressure converges in less than 10 iterations under most circumstances, resulting in significant computational savings.

Overall, we have the following computational framework:

1. Initialize by solving the mesh equation to steady state.

The mesh equation can be discretized as,

$$\frac{r^{n+1} - r^n}{\Delta\tau} = a \Delta r^{n+1} + \nabla \cdot (w^n \nabla r^n) - a \Delta r^n \quad (43)$$

$$\frac{y^{n+1} - y^n}{\Delta\tau} = a \Delta y^{n+1} + \nabla \cdot (w^n \nabla y^n) - a \Delta y^n \quad (44)$$

where $\alpha = \max(w^n)$. The superscript $n + 1$ denotes the current time step, while the superscript n denotes the previous time step. The function w is the key factor for obtaining a satisfactory mesh. Since the interface is the region of interest, it is natural to consider the level-set function ϕ . Consequently, we consider two different w 's: one is $w = \sqrt{1 + (\phi + 0.5h)^{-1}}$ and the other is $w = C^{1-|\phi|/|\phi|_{\max}}$, with C being a constant and $|\phi|_{\max}$ being the maximum absolute level-set function along the ξ or η direction. It also should be noted that the variations of other variables can be easily taken into account in the adaptation process.

2. Update mesh.

Replacing $\Delta\tau$ by Δt from the above equations, solve for one time step to evolve the mesh in the forward direction.

3. Solve the level-set advection equation, reinitialize the level-set function, and determine the properties.

The second-order ENO scheme is applied to discretize ϕ_ξ and ϕ_η . A few iteration steps are used in the reinitialization procedure. We use the length scale of $3h$ to

bridge the property difference through the interface with uniform grids. With nonuniform grids, the mesh concentration around the interface yields the possibility of narrowing this thickness, and this possibility deserves further investigation.

4. Solve the energy equation for temperature.

Upwind differencing is used for T_{ξ} and T_{η} . The diffusion terms of $T_{\xi\xi}$ and $T_{\eta\eta}$ are implicitly discretized, and $T_{\xi\eta}$ and $T_{\eta\xi}$ are continuously updated by the current iteration until convergence.

5. Solve the momentum equation for velocity using the pressure at the previous time step. To make the equations concise, define

$$L(\psi) = \left\{ \frac{r\mu d\eta_e}{Jd\eta_c} [\psi_{\xi} - (y_{\eta}y_{\xi} + r_{\eta}r_{\xi})\psi_{\eta}] \right\}_{\xi} + \left\{ \frac{r\mu d\xi_e}{Jd\xi_c} [\psi_{\eta} - (y_{\eta}y_{\xi} + r_{\eta}r_{\xi})\psi_{\xi}] \right\}_{\eta} \quad (45)$$

The following equations are computed for intermediate velocities \hat{u}^* and \hat{v}^* :

$$\rho Jr \frac{(\hat{u}^* - \hat{u}^n)}{\Delta t} = -rp_{\xi}^n + (r_{\xi}r_{\eta} + y_{\xi}y_{\eta})rp_{\eta}^n + s1(\hat{u}^n) + L(\hat{u}^*) - L(\hat{u}^n) \quad (46)$$

$$\rho Jr \frac{(\hat{v}^* - \hat{v}^n)}{\Delta t} = -rp_{\eta}^n + (r_{\xi}r_{\eta} + y_{\xi}y_{\eta})rp_{\xi}^n + s2(\hat{v}^n) + L(\hat{v}^*) - L(\hat{v}^n) \quad (47)$$

where $s1(\hat{u}^n)$ and $s2(\hat{v}^n)$ represent the other terms from Eqs. (37) and (38), respectively.

6. Correct the velocities.

$$\hat{u}^{**} = \hat{u}^* + \frac{\Delta t}{\rho J} [p_{\xi}^n - (r_{\xi}r_{\eta} + y_{\xi}y_{\eta})p_{\eta}^n] \quad (48)$$

$$\hat{v}^{**} = \hat{v}^* + \frac{\Delta t}{\rho J} [p_{\eta}^n - (r_{\xi}r_{\eta} + y_{\xi}y_{\eta})p_{\xi}^n] \quad (49)$$

7. Solve the following Poisson equation for pressure:

$$\begin{aligned} & \left\{ \frac{r \Delta t d\eta_e}{\rho J d\eta_c} [p_{\xi}^{n+1} - (r_{\xi}r_{\eta} + y_{\xi}y_{\eta})p_{\eta}^{n+1}] \right\}_{\xi} + \left\{ \frac{r \Delta t d\xi_e}{\rho J d\xi_c} [p_{\eta}^{n+1} - (r_{\xi}r_{\eta} + y_{\xi}y_{\eta})p_{\xi}^{n+1}] \right\}_{\eta} \\ & = -U_{\xi}^{**} - V_{\eta}^{**} \\ & + \frac{rk}{h_{fg}\rho^2 J^2} \left[\frac{d\eta_e}{d\eta_c} T_{\xi} \rho_{\xi} - \frac{d\eta_e}{d\eta_c} (r_{\xi}r_{\eta} + y_{\xi}y_{\eta}) T_{\xi} \rho_{\eta} \right. \\ & \left. - \frac{d\xi_e}{d\xi_c} (r_{\xi}r_{\eta} + y_{\xi}y_{\eta}) T_{\eta} \rho_{\xi} + \frac{d\xi_e}{d\xi_c} T_{\eta} \rho_{\eta} \right] + \dot{V}_{micro} \quad (50) \end{aligned}$$

Under the assumptions of constant wall temperature and constant contact angle, \dot{V}_{micro} is a function of the distance between the first computational nodes and the wall. This function can be fit from data collected by varying that distance. During each time step, the average distance of several first computational nodes

at the interface region is used to retrieve the corresponding micro-layer heat transfer from that function.

8. Correct the velocity.

$$\hat{u}^{n+1} = \hat{u}^{**} - \frac{\Delta t}{\rho J} \left[p_{\xi}^{n+1} - (r_{\xi} r_{\eta} + y_{\xi} y_{\eta}) p_{\eta}^{n+1} \right] \quad (51)$$

$$\hat{v}^{n+1} = \hat{v}^{**} - \frac{\Delta t}{\rho J} \left[p_{\eta}^{n+1} - (r_{\xi} r_{\eta} + y_{\xi} y_{\eta}) p_{\xi}^{n+1} \right] \quad (52)$$

After this step, the continuity equation is ensured to be satisfied.

