Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 202 (2005) 65–93

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A hybrid particle-mesh method for viscous, incompressible, multiphase flows

Jie Liu *, Seiichi Koshizuka, Yoshiaki Oka

Nuclear Engineering Research Laboratory, Graduate School of Engineering, The University of Tokyo, 2-22 Shirane, Shirakata, Tokai-mura, Naka-gun, Ibaraki 319-1188, Japan

Received 8 September 2003; received in revised form 7 June 2004; accepted 2 July 2004
Available online 13 August 2004

Abstract

A hybrid method to simulate unsteady multiphase flows in which a sharp interface separates incompressible fluids of different density and viscosity is described. One phase is represented by moving particles and the other phase is defined on stationary mesh. The flow field is discretized by a conservative finite volume approximation on the stationary mesh, and the interface is automatically captured by the distribution of particles moving through the stationary mesh. The effects of surface tension and wall adhesion are evaluated by the continuum surface force model. The different phases are treated as one fluid with variable material properties. Advection of fluid properties such as density and viscosity is done by following the motion of the particles. The method simplifies the calculation of interface interaction, enables accurate modeling of two- and three-dimensional multiphase flows and does not impose any modeling restrictions on the dynamic evolutions of fluid interfaces having surface tension. Several two-dimensional numerical simulations are given to illustrate the efficiency of the hybrid method.

MSC: 65E05; 76T99
Keywords: Multiphase flow; Hybrid method; Particle method; Mesh method; Interface tracking; Finite volume method; Surface tension; Wall adhesion; Contact angle

1. Introduction

Accurate simulation of multiphase flows with a sharp interface has considerable difficulty in numerical methods. Problems with a moving interface are important in many technological applications in which

* Corresponding author. Tel.: +81-29-287-8433; fax: +81-29-287-8488.
E-mail address: liuj@utnl.jp (J. Liu).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.07.002

moving interfaces play a dominant role. Capillarity phenomena, wetting effect, deformation of droplet or bubble, motion of boundaries between immiscible fluids are some of well-known examples in engineering and science. In order to describe quantitatively such problems, the understanding of physical processes taking place on the interfaces is necessary. In particular, successful simulation of the moving interfaces depends on the numerical method in which the corresponding interfaces can be captured effectively.

The most popular approach to compute multiphase flows is to capture the front directly on a regular, stationary mesh. For example, marker-and-cell (MAC) [1] method uses marker particles to identify each fluid. The volume-of-fluid (VOF) [2] method uses a marker function to identify the interface. The main difficulty in using these methods is the maintenance of a sharp boundary between two phases and the computation of surface tension. Further developments to the methods that capture fluid interfaces on a fixed grid include the CIP [3] method and the phase field method [4], etc.

Adaptive (moving) grid methods alter the computational grid so that the interface always coincides with one of the grid lines. The interface is then a well-defined, continuous curve. The information regarding the location and curvature of the interface is readily available. The review of techniques for the construction of boundary conforming coordinates is referred to [5]. The main advantage of this approach is that it is possible to maintain sharp resolution of the interface, while the main disadvantage is the difficulty in adjusting the grids to follow the highly deformed interfaces.

The front tracking is another method where the interface itself is descried by additional computational elements, and the fixed grid is kept unchanged. The major difficulty of direct front tracking is the question of how the interface grid interacts with the stationary grid, and vice versa. It is also necessary to restructure the interface grid dynamically as the calculations proceed. Computational points must be added in regions where the interface grid points become scattered, and eliminated from regions of concentration. Another problem in front tracking results from the interaction of a front with another front. Double interfaces have to be merged into one interface or eliminated. A merging algorithm is usually used. A development to this method made by Tryggvason et al. [6] is a hybrid between front capturing and front tracking technique. A stationary regular grid is used for the fluid flow, but the interface is tracked by a separated grid of lower dimension. However, unlike front tracking methods where each phase is treated separately, all phases are treated together by solving a single set of governing equations for the whole field. The explicit tracking of the interface reduces errors associated with the surface tension computations, and its flexibility makes it applicable to problems where complex interface physics must be accounted for. The specific difficulties with front tracking are discussed by Glimm et al. [7] and two computational algorithms to handle changes in the interface topology for dynamic interface tracking in three dimensions have been described [8].

From the point of view of numerical algorithms, the methods mentioned above are classified as Eulerian method. Lagrangian method is another category, which is suited for moving boundary problems because it permits interfaces to be specifically delineated and precisely followed and it allows interface boundary conditions to be easily applied [5]. The two main problems of the Lagrangian methods are mesh tangling and numerical inaccuracy due to highly irregular meshes. Particle method is another Lagrangian description of flows in which particles are explicitly associated with different materials and thus the interfaces can be easily followed. The well-known example is the particle-in-cell (PIC) [9] algorithm. The basic idea of PIC is to divide the region of interest into regular Eulerian cells for purposes of computing field variables such as pressure and fluid velocity, and to simulate the material transport from cell to cell in a Lagrangian fashion in the form of discrete simulation particles. The attractive features of the method are no mass diffusion and the ability of treating large distortions of fluids, large slippages, and colliding interfaces. The problems consist of a large numerical diffusion caused by transferring the momentum from the grid to the particle and back, numerical noise caused by the use a finite numerical of fluid particles, and limited spatial resolution because of a fixed uniform Eulerian grid being used.

The Lagrangian particle method called smooth particle hydrodynamics (SPH) [10] abandons grid completely. The approximation of spatial derivatives is derived from the interpolation among particles. Moving particle semi-implicit (MPS) [11–19] method is another meshless method where the fluids are represented by a finite number of moving particles. Governing equations are discretized based on particle interaction models representing gradient, Laplacian, and free surface. Grids are not necessary at all. Thus, the method is free from the grid distortion. A semi-implicit algorithm is employed to analyze incompressible flows. The numerical diffusion derived from the convection terms does not arise because of fully Lagrangian description. The MPS method was applied to multi-phase and multi-fluid thermal-hydraulic problems such as free surface flows [12], droplet breakup [13], single bubble rise and nucleate boiling [14,15], jet injection [16], fragmentation [17,18], large deformation and fracture of solid [19], etc.

For the accurate prediction of multiphase flows, the numerical method must possess the ability of evaluation surface tension forces at interface. In the continuum surface force (CSF) model [20], a surface force is modeled with a localized volume force. Instead of a surface tensile force or a surface pressure boundary condition applied at a discontinuity, a volume force acts on fluid interfaces within finite thickness continuously. Surface tension modeled with the continuum method eliminates the need for interface reconstruction, and can be easily calculated by applying an extra body force in the momentum equation.

In this paper, we present a hybrid method for incompressible, viscous, multiphase flows by coupling MPS method with mesh method. One fluid (or phase) is represented by moving particles, while another fluid (or phase) is defined on the mesh. The interfaces are automatically determined by the distribution of particles without specific front tracking algorithms, and no numerical diffusion arises. The original underlying mesh is retained through the simulations and no restructuring is needed. The CSF model will be used to evaluate the surface tension force, and all phases are treated together by solving a single set of governing equations for the whole flow field. The primary advantage of this approach is that the interfaces carry the jump of properties and that, at each time step, the property field is easily calculated from the information carried by the distribution of particles. The interactions of the interfaces automatically taken care of in the present method is considered one of the major differences from the previous front-tracking methods.

2. Numerical method

2.1. Description of multiphase flow by particle and mesh

In the present method, the multiphase flow region of interest is divided into regular Eulerian mesh for purposes of computing field variables such as pressure and fluid velocity. Some part of the region is occupied by a finite ensemble of particles, which represents one phase (liquid phase, for example, in gas-liquid two-phase flow). The rest part of the region not occupied by the particles is another phase (gas phase, for example). The general setup of particle and mesh is sketched in Fig. 1.

The use of particles is the same way as that in MPS method. Each particle has a set of attributes, such as mass and position. Since the particles are explicitly associated with one phase, the interfaces between different phases can be easily followed by the distribution of particles. The whole multiphase flow system will be calculated on the mesh, and the advection of the density and the viscosity fields is achieved by following the motion of particles. The velocities calculated on the mesh are used to advance the particles. The attractive features of the method are that the mass conservation is preserved, and the interaction of interfaces is possible. Furthermore, the continuous change of physical quantities to eliminate the numerical noise is ensured. The transfer of information between the particles and the grid will be given in detail in the following sessions.

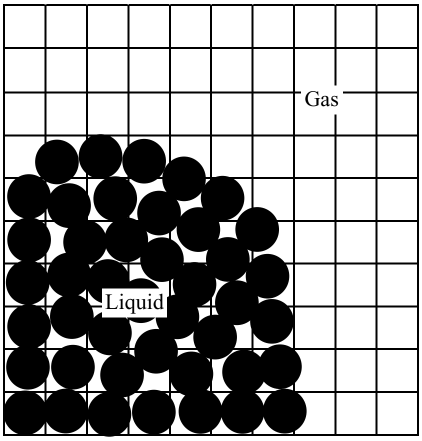

Fig. 1. The setup of mesh and particle.

2.2. Governing equations

The equations governing the motion of unsteady, viscous, incompressible multiphase flows are the laws of conservation of mass

$$\oint_S \vec{v}\cdot\vec{n}\ \mathrm{d}S = 0, \tag{1}$$

and momentum

$$\frac{\partial}{\partial t}\int_V \rho\,\vec{v}\ \mathrm{d}V + \oint_S \rho\,\vec{v}(\vec{v}\cdot\vec{n})\mathrm{d}S = -\oint_S p\cdot\bar{\bar{I}}\cdot\vec{n}\ \mathrm{d}S - \oint_S \bar{\bar{\tau}}\cdot\vec{n}\ \mathrm{d}S + \int_V \vec{f}_V\,\mathrm{d}V, \tag{2}$$

where t represents time, $\vec{v}$, p, and ρ are velocity vector, the pressure and the density of the fluid, respectively; $\bar{\bar{I}}$ is the identity matrix, V is the volume occupied by the fluid, S is the surface area formed by the boundary of this volume, and $\vec{n}$ is the outward pointing unit vector, normal to S. $\vec{f}_V$ is the sum of forces acting on the fluid, $\bar{\bar{\tau}}$ is the stress tensor, which for a Newtonian fluid in two dimensions is

$$\bar{\bar{\tau}} = \begin{bmatrix} -2\mu\frac{\partial u}{\partial x} & -\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \\ -\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) & -2\mu\frac{\partial v}{\partial y} \end{bmatrix}, \tag{3}$$

where μ is the dynamic viscosity, and u and v are the horizontal and vertical components of the velocity, respectively.

Although $\vec{f}_V$ has the form of a volume force, it also includes the effect of the surface tension in the present method, which will be discussed in the following session.

2.3. Surface tension model

In the present method, the interfaces are represented by the interfacial particles. The interfacial particles can be easily determined by the particle number density, which was defined originally in the MPS method [11]. The particle number density n of particle i is defined as

$$n_i = \sum_{j \neq i} w\left(|\vec{r}_j - \vec{r}_i|\right), \tag{4}$$

where $\vec{r}$ is the coordinate, and w is weight function defined by

$$w(r) = \begin{cases} \frac{r_{\mathrm{e}}}{r} - 1, & r \leqslant r_{\mathrm{e}}, \\ 0, & r > r_{\mathrm{e}}, \end{cases} \tag{5}$$

here r is the distance between two particles, r_{e} is radius of interaction area. The particle number density is proportional to the fluid density. The mass conservation law of incompressible flow requires that the fluid density should be constant. This is equivalent to the particle number density being constant, n^0. The method to maintain this constant value will be discussed in the following session. A particle whose particle number density satisfies

$$n_i < \beta n^0 \tag{6}$$

is regarded as the interface, where β is a parameter below 1.0. Unlike the front tracking method, in which the interface consists of connected marker points in two dimensions, it is not necessary to order these interfacial particles to calculate such quantities as surface tension. We will model the interfacial quantities by using the particle interactions with the kernel function, which are the same as those in MPS method [11].

The surface tension force is calculated on the particles that are regarded as interface. In the present method, surface tension is modeled as a volume force derived from the CSF model [20]. The surface force per unit interfacial area can be written as

$$\vec{f}_{\mathrm{sa}} = \sigma \kappa_i \hat{n}_i, \tag{7}$$

where σ is the surface tension coefficient, κ is the curvature of the interface and $\hat{n}$ is unit normal to the interface. The curvature is calculated form

$$\kappa = -(\nabla \cdot \hat{n}) \tag{8}$$

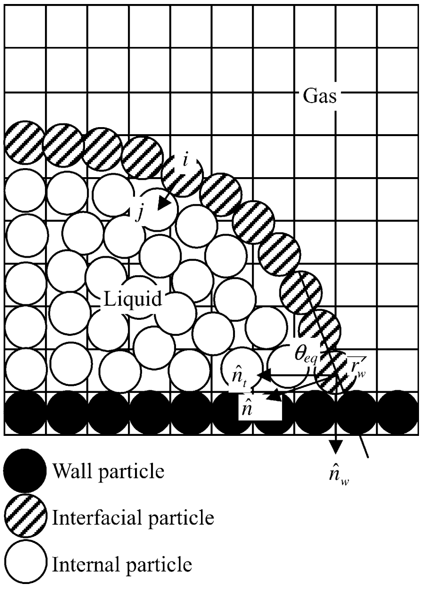

Fig. 2. Definition of vector normal and contact angle.

and the unit normal is derived from normal vector

$$\hat{n} = \frac{\vec{n}}{|\vec{n}|}. \tag{9}$$

The normal vector can be obtained by a gradient of particle number density at the interface. As shown in Fig. 2, a gradient vector between two particles i and j possessing n_i and n_j at coordinates $\vec{r}_i$ and $\vec{r}_j$ is simply defined by $(n_j - n_i)(\vec{r}_j - \vec{r}_i)/|\vec{r}_j - \vec{r}_i|^2$. The gradient vectors between particle i and its neighboring particles j are weighted with the kernel function and averaged to obtain a gradient vector at particle i:

$$\vec{n}_i = (\nabla n)_i = \frac{d}{n_i} \sum_{j \neq i} \left[\frac{n_j - n_i}{|\vec{r}_j - \vec{r}_i|^2} (\vec{r}_j - \vec{r}_i) w(|\vec{r}_j - \vec{r}_i|) \right], \tag{10}$$

where d is the number of space dimensions [11].