9. Go to Step 2 for the next time step.

3.1. Boundary Conditions

At the wall ($y = 0$),

$$u = 0 \quad v = 0 \quad \frac{\partial \phi}{\partial y} = -\cos \varphi \quad T = T_w \quad (53)$$

where the third boundary condition is accurate for the zero level-set function. The boundary conditions away from the interface make no significant difference in terms of interface location and property determination. For the sake of simplicity, the boundary condition is applied along the wall.

At the top of the computational domain ($y = Y$),

$$\frac{\partial u}{\partial y} = 0 \quad \frac{\partial v}{\partial y} = 0 \quad \frac{\partial \phi}{\partial y} = 0 \quad T = T_l \quad (54)$$

At the planes of symmetry ($r = 0, R$),

$$u = 0 \quad \frac{\partial v}{\partial r} = 0 \quad \frac{\partial \phi}{\partial r} = 0 \quad \frac{\partial T}{\partial r} = 0 \quad (55)$$

3.2. Initial Conditions

Initially, the fluid velocity is set to be zero. The temperature profile is taken to be linear in the natural-convection thermal boundary layer, and its thickness, δ_T , is given by Kays and Crawford [23]:

$$\delta_T = 7.14 \left(\frac{\nu_l \alpha_l}{g \beta_T \Delta T} \right)^{1/3} \quad (56)$$

4. NUMERICAL EXPERIMENTS

4.1. Bubbles Rising in a Liquid

To validate this moving-mesh method coupled with level-set functions, we simulate bubbles rising in a quiescent liquid and compare the results to those given

by Ryskin and Leal [24] and by Son [25]. Three cases with the same dimensionless parameters as described by Son [25] are studied in this present work. The computational domain is moved at the same velocity as the bubble rising velocity in order to prevent the bubble from moving out of the domain. The bubble radius R and the characteristic velocity \sqrt{gR} are used to scale the equations. The ratios of vapor and liquid properties are $\rho_v/\rho_l = 10^{-3}$ and $\mu_v/\mu_l = 10^{-2}$. Other dimensionless parameters are defined as

$$N_f = \frac{\rho_l R \sqrt{gR}}{\mu_l} \quad E_0 = \frac{\rho_l g R^2}{\sigma} \quad (57)$$

The computational domain is $6.4R \times 12.8R$. The boundary conditions are

$$\begin{aligned} &\text{At } (y = 0), \\ &u = 0 \quad \frac{\partial v}{\partial y} = 0 \quad \frac{\partial \phi}{\partial y} = 0 \quad p = 0 \end{aligned} \quad (58)$$

$$\begin{aligned} &\text{At } (y = Y), \\ &u = 0 \quad v = 0 \quad \frac{\partial \phi}{\partial y} = 0 \quad \frac{\partial p}{\partial y} = 0 \end{aligned} \quad (59)$$

$$\begin{aligned} &\text{At the planes of symmetry } (r = 0, R), \\ &u = 0 \quad \frac{\partial v}{\partial r} = 0 \quad \frac{\partial \phi}{\partial r} = 0 \quad \frac{\partial p}{\partial r} = 0 \end{aligned} \quad (60)$$

The computational results are shown in Table 2. The computation is carried out on 64×128 and 96×192 nonuniform grids until the terminal velocity is reached; the bubble rising velocities from 96×192 nonuniform grids are shown in Figure 2. The relative differences in all cases are less than 4.1%.

In this test problem, the monitor function is taken to be $w = \sqrt{1 + (\phi + 0.5h)^{-1}}$. The bubble shapes and grid distributions at steady state for the three cases are shown in Figure 3. The method can capture the interface deformation and keep the interface smooth, as depicted in Figure 3.

Table 2. Comparison of the dimensionless terminal velocities of rising bubbles

Case	N_f	E_0	Ryskin and Leal [24]	Present study		Relative difference (%)	
				64×128	96×192	64×128	96×192
(a)	1.34	14.3	0.374	0.378	0.367	1.1	1.8
(b)	5.87	4.13	0.852	0.824	0.833	3.3	2.2
(c)	27.6	0.61	1.814	1.740	1.766	4.1	2.6

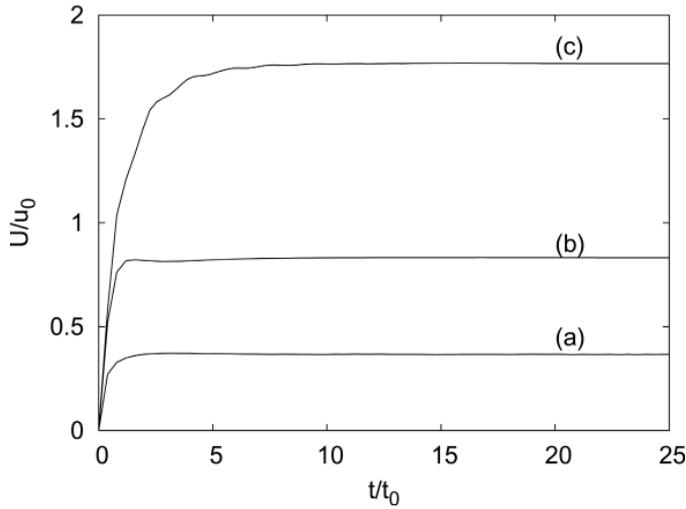


Figure 2. Bubble velocities for three cases: (a) $N_f = 1.34$, $E_o = 14.3$; (b) $N_f = 5.87$, $E_o = 4.13$, (c) $N_f = 27.6$, $E_o = 0.61$.

4.2. Spherically Growing Bubbles in a Superheated Liquid

This phase-change problem with an analytical solution was described by Son [26] and is used to test the capability of our method to handle heat transfer processes. A bubble embryo in an initially uniformly superheated liquid will keep growing spherically under zero gravity conditions and without any other external disturbance. In this case, the vapor is at the saturation temperature and its velocity is zero. The dimensionless governing equations for temperature and velocity on the liquid side can be written as

$$\frac{\partial \theta}{\partial t} + u \frac{\partial \theta}{\partial r} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(\frac{1}{\text{Pe}} r^2 \frac{\partial \theta}{\partial r} \right) \quad (61)$$

$$\frac{\partial}{\partial t} r^2 u = 0 \quad (62)$$

dimensionless parameters are defined as

$$\text{Pe} = \frac{\rho_l c_{pl} u_0 l_0}{k_l} \quad \text{Ja} = \frac{c_{pl} \Delta T}{h_{fg}} \quad \text{Re} = \frac{\rho_l u_0 l_0}{\mu_l} \quad \text{We} = \frac{\rho_l u_0^2 l_0}{\sigma}$$