The divergence of unit normal vector is calculated by

$$(\nabla \cdot \hat{n})_i = \frac{d}{n_i} \sum_{j \neq i} \left[\frac{(\hat{n}_j - \hat{n}_i)(\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^2} w(|\vec{r}_j - \vec{r}_i|) \right], \tag{11}$$

where $\hat{n}$ is unit normal vector, n_i is particle number density. The values calculated by Eqs. (10) and (11) are first order accurate because of the asymmetric distribution of particles at the interface.

The surface force $\vec{f}_{\mathrm{sa}}$ per unit interfacial area acting on particle i can be transferred to volume force $\vec{f}_{\mathrm{sv}}$. If the average distance between particles is d, the volume force is estimated by $\vec{f}_{\mathrm{sv}} = \vec{f}_{\mathrm{sa}}/d$ in two dimensions because the area of a particle is d^2.

With the volume force, surface tension effects at interface are modeled as a body force included in the momentum equation (2), instead of a surface tensile force or a surface pressure boundary condition applied at a discontinuity.

2.4. Boundary conditions: wall adhesion

Wall adhesion is the surface force acting on fluid interfaces at points of contact with "walls", which are static, rigid boundaries. Wall adhesion forces are calculated in the same manner as volume forces due to surface tension are calculated, using Eq. (7) for $\vec{f}_{\mathrm{sa}}$, except that a boundary condition is applied to the interface unit normal $\hat{n}$ prior to evaluating Eq. (7). The condition is applied only to those particles near a rigid boundary, as shown in Fig. 2. The wall adhesion boundary condition becomes an expression for the unit interface normal $\hat{n}$ at points of contact $\vec{r}_{\mathrm{w}}$ along the wall [20]:

$$\hat{n} = \hat{n}_{\mathrm{w}} \cos\theta_{\mathrm{eq}} + \hat{n}_{\mathrm{t}} \sin\theta_{\mathrm{eq}}, \tag{12}$$

where θ_{eq} is the static contact angle between the fluid and the wall, $\hat{n}_{\mathrm{w}}$ is the unit wall normal directed into the wall, and $\hat{n}_{\mathrm{t}}$ is tangent to the wall, normal to the contact line between the interface and the wall at $\vec{r}_{\mathrm{w}}$. The unit tangent $\hat{n}_{\mathrm{t}}$ is directed into the fluid. The angle θ_{eq} is not a fluid material property, but a system property, depending also on properties of the wall itself. The value of θ_{eq} is measured experimentally when the fluid is at rest. So in the present method the treatment of wall adhesion is static because θ_{eq} is assumed to be a constant.

2.5. Mesh calculation by finite volume method

Computational solutions of Eqs. (1) and (2) are obtained on a staged mesh (Fig. 3). The pressure, density, and viscosity are defined at the center of each cell; the horizontal component of the velocity is placed at

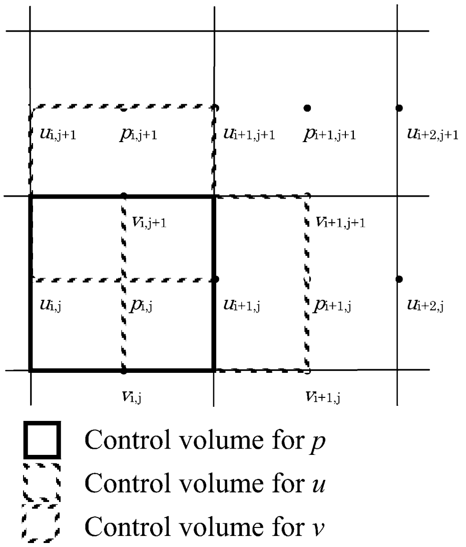

Fig. 3. Staggered mesh.

the center of the vertical cell faces; and the vertical component of the velocity is located at the center of the horizontal cell faces.

In the present method, the interface is automatically determined by the distribution of particles. The mesh is stationary and the particle is moving so information transfer is required from each other. The transfer is as follows: (1) a resultant surface force is computed on the interfacial particles and must be distributed to the background mesh; (2) the density and viscosity of the particles are distributed to the background mesh; (3) the velocity of the fluid is computed on the background mesh and must be interpolated to the particles.

The resultant surface force is distributed to the Cartesian mesh using an area-weighted extrapolation, and incorporated into Eq. (2) through the force term, $\vec{f}_V$. The procedure is as follows: (1) the cell that encloses the center of the interfacial particle is found; (2) the neighbors of the cell are found; (3) the fractional areas that the particle occupied on the neighbor cells are computed; (4) these fractional areas are used to distribute the surface force $\vec{f}_{\mathrm{sv}}$ to the four cells. This procedure is sketched in Fig. 4. The expression for the force assigned to each cell is

$$(\vec{f}_v)_k = \frac{1}{V}\vec{f}_{\mathrm{sv}}A_k, \tag{13}$$

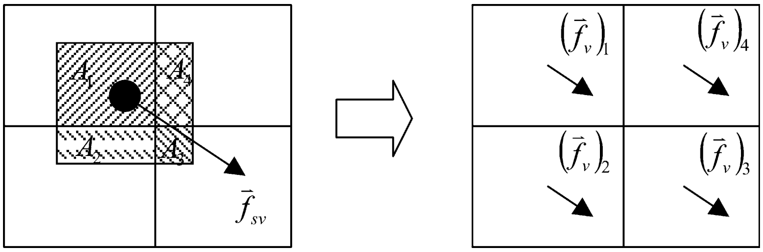

Fig. 4. Extrapolation of surface force from particle to mesh.

where k indicates each cell around the center of the particle, V is the volume of cell, A_k is the fractional area associated with the ($\sum_{k=1}^{4} A_k = 1$). The distribution is performed for all interfacial particles, and the contributions of different particles to the same cell are added. A similar expression is used for the density and viscosity of particles. The cell neighbors are included in the averaging as a form of smoothing to avoid sharp jumps in the density and viscosity, which can produce spurious spikes in the solution.

The description of fluid by particles prohibits intersected interfaces, which is a serious problem in front tracking method. If the size of particle is the same as the computational cell, at least one particle per cell is guaranteed. Since the particles move independently of one another, special measures must be taken to prevent particles from clustering. This will be discussed in the following session.

The finite-volume discretization of Eqs. (1) and (2) is used,

$$\sum_{\text{faces}} \vec{v} \cdot \vec{n}\, \Delta S = 0, \tag{14}$$

$$\frac{(\rho \vec{v} V)^{n+1} - (\rho \vec{v} V)^{n}}{\Delta t} = -\sum_{\text{faces}} \left(\bar{\bar{F}}_{\text{conv}} \cdot \vec{n}\, \Delta S \right)^{n} - \sum_{\text{faces}} \left(\bar{\bar{F}}_{\text{press}} \cdot \vec{n}\, \Delta S \right)^{n+1} + \sum_{\text{faces}} \left(\bar{\bar{F}}_{\text{visc}} \cdot \vec{n}\, \Delta S \right)^{n} + \vec{f}_{v} V, \tag{15}$$

where V is the size of the control volume; ΔS is the size of a face; $\vec{n}$ is the outward normal to the face; and the total flux has been divided into three components: convective fluxes ($\bar{\bar{F}}_{\text{conv}}$), pressure fluxes ($\bar{\bar{F}}_{\text{press}}$), and viscous fluxes ($\bar{\bar{F}}_{\text{visc}}$), which are defined in two dimensions as

$$\begin{aligned} \bar{\bar{F}}_{\text{conv}} &= \begin{bmatrix} \rho u^2 & \rho uv \\ \rho uv & \rho v^2 \end{bmatrix}, \\ \bar{\bar{F}}_{\text{press}} &= \begin{bmatrix} p & 0 \\ 0 & p \end{bmatrix}, \\ \bar{\bar{F}}_{\text{visc}} &= \begin{bmatrix} 2\mu \frac{\partial u}{\partial x} & \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & 2\mu \frac{\partial v}{\partial y} \end{bmatrix}, \end{aligned} \tag{16}$$

where u and v are the x- and y-components of the velocity vector, respectively.

In the staggered mesh, the control volume used for conservation of mass is centered at the cell centroid, the one of the conservation of momentum in the x-direction is centered at the right face, and the one for the conservation of momentum in the y-direction is centered at the upper face (Fig. 3).

The equations are solved by means of a projection method. Here the first order method is presented for discussion purposes, because the information transfer between the stationary mesh and moving particles is clear by using forward Euler time integration. We will address the effect of second order method through calculation examples in the following session. Once the mesh density field has been updated, the velocity field can be computed. By the standard first order projection method, the momentum equation is split into two parts:

$$\frac{\rho^{n+1} \vec{v}^{*} V - (\rho \vec{v} V)^{v}}{\Delta t} = -\sum_{\text{faces}} \left(\bar{\bar{F}}_{\text{conv}} \cdot \vec{n}\, \Delta S \right)^{n} + \sum_{\text{faces}} \left(\bar{\bar{F}}_{\text{visc}} \cdot \vec{n}\, \Delta S \right)^{n} + \vec{f}_{v} V \tag{17}$$

and

$$\frac{(\rho \vec{v} V)^{n+1} - \rho^{n+1} \vec{v}^{*} V)}{\Delta t} = -\sum_{\text{faces}} \left(\bar{\bar{F}}_{\text{press}} \cdot \vec{n}\, \Delta S \right)^{n+1}, \tag{18}$$

where $\vec{v}^{*}$ is temporal velocity. By using the mass conversion equation at new time,

$$\sum_{\text{faces}} \vec{v}^{n+1} \cdot \vec{n}\,\Delta S = 0, \tag{19}$$

the pressure equation is found by eliminate $\vec{v}^{n+1}$ from Eq. (18) as

$$\sum_{\text{faces}} \left[\frac{1}{\rho^{n+1}} \sum_{\text{faces}} (\bar{\bar{F}}_{\text{press}} \cdot \vec{n}\,\Delta S)^{n+1} \right] \cdot \vec{n}\,\Delta S = \frac{V}{\Delta t} \sum_{\text{faces}} \vec{v}^{*} \cdot \vec{n}\,\Delta S. \tag{20}$$

The updated velocity is then found from Eq. (18). The second order finite-volume discretization is used for the viscous term and pressure term, while the first order donor-cell scheme is used for the convection term. On the staggered mesh, a simple averaging is used for points where the variables are not defined. The time advection is calculated by explicit first order method.

In an ideal two-phase flow where the densities of the two phases are the same, Eq. (20) produces separable elliptic equation solvable by the specialized technique such as the conjugate gradient (CG) method. But when the densities are different and the equation is non-separable, the choice of method is limited. For large density ratios, more efficient methods to solve the Poisson equation of pressure may fail to converge. Here a simple successive over relaxation (SOR) method is used when the density ratio is large, and the multigrid method is applied to accelerate the iteration. The over-relaxation parameter by which the pressure equation is solved depends on the density ratio of different fluids. Sometimes the alternating direction implicit (ADI) method is used when the density ratio is modest.

2.6. Particle calculation

Since the fluid velocities are computed on the fixed mesh and particles move with the fluid velocities, the velocity of particles must be found by interpolating from the fixed mesh. The interpolation starts by identifying the grid points that are closest to the particle. The particle velocity is then interpolated by area-weighted interpolation which incorporates most of the features developed for the distribution of the surface force described above. Once the velocity of each particle has been found, its new position can be found by integration

$$\vec{r}^{*} = \vec{r}_i^{n} + \vec{v}_i^{n} \Delta t, \tag{21}$$

where $\vec{r}$ is the particle position, $\vec{v}$ is the particle velocity, and Δt is the time step. The accuracy of this integration is the same as that in solving momentum equation (15), which is explicit first order. In the future we will use higher order time integration method to increase the accuracy.

The continuity equation requires that the fluid density should be constant. This is equivalent to the particle number density being constant, n^0. When the particle number density n* is not n^0, it is corrected to n^0

$$n^{*} + n' = n^0, \tag{22}$$

where n' is the correction value. This is related to the velocity correction value $\vec{v}'$ through the mass conservation equation [11]:

$$\frac{1}{\Delta t} \frac{n'}{n^0} = -\nabla \cdot \vec{v}'. \tag{23}$$

The velocity correction value is derived from a correction pressure gradient term as

$$\vec{v}' = -\frac{\Delta t}{\rho} \nabla p', \tag{24}$$

where p' is the correction pressure, ρ is the density of fluid represented by particles. With Eqs. (22)–(24), a Poisson equation of correction pressure is obtained:

$$\langle \nabla^2 p' \rangle_i = -\frac{\rho}{\Delta t^2} \frac{\langle n^* \rangle_i - n^0}{n^0}. \tag{25}$$

Deterministic particle interaction models representing gradient and Laplacian given in the MPS method [11–19] are second order accurate for symmetrically distributed particles. The Poisson equation is solved by the incomplete Cholesky conjugate gradient (ICCG) method [11] with a Dirichlet boundary condition ($p' = 0$) given to the particles at interfaces. It should be noted that the correction pressure in Eq. (25) is different from the pressure defined in Eq. (20). Here the correction pressure p' is defined on particles, which originates from the deviation of particle number density. This pressure correction value is used to modify the position of particles. While the pressure p in Eq. (20) is defined on mesh, which is solved on the whole computational domain as a field variable.