The boundary conditions at $r = R$ are derived from the conditions of the mass continuity and energy balance at the interface:

$$\theta = 0 \quad \frac{dR}{dt} = \frac{\rho_l}{\rho_v} \left(\frac{dR}{dt} - u \right) = \frac{\rho_l}{\rho_v} \frac{\text{Ja}_{\text{sup}}}{\text{Pe}} \frac{\partial \theta}{\partial r} \quad (63)$$

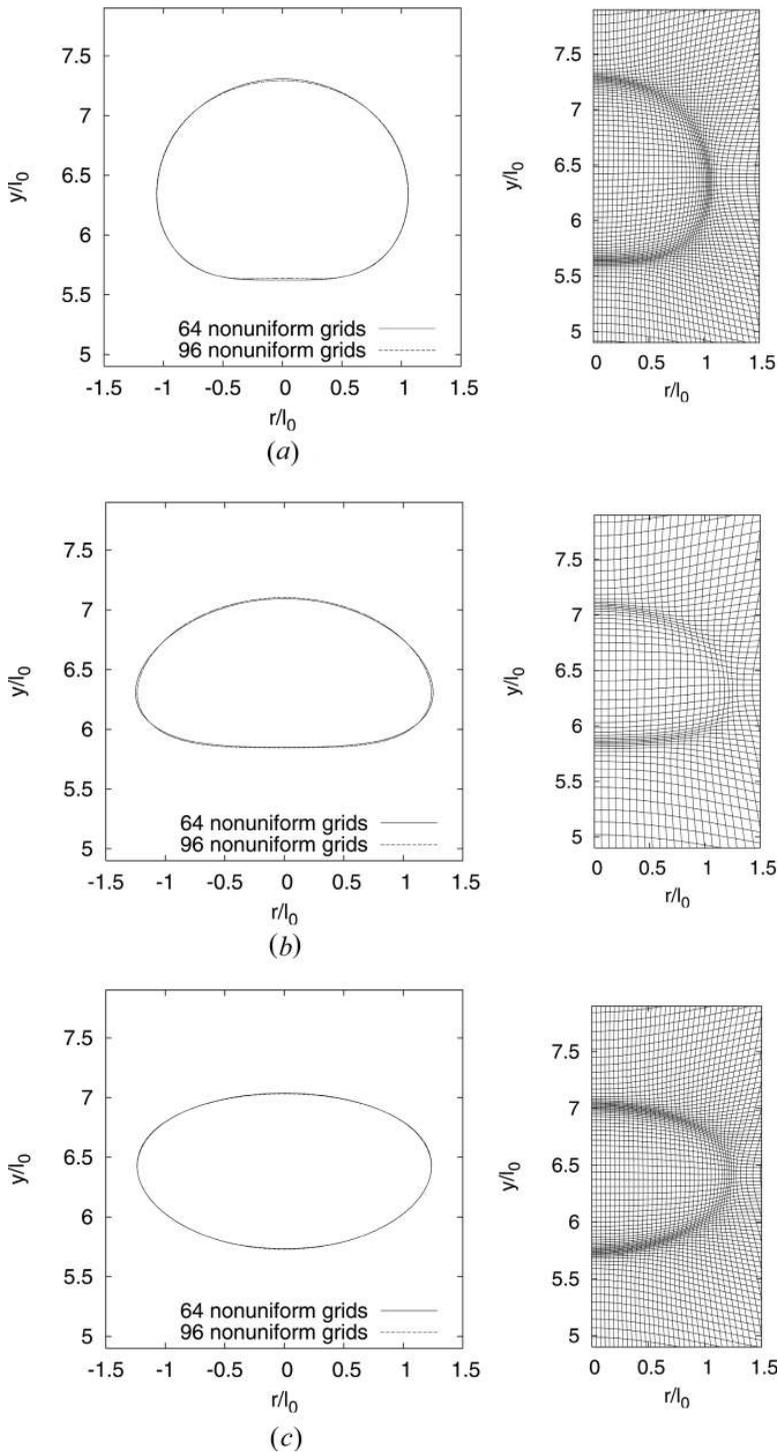


Figure 3. Bubble shapes and grid distribution at the terminal velocities for three cases.

Using a similarity variable $\zeta = r/R(t)$, we have an analytical solution to the spherical bubble growth,

$$R = \sqrt{2 \frac{\rho_l \text{Ja}_{\text{sup}}}{\rho_v \text{Pe}} \theta'_1 t} \quad (64)$$

where θ'_1 is obtained from the following equation:

$$\int_1^\infty \frac{\theta_1}{\zeta^2} \exp \left\{ \frac{\rho_l \text{Ja}_{\text{sup}} \theta'_1}{\rho_v} \left[\frac{1-\zeta^2}{2} + \left(1 - \frac{\rho_v}{\rho_l}\right) \left(1 - \frac{1}{\zeta}\right) \right] \right\} d\zeta = 1 \quad (65)$$

In this study, the dimensionless parameters using the properties of saturated water at 1.013×10^5 Pa are given as

$$\frac{\rho_l}{\rho_v} = 1,603 \quad \frac{\mu_l}{\mu_v} = 23.5 \quad \text{Re} = 118.5 \quad \text{We} = 0.04 \quad \text{Pe} = 207.4$$

$$\text{Ja}_{\text{sup}} = 1.12 \times 10^{-2} \text{ for } \Delta T_{\text{sup}} = 6^\circ\text{C} \quad \text{Ja}_{\text{sup}} = 1.50 \times 10^{-2} \text{ for } \Delta T_{\text{sup}} = 8^\circ\text{C}$$

The numerical computation is conducted on a quarter of the whole domain. The characteristic scales of $l_0 = 0.5$ mm and $u_0 = 0.07$ m/s are used to nondimensionalize the equations. The numerical computation of bubble radii as a function of time are compared with the exact solution in Figure 4. It shows that in both cases the uniform grids are not fine enough to resolve the bubble growth rates accurately. Except at the initial stage of the computation, the results show good agreement between the analytical growth rates and the numerical growth rates from nonuniform grids with the same number of mesh points as uniform grids. The advantage