The velocity correction is computed by Eq. (24), and the position of particle is modified by the velocity correction value:

$$\vec{r}_i^{\,n+1} = \vec{r}_i^{\,*} + \vec{v}_i^{\,\prime} \Delta t. \tag{26}$$

In each time step a well-distributed particles is guaranteed by keeping the particle number density. After the particle's position is adjusted, the particle's velocity is omitted. Only the velocities defined on mesh remain.

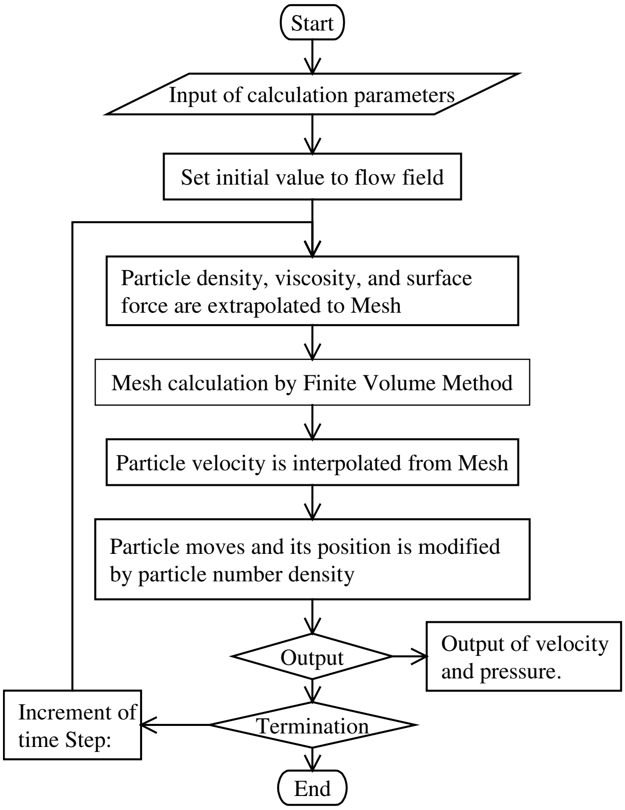
Start
Input of calculation parameters
Set initial value to flow field
Particle density, viscosity, and surface force are extrapolated to Mesh
Mesh calculation by Finite Volume Method
Particle velocity is interpolated from Mesh
Particle moves and its position is modified by particle number density
Output
Output of velocity and pressure.
Termination
Increment of time Step:
End

Fig. 5. Algorithm of the hybrid method.

2.7. Solution algorithm

The overall algorithm is described in Fig. 5. First, the particle's density and viscosity, and the surface force calculated on the interfacial particles are extrapolated to mesh, and then the velocity and pressure fields are calculated on mesh by finite volume method. Next, the particle's velocity is interpolated from the mesh, and then the particle moves with this velocity. Finally, the position of particle is modified by the particle number density.

Since the velocity update algorithm of Eqs. (17) and (18) is explicit, there is a restriction on the maximum time step for a stable solution,

$$0.25(|u| + |v|)^2_{\max}\Delta t\frac{1}{v} \leqslant 1 \quad \text{and} \quad \frac{\Delta t}{\frac{1}{v}\Delta x^2} \leqslant 0.25, \tag{27}$$

where v is kinematic viscosity, Δx is the size of mesh cell, and it is assumed that $\Delta x = \Delta y$.

3. Computational examples

Here present the results for several standard static and dynamic problems with surface tension to illustrate the flexibility and accuracy of the hybrid method.

3.1. Equilibrium rod

A static liquid drop will become spherical under surface tension in the absence of gravitational or other external forces. For an infinite cylinder surrounded by a background fluid at zero pressure, the internal drop pressure, p_{drop}, will be

$$p_{\text{drop}} = \sigma\kappa = \sigma/R, \tag{28}$$

where R is the drop radius. Here the calculation of equilibrium rod with surface tension is carried out on a two-dimensional computational domain. A regular orthogonal 66 × 66 mesh ($\Delta x = \Delta y = 0.001$ m) parti-

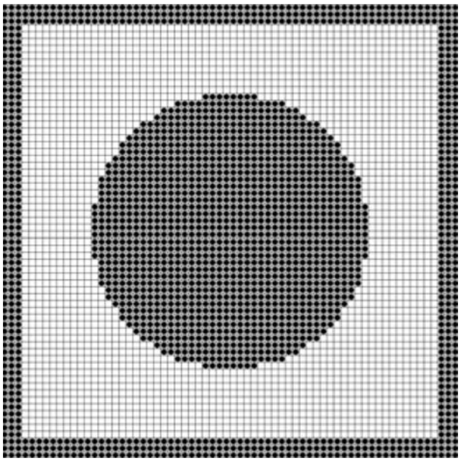

Fig. 6. Initial configuration of mesh and particles.

tions the domain, with background density $\rho = 500$ kg/m^3. A fluid drop with radius $R = 0.02$ m, density $\rho = 1000$ kg/m^3, and surface tension coefficient $\sigma = 0.02361$ N/m, is centered at the point (0.03, 0.03). The fluid drop is represented by 2020 particles with distance $D = 0.001$ m between them. The configuration of mesh and particle is shown in Fig. 6. Since the initial distribution of particles does not form exactly a rod, the particle's position will be adjusted under surface tension to form a more circular one. Fig. 7 shows the position of particles at about 0.5 s. From Eq. (28), the pressure inside the fluid drop is 1.1805 Pa under the surface tension coefficient given above. This value is compared with the mean drop pressure computed by an average process defined as

$$\langle p \rangle = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i, \tag{29}$$

where the sum is over the N_p particles lying within the drop that have particle number density $n_i > \beta n^0$. Fig. 8 shows the computed mean drop pressure by Eq. (29) and its variation with time. Since the particles on the

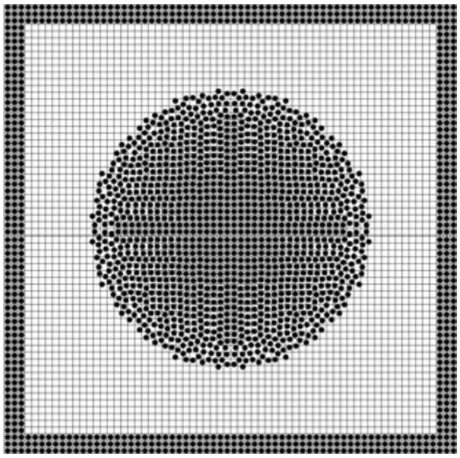

Fig. 7. Mesh and particles at time 0.5 s.

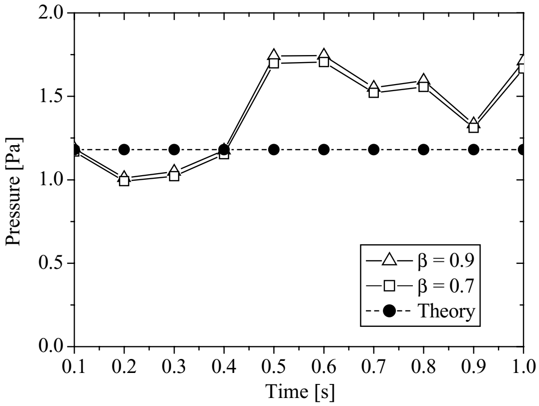

Fig. 8. Mean pressure of the liquid rod.

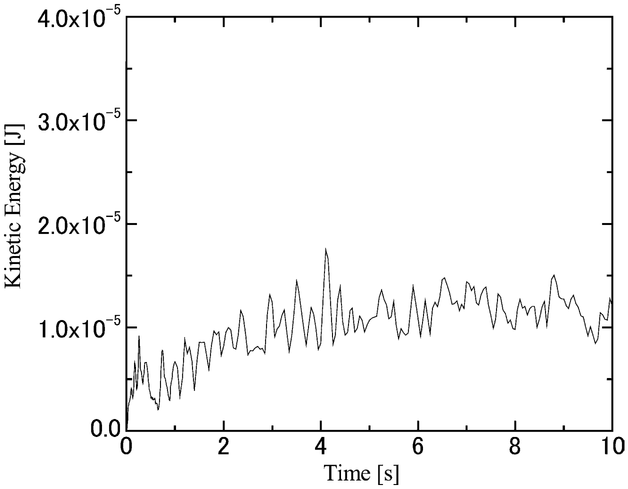

Fig. 9. Kinetic energy versus time for the equilibrium rod.

interface oscillate around the equilibrium position, the mean drop pressure waves around the analytical value. The result suggests that smoothing algorithm for the surface tension calculation is necessary to obtain an accurate, uniform drop pressure. Without smoothing process, relative errors in the curvature of interface are nearly 80%. However, the convergence of surface tension model is demonstrated by the kinetic energy versus time shown in Fig. 9, in which the kinetic energy increases during 0–4 s but maintains at about 1×10^{-5} J. The result in Fig. 8 also shows that the definition of interfacial particles by Eq. (6) is not sensitive to the parameter β.

The mesh refinement study for the equilibrium rod is shown in Fig. 10, with a coarse mesh ($\Delta x = \Delta y = 0.002$ m) and a fine mesh ($\Delta x = \Delta y = 0.0005$ m). To examine the convergence rate of the solu-

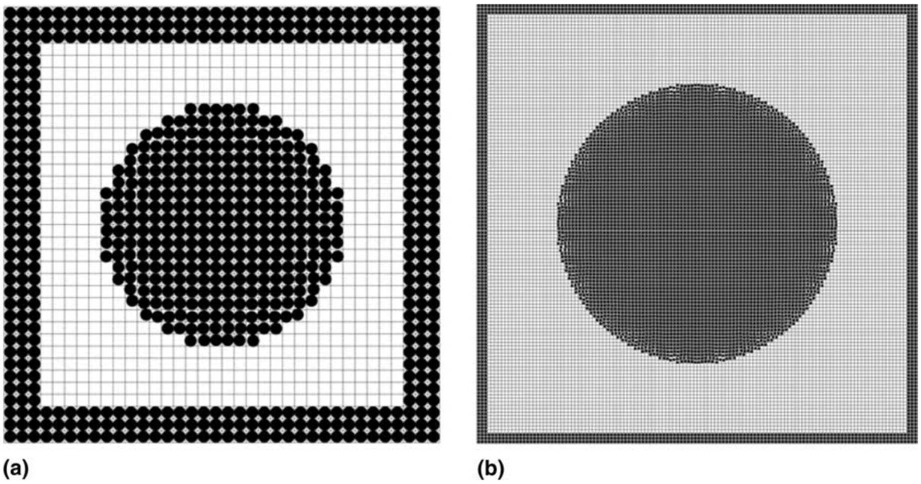

Fig. 10. Mesh refinement test for equilibrium rod. (a) Coarse mesh, (b) fine mesh.

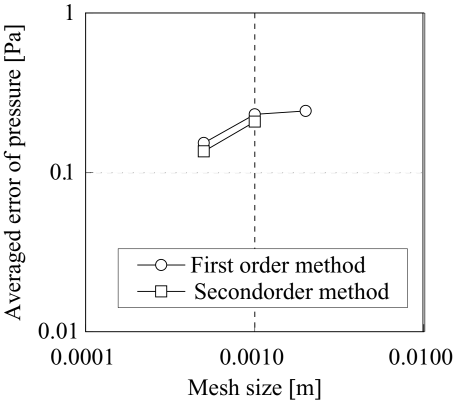

Fig. 11. The error of mean drop pressure as a function of mesh size.

tion with respect to mesh size, the time-averaged error of the mean drop pressure is given in Fig. 11 as a function of the mesh size. This figure shows that the convergence rate with respect to mesh size is close to linear.

In the present method, the surface tension force is calculated based purely on the distribution of particles. Therefore, the accuracy of surface tension model has no relation with the density of particle. As discussed in the session of the pressure equation, the efficiency of Poisson solver depends on the density ratio of different fluids. In general, the Poisson solver requires more iteration with smaller relaxation parameter for large density ratios.

In order to achieve a second order accuracy, we have tried to use second order centered difference scheme for the spatial variables in mesh calculation. But to achieve a second order accuracy in time is more complex because the implicit treatments of particle motion and surface tension force are necessary. In this static drop problem we have used a simple predictor-corrector scheme. At first the explicit solution at third step $n + l$ serves as a predictor. Then the centered difference scheme for the time is used for time integration. The surface tension force is calculated using position of the interfacial particles at time of $n + 1/2$. The calculations by the second order scheme on two different meshes ($\Delta x = \Delta y = 0.001$ m and $\Delta x = \Delta y = 0.0005$ m) are carried out with the same time intervals as those in the calculations by the first order scheme. The time-averaged errors of the mean drop pressure are compared in Fig. 11. The result shows that the accuracy is improved but the convergence rate with respect to mesh size is not quadratic. The main reason is that the accuracy of surface tension model is first order. As discussed above, the non-smooth distribution of the interfacial particles leads to pressure fluctuations, which also contributes to a decrease in the overall convergence rate and the numerical stability. In the future we will use smoothing technique to increase the accuracy of surface tension model.