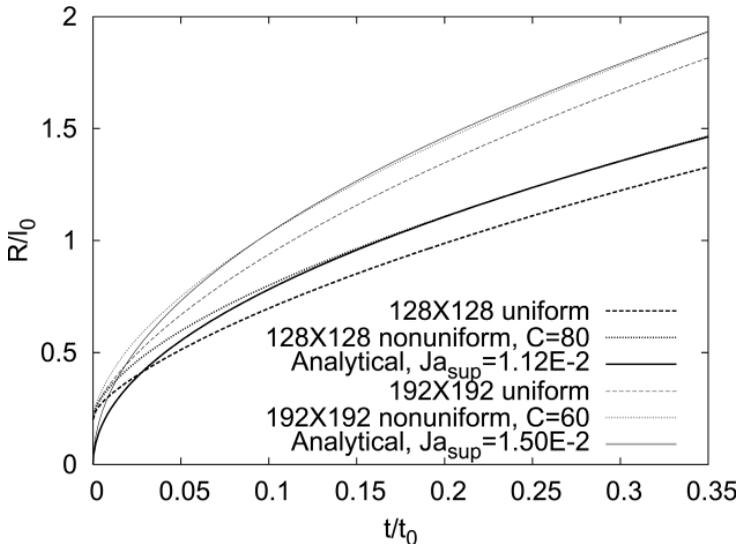


Figure 4. Comparison of the simulated bubble growth with the analytical solutions.

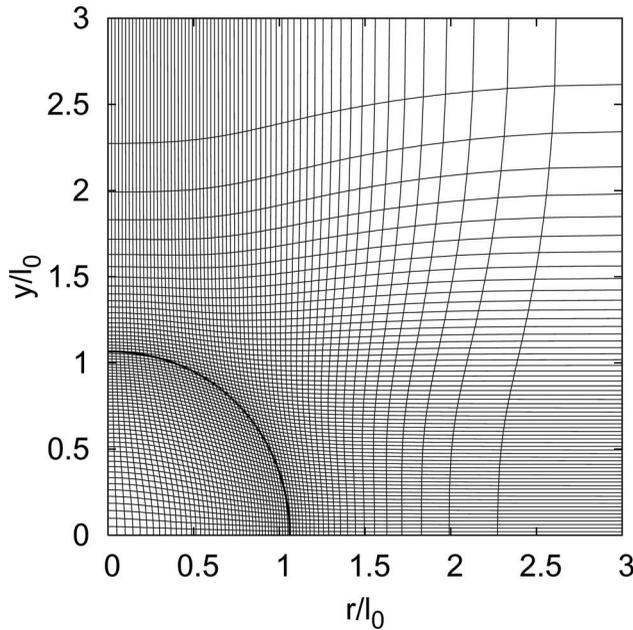


Figure 5. Grid distribution: $Ja_{\text{sup}} = 1.12 \times 10^{-2}$ at $t/t_0 = 0.19$.

of the current adaptive method is quite obvious in this case. In this test case, we use $w = C^{1-|\phi|/|\phi|_{\text{max}}}$ with $C = 80$ based on 128×128 mesh for $Ja_{\text{sup}} = 1.12 \times 10^{-2}$ and 192×192 nonuniform grids with $C = 60$ for $Ja_{\text{sup}} = 1.50 \times 10^{-2}$. Figure 5 presents the grid distribution at $t/t_0 = 0.19$ for the case of $Ja_{\text{sup}} = 1.12 \times 10^{-2}$. Note that

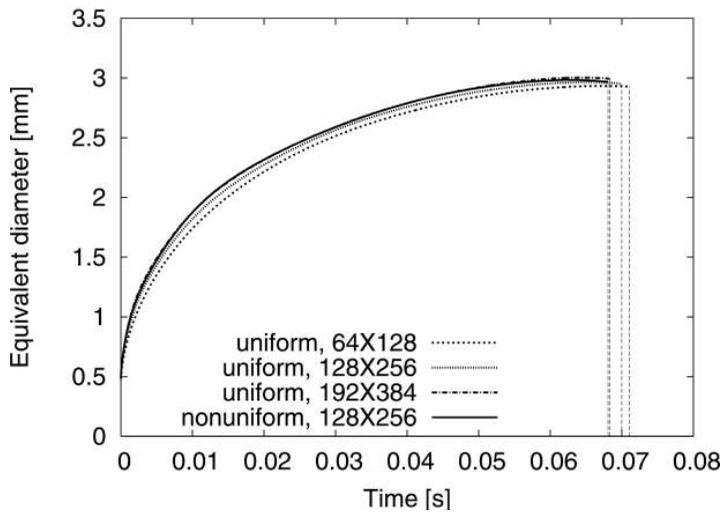


Figure 6. Comparison of growth rates for various grid sizes under conditions of wall superheat = 8°C , liquid subcooling = 1°C , contact angle = 54° , pressure = 1.013×10^5 Pa.

for the sake of clarity, only every alternate grid point is plotted. The interface is also shown in Figure 5.

4.3. Subcooled Nucleate Boiling

A numerical simulation of single-bubble dynamics during subcooled nucleate boiling for a wall superheat of 8°C , liquid subcooling of 1°C , contact angle of 54° , and system pressure of $1.013 \times 10^5 \text{ Pa}$ is conducted with different grid sizes. The growth rate of the bubble is adopted as the primary means of comparing the effect of different parameters. The actual volume of the bubble is first computed; thereafter it is converted into an equivalent diameter of a sphere.

Figure 6 shows the growth rate of the bubble with four different grid sizes. For all the grids, the bubble detaches from the wall at a size which is smaller than its maximum size. This bubble shrinkage near the point of departure is caused by the