3.2. Non-equilibrium rod

An initially square drop is numerically calculated to observe the behavior responding to unbalanced surface tension forces. A square ethanol drop with $\rho = 797.88$ kg/m^3 and $\sigma = 0.02361$ N/m is centered on a domain with 96×96 mesh ($\Delta x = \Delta y = 0.0025$ m). The drop consists of 30×30 particles ($D = 0.0025$ m). A small density ($\rho = 1.0$ kg/m^3) and a small viscosity ($\mu = 10^{-5}$ Pa s) are given to the background fluid defined on the mesh. The gravitational and other external forces are neglected. The oscillations of the surface of the drop about its equilibrium shape are shown in Fig. 12 at a time sequence, $t = 0$, 0.575, 1.250, and

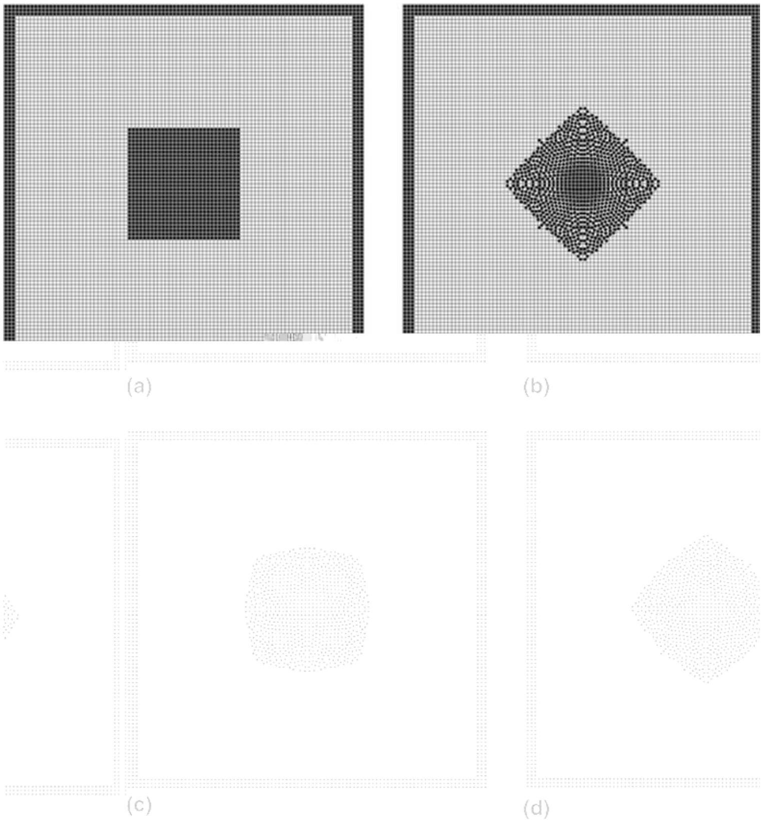

Fig. 12. Vibration of square ethanol drop under surface tension. (a) $t = 0$ s, (b) $t = 0.575$ s, (c) $t = 1.25$ s, (d) $t = 1.825$ s.

1.825 s. The initially square shape of the drop results in very strong surface forces at the high-curvature corners, setting the drop into oscillation. The period of oscillation is approximately 1.25 s. An analytic expression for the oscillation frequency of a 2D drop in zero gravity [21] is

$$\omega_n^2 = \frac{(n^3 - n)\sigma}{(\rho_{\mathrm{d}} + \rho_{\mathrm{e}})R_0^3}, \tag{30}$$

where ρ_{d} is the drop density, ρ_{e} is the density of the external fluid, n is the mode number of oscillation ($n = 4$ in the present calculation), R_0 is unperturbed radius of the drop. Eq. (30) gives a theoretical period of the oscillation $\tau = 1.299$ s by the definition as $\tau = 2\pi/\omega_n$. The oscillation period by the simulation is a little shorter than the analytic one. The discrepancy comes from the presence of boundaries a finite distance away, and the effect of nonlinearity in the numerical simulation because the theoretical value by Eq. (30) is from a linear analysis.

In the numerical calculation, the oscillations damp because of viscosity of fluids. The decay of the oscillation is apparent in the variation of kinetic energy shown in Fig. 13. At about $t = 4.0$ s, the drop is nearly circular in cross section. Animations of computational results indicated that the distribution of particles at interface cause discontinuous changes of interface location. These provide a small impulse trough the surface tension to sustain the oscillations.

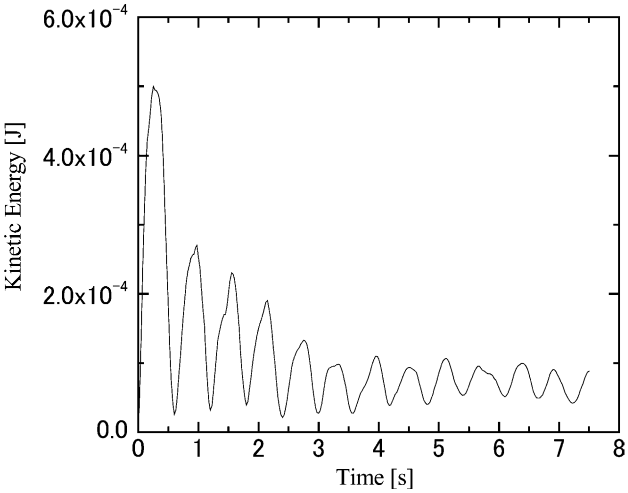

Fig. 13. Kinetic energy versus time for the non-equilibrium rod.

The mesh refinement study for the non-equilibrium rod is carried out with a fine mesh ($\Delta x = \Delta y = 0.0015$ m). The oscillations of the drop surface about its equilibrium shape are shown in Fig. 14 at a time sequence, $t = 0$, 0.5, 1.1, and 1.75 s. The variation of kinetic energy is compared with that by coarse mesh in Fig. 15. The initially square shape of the drop results in even stronger surface forces at the high-curvature corners by the fine mesh, resulting a shorter period for the first oscillation. But the period of the following oscillations is about the same as that by coarse mesh.

3.3. Equilibrium contact angle

By definition, wetting is the ability of liquids to form a boundary surface with solids. It plays an important role in many technical processes such as adhesives, coating, oil recovery, gluing cleaning, etc. The phenomenon of wetting or non-wetting of a solid wall by a liquid is illustrated by studying what is known as contact angle. The contact angle θ describes the shape of a liquid drop adsorbed at the solid wall. If $\theta = 0$, the solid wall preferentially adsorbs a layer of liquid, which is described as wetting. While if $0 < \theta < 90°$, the solid wall is wetted with only cap-shaped liquid drop adsorbed. If $\theta > 90°$ it is a layer of the gas phase which is adsorbed at the solid wall, sometimes described as non-wetting.