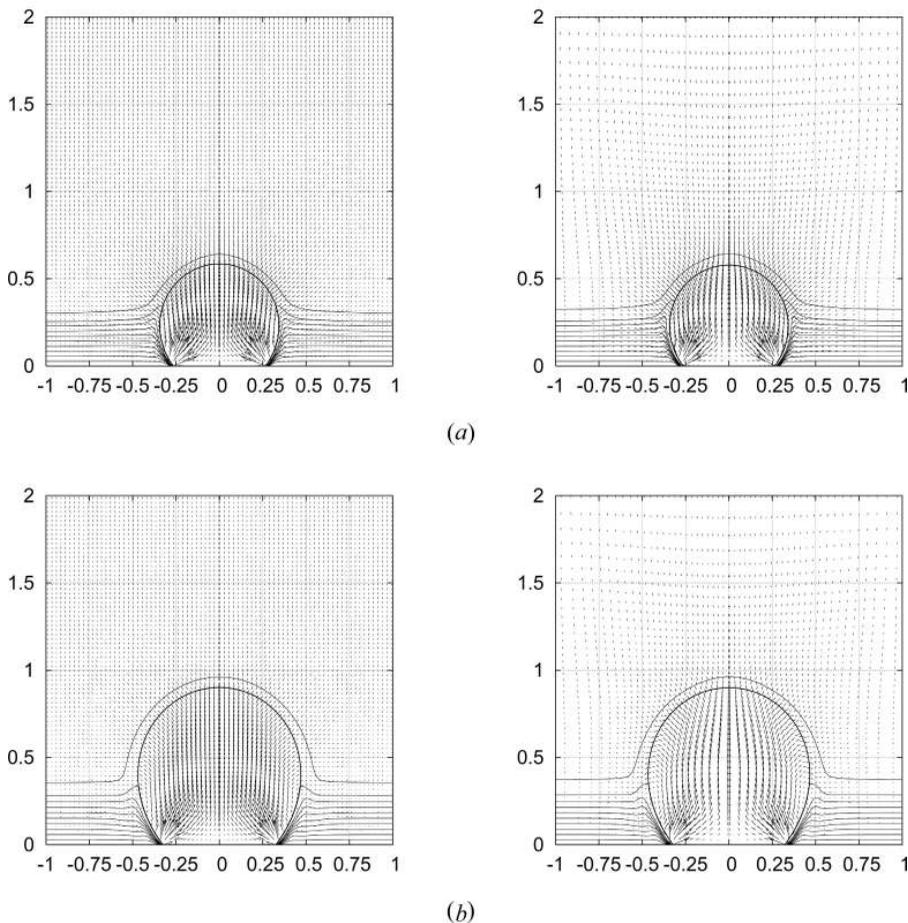


Figure 7. Comparison of temperature profiles and velocity vectors between 192×384 uniform (left) and 128×256 nonuniform grids (right): (a) $t = 0.008 \text{ s}$; (b) $t = 0.024 \text{ s}$.

negative total heat transfer to the bubble; i.e., condensation heat transfer is larger than the sum of evaporation heat transfer around the bubble and from the micro-layer, as described by Singh [27].

Figures 7 and 8 show the comparison between 192×384 uniform and 128×256 nonuniform grids of the calculated flow field and temperature field during the bubble-growth process. They are almost identical. From the numerical simulation, it can be seen that the growing bubble pushes out the liquid. The location where the vapor-liquid interface contacts the wall is observed to move outward and then inward as the bubble grows and departs. The highest heat transfer rate occurs at the base of the bubble. The nonuniform velocity inside the bubble results in a noticeable vortex. The isotherm that terminates at the bubble interface represents the saturation temperature. It distinguishes the area where evaporation takes

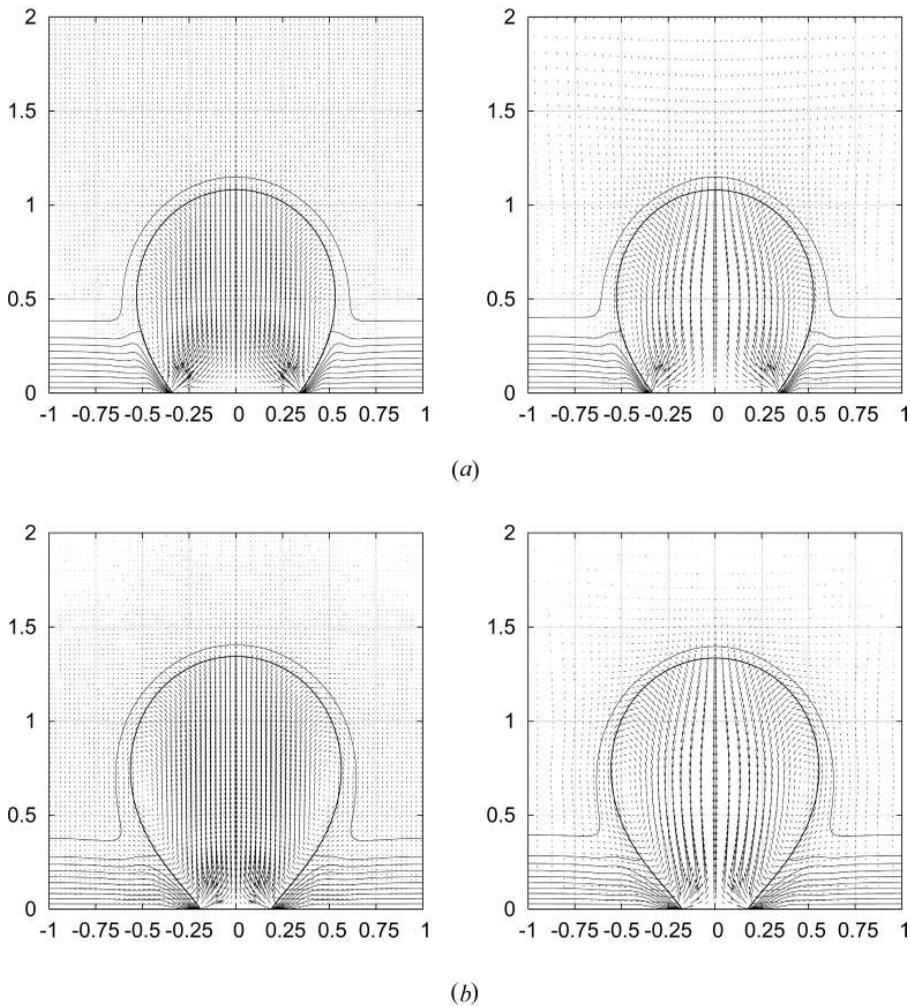


Figure 8. Comparison of temperature profiles and velocity vectors between 192×384 uniform (left) and 128×256 nonuniform grids (right): (a) $t = 0.040$ s; (b) $t = 0.064$ s.

place from the area where condensation occurs. We can observe that the condensation area dominates the evaporation area during most of the growth period. The adaptive grid distribution near the interface is shown in Figure 9. Only every alternate grid point is plotted, for clarity.

Due to the explicit treatment of the advection terms, the time step is chosen to satisfy the CFL condition after the grid size is determined. The time step Δt is chosen

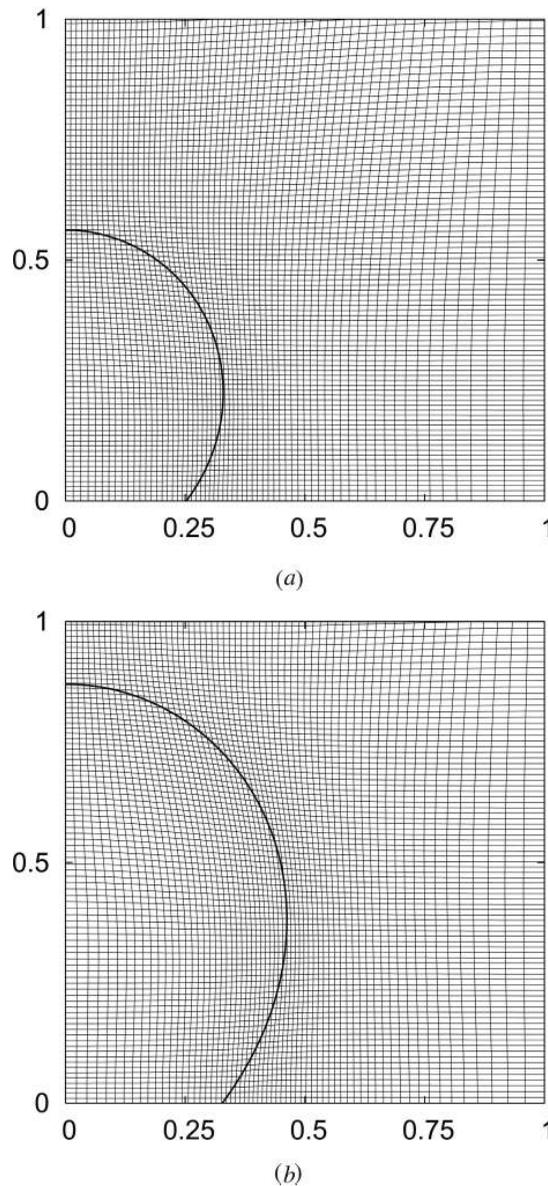


Figure 9. Evolving grid distribution: (a) $t = 0.008$ s; (b) $t = 0.024$ s.

by decreasing it until no significant difference can be found in subsequent results. Usually, dimensionless Δt is less than 2.0×10^{-4} .

These results show that the current adaptive method with a 128×256 nonuniform mesh can achieve similar results as methods based on a 192×384 uniform mesh. The results of this test case do not imply that in other situations the current method can produce the same results as a method based on 1.5 times uniform grids. Another case with high subcooling is studied next to evaluate the effectiveness of this new method.

Figure 10 shows the growth rate of the bubble for the following conditions: wall superheat = 8°C , liquid subcooling = 7°C , contact angle = 54° , pressure = 1.013×10^5 Pa. In this study, the same thickness is used to bridge the property difference across the bubble interface with $w = C^{1-|\phi|/|\phi|_{\max}}$ and $C = 10, 20$ is adopted to distribute grids in nonuniform cases. Computations are performed for both uniform and nonuniform grids. In all cases, the bubble grows to a maximum size and then begins to shrink. This shrinkage is due to the large condensation heat transfer that occurs at the top of the bubble. As a result, the bubble never achieves the bubble departure size. Eventually, a balance is achieved in evaporation and condensation rate, and the bubble remains attached to the heating surface. Figure 10 shows that as the grid size decreases, the temperature gradients near the interface are more accurately captured, especially in the regions where condensation occurs. As a result, finer meshes give rise to a smaller bubble. After 0.15 s, the differences between bubble diameters calculated with different grids, except for the case with 64×128 uniform grids, become insignificant. This case shows that the current method is quite effective for numerical simulations with higher subcooling. The computation results from nonuniform grids with $C = 10$ and 20 are similar.

In Figure 11, the evaporation heat flux is indicated by the concentration of isotherms at the base of the bubble. Vapor flowing upward from the base

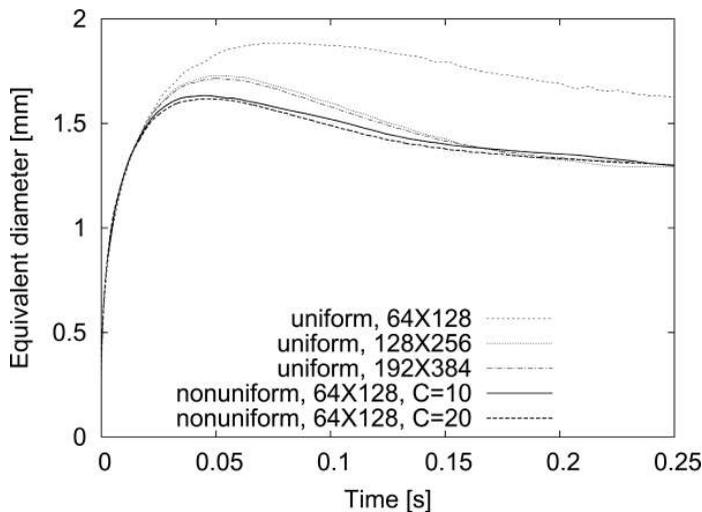


Figure 10. Comparison of growth rates for various grid sizes under conditions of wall superheat = 8°C , liquid subcooling = 7°C , contact angle = 54° , pressure = 1.013×10^5 Pa.

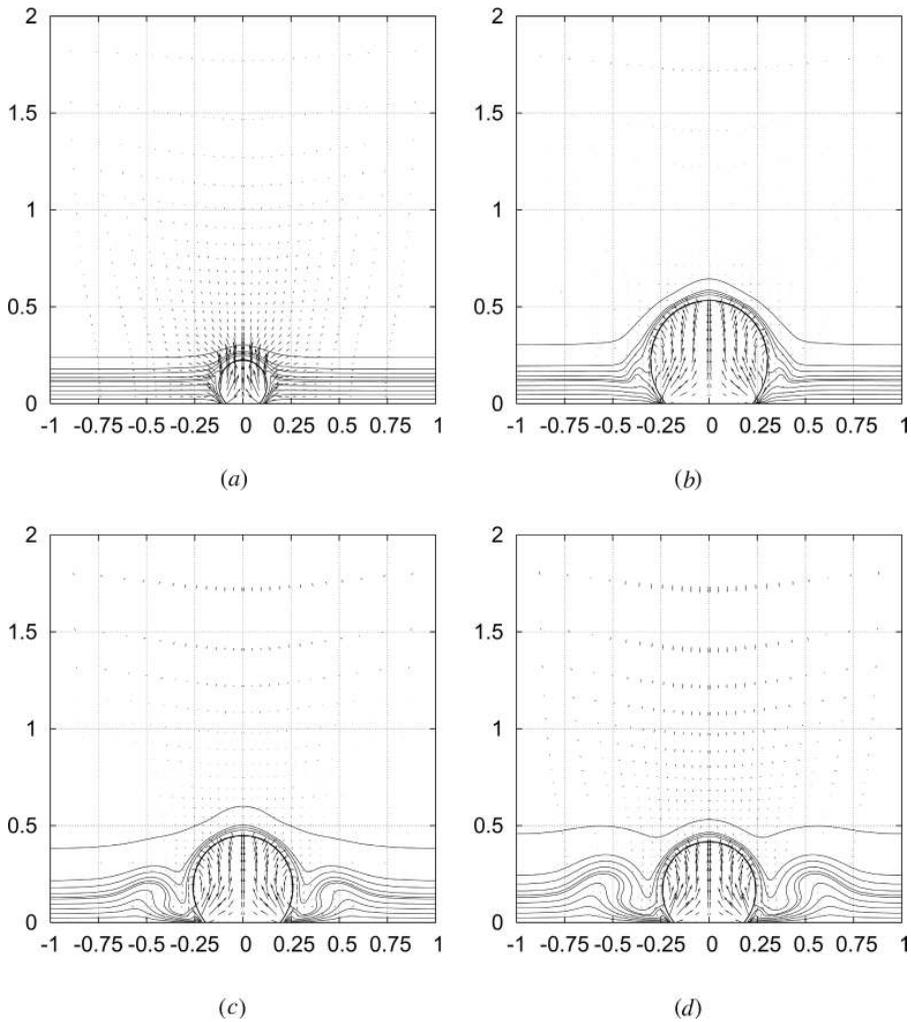


Figure 11. Temperature profiles and velocity vectors: (a) $t = 0.0002$ s; (b) $t = 0.0400$ s; (c) $t = 0.1438$ s; (d) $t = 0.2493$ s.

condenses over most of the interface. The liquid that has just condensed around the interface flows downward toward the wall and thins down the thermal layer near the base of the bubble. This in turn leads to saddle points in the isotherms. The isotherm distribution is quite different from the case for low subcooling as shown in Figures 7 and 8.

Figure 12a shows the features of the grid distribution. The difference between the neighboring grids around the interface is small and smooth, which is important in mesh generation. Meanwhile, high adaptivity is maintained, which can be seen from the ratio of the maximum to the minimum area of cells shown in Figure 12b.

The other topics that need to be addressed are the memory usage and computational time of the present method. To make discretization handy, the coordinates of

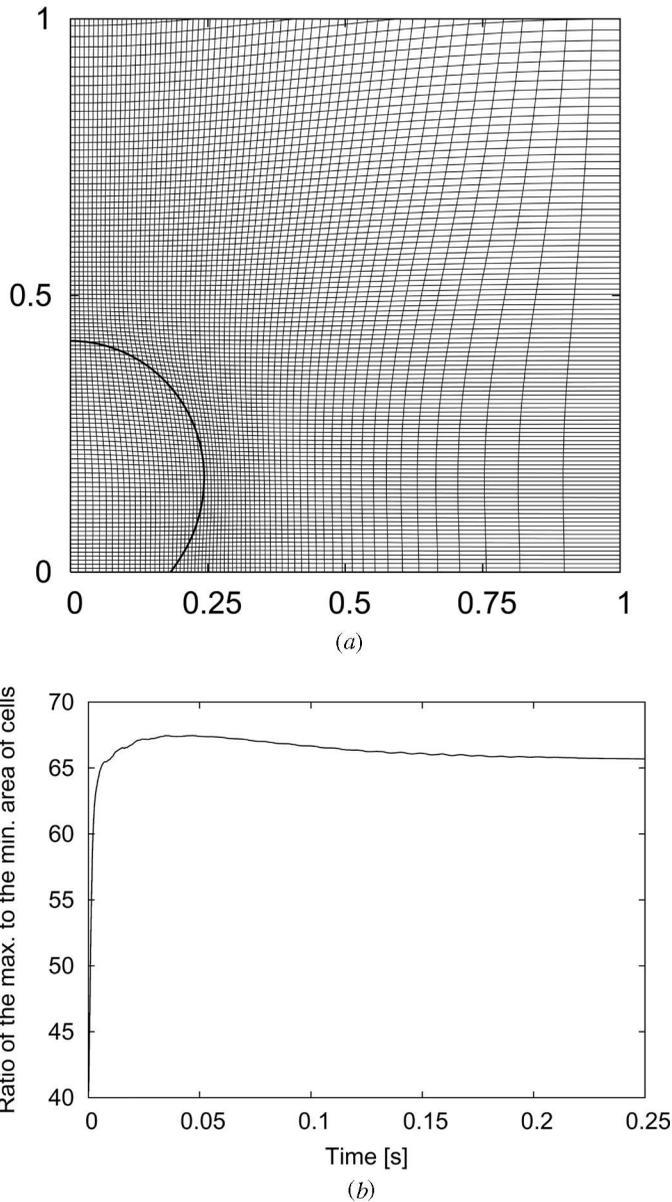


Figure 12. (a) Grid distribution at $t = 0.2493$ s. (b) Ratio of maximum to minimum areas of grids.

the centers of each cell and its corners are kept in memory. In addition to this geometry information, the Cartesian velocity components at the centers of $d\xi_e$ and $d\eta_e$ are also stored. The memory usage is $O(N^2)$ in the current two-dimensional code, where N is the number of mesh points along one of the spatial directions. In terms of the number of variables stored in memory, for a fixed N the present method is approximately 2.5 times more expensive than that using uniform grids. However,

Table 3. Comparison of CPU times for various grid sizes

Grid type	Uniform grids			Nonuniform grids	
	64 × 128	128 × 256	192 × 384	64 × 128, C = 10	64 × 128, C = 20
Δt	4.50E-4	2.25E-4	1.50E-4	1.00E-4	1.00E-4
CPU time (s)	3.39E+3	4.23E+4	1.51E+5	3.62E+4	4.45E+4

we have significant reduction in computational time. Table 3 shows the comparison of CPU times for various grid sizes. All the computations were carried out on a 3.20-GHz PC computer using double precision. For uniform grids, the CPU times will be 12.5 and 44.5 times more expensive if the grid number is doubled and tripled, respectively. For nonuniform grids, C can affect the computational time. The CPU times in both cases for nonuniform grids with $C = 10, 20$ have the same magnitude as that for the 128×256 uniform grids. Although a more restricted time step is used, the total CPU times for $C = 10, 20$ are 24% and 30% of the CPU time for 192×384 uniform grids, respectively.

To simulate nucleate boiling under microgravity conditions, we can see that the characteristic length scale l_0 based on the balance between gravity and surface tension forces as given in Eq. (16) will be $g^{-0.5}$ times the corresponding value under Earth-normal gravity conditions. This length will result in a very coarse mesh distribution for a reasonable CPU time when a uniform grid is used. In turn, this may lead to inaccuracy in results. Thus, in order to accurately capture the temperature gradient around the bubble interface with a uniform grid, a large number of grids will be required. Alternatively, the current adaptive mesh can be used very advantageously to cluster the nodes around the bubble interface. As a result, significant savings in computational time can be achieved.

5. CONCLUSIONS

A numerical procedure coupling the level-set function with the moving-mesh method has been developed that enables study of nucleate pool boiling under strong temperature gradients near the interface.

The test problems show that the current method can be more accurate than its counterpart with uniform grid structure. The simulation of bubble dynamics under subcooled boiling conditions shows that the new adaptive method can achieve the same accuracy as the methods based on uniform grids with many more mesh points.

REFERENCES

1. R. C. Lee and J. E. Nydahl, Numerical Calculation of Bubble Growth in Nucleate Boiling from Inception through Departure, *ASME J. Heat Transfer*, vol. 111, pp. 474–479, 1989.
2. R. Mei, W. C. Chen, and J. F. Klausner, Vapor Bubble Growth in Heterogeneous Boiling—1. Formulation, *Int. J. Heat Mass Transfer*, vol. 38, pp. 909–919, 1995.
3. S. W. J. Welch, Direct Simulation of Vapor Bubble Growth, *Int. J. Heat Mass Transfer*, vol. 41, pp. 1655–1666, 1998.

4. H. Y. Yoon, S. Koshizuka, and Y. Oka, Direct Calculation of Bubble Growth, Departure, and Rise in Nucleate Pool Boiling, *Int. J. Multiphase Flow*, vol. 27, pp. 277–298, 2001.
5. G. Son, V. K. Dhir, and N. Ramanujapu, Dynamics and Heat Transfer Associated with a Single Bubble during Nucleate Boiling on a Horizontal Surface, *J. Heat Transfer*, vol. 121, pp. 623–631, 1999.
6. M. Sussman, P. Smereka, and S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, *J. Comput. Phys.*, vol. 114, pp. 146–159, 1994.
7. S. Osher and J. A. Sethian, Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.
8. M. J. Berger and J. Olinger, Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, *J. Comput. Phys.*, vol. 53, pp. 484–512, 1984.
9. M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, An Adaptive Level Set Approach for Incompressible Two Phase Flows, *J. Comput. Phys.*, vol. 148, pp. 81–124, 1999.
10. J. Qian, Approximations for Viscosity Solutions of Hamilton-Jacobi Equations with Locally Varying Time and Space Grids, *SIAM J. Numer. Anal.*, vol. 43, pp. 2371–2401, 2006.
11. A. M. Khokhlov, Fully Threaded Tree Algorithms for Adaptive Refinement Fluid Dynamics Simulations, *J. Comput. Phys.*, vol. 143, pp. 519–543, 1998.
12. J. Strain, Tree Methods for Moving Interfaces, *J. Comput. Phys.*, vol. 151, pp. 616–648, 1999.
13. T. C. Cecil, S. Osher, and J. Qian, Simplex Free Adaptive Tree Fast Sweeping and Evolution Methods for Solving Level Set Equations in Arbitrary Dimension, *J. Comput. Phys.*, vol. 213, pp. 458–473, 2006.
14. H. Z. Tang, T. Tang, and P. Zhang, An Adaptive Mesh Redistribution Method for Nonlinear Hamilton-Jacobi Equations in Two- and Three-Dimensions, *J. Comput. Phys.*, vol. 188, pp. 543–572, 2003.
15. H. Ceniceros and T. Y. Hou, An Efficient Dynamically Adaptive Mesh for Potentially Singular Solutions, *J. Comput. Phys.*, vol. 172, pp. 609–639, 2001.
16. W. Huang and R. D. Russell, Moving Mesh Strategy Based on a Gradient Flow Equation for Two-Dimensional Problems, *SIAM J. Sci. Comput.*, vol. 20, pp. 998–1015, 1999.
17. K. C. Karki and S. V. Patankar, Calculation Procedure for Viscous Incompressible Flows in Complex Geometries, *Numer. Heat Transfer*, vol. 14, pp. 295–307, 1988.
18. W. Yue, C. Lin, and V. C. Patel, Numerical Simulation of Unsteady Multidimensional Free Surface Motions by Level Set Method, *Int. J. Numer. Meth. Fluids*, vol. 42, pp. 853–884, 2003.
19. G. Son and V. K. Dhir, Numerical Simulation of Melting of a Horizontal Substrate Placed beneath a Heavier Liquid, *Int. J. Heat Mass Transfer*, vol. 41, pp. 89–99, 1998.
20. J. H. Lay and V. K. Dhir, Shape of a Vapor Stem during Nucleate Boiling of Saturated Liquids, *ASME J. Heat Transfer*, vol. 117, pp. 394–401, 1995.
21. P. C. Wayner, Jr., Evaporation and Stress in the Contact Line Region, *Proc. Engineering Foundation Conf. on Pool and External Flow Boiling*, Santa Barbara, CA, 1992, pp. 251–256.
22. J. Qian and S. Leung, A Level Set Based Eulerian Method for Paraxial Multivalued Travel Times, *J. Comput. Phys.*, vol. 197, pp. 711–736, 2004.
23. W. M. Kays and M. E. Crawford, *Convective Heat and Mass Transfer*, p. 328, McGraw-Hill, New York, 1980.
24. G. Ryskin and L. G. Leal, Numerical Solution of Free-Boundary Problems in Fluid Mechanics, Part I, *J. Fluid Mech.*, vol. 148, pp. 1–17, 1984.

25. G. Son, Efficient Implementation of a Coupled Level-Set and Volume-of-Fluid Method for Three-Dimensional Incompressible Two-Phase Flows, *Numer. Heat Transfer B*, vol. 43, pp. 549–565, 2003.
26. G. Son, A Numerical Method for Bubble Motion with Phase Change, *Numer. Heat Transfer B*, vol. 39, pp. 509–523, 2001.
27. S. Singh, Effect of Gravity and Liquid Subcooling on Bubble Dynamics, Master's thesis, University of California, Los Angeles, 1999.