A liquid drop lying on a solid wall is numerically simulated to observe the equilibrium shape of drop under different contact angles. A square ethanol drop with $\rho = 797.88$ kg/m^3 and $\sigma = 0.02361$ N/m is initially put on the solid wall. The drop consists of 30 × 30 particles ($D = 0.002$ m). A small density ($\rho = 1.0$ kg/m^3) is given to the background fluid defined on the 146 × 46 mesh ($\Delta x = \Delta y = 0.002$ m). The gravitational and other external forces are neglected. A large viscosity ($\mu = 10^{-2}$ Pa s) is given for the background fluid. The shapes of the drop at equilibrium state are shown in Fig. 16 under different static contact angles, $\theta_{eq} = 30°, 60°, 90°, 120°$, and 150°. The initially square shape of the drop deforms under the strong surface forces at the high-curvature corners and the adhesive forces of solid wall. The oscillations of drop shape damp because of large viscosity, and finally an equilibrium state of drop shape is obtained. Fig. 17 gives the plots of kinetic energy versus time for contact angles 60° and 120° as two examples to demonstrate the stability of the numerical method.

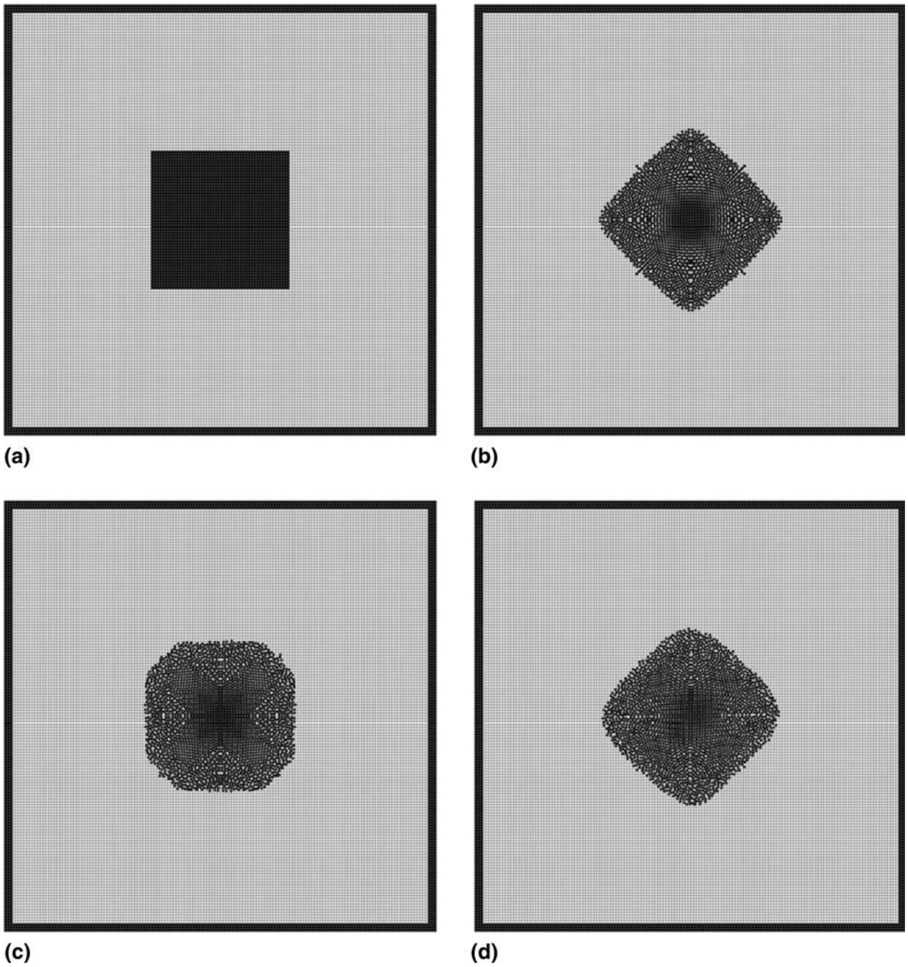

Fig. 14. Vibration of square ethanol drop under surface tension by fine mesh. (a) $t = 0$ s, (b) $t = 0.5$ s, (c) $t = 1.1$ s, (d) $t = 1.75$ s.

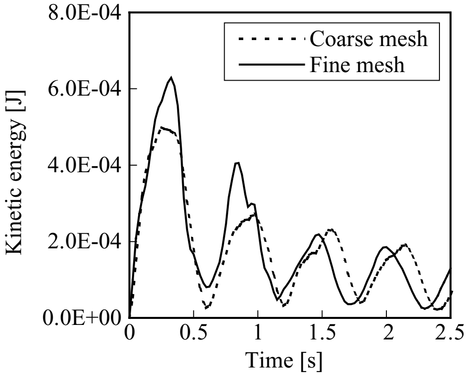

Fig. 15. Comparison of kinetic energy for the non-equilibrium rod.

3.4. Flow induced by wall adhesion

As an example of the effects of wall adhesion, consider a shallow pool of water located at the bottom of a two-dimensional tank. Assume that the water interface wants to attain a specified contact angle θ_{eq} with the tank wall, different from the initial angle of 90° (a horizontal interface). Two different cases are computed in Figs. 18 and 19, one in which the water wets the wall ($\theta_{\mathrm{eq}} < 90°$), and one in which the water does not wet the wall ($\theta_{\mathrm{eq}} > 90°$). The results are computed on a 56×76 mesh ($\Delta x = \Delta y = 0.002$ m) with particles representing the water. For the wetting case, the pool of water is 0.05 m deep and $\theta_{\mathrm{eq}} = 5°$. For the non-wetting case, the pool of water is 0.02 m deep and $\theta_{\mathrm{eq}} = 175°$. All external forces such as gravity are set to zero.

Fig. 18

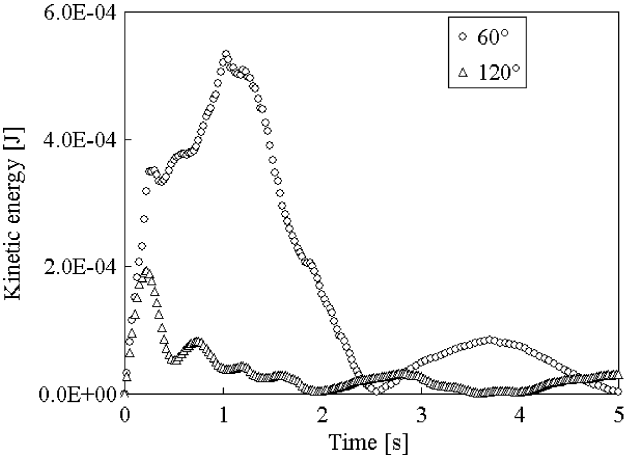

Fig. 17. kinetic energy versus time for contact angles 60° and 120°.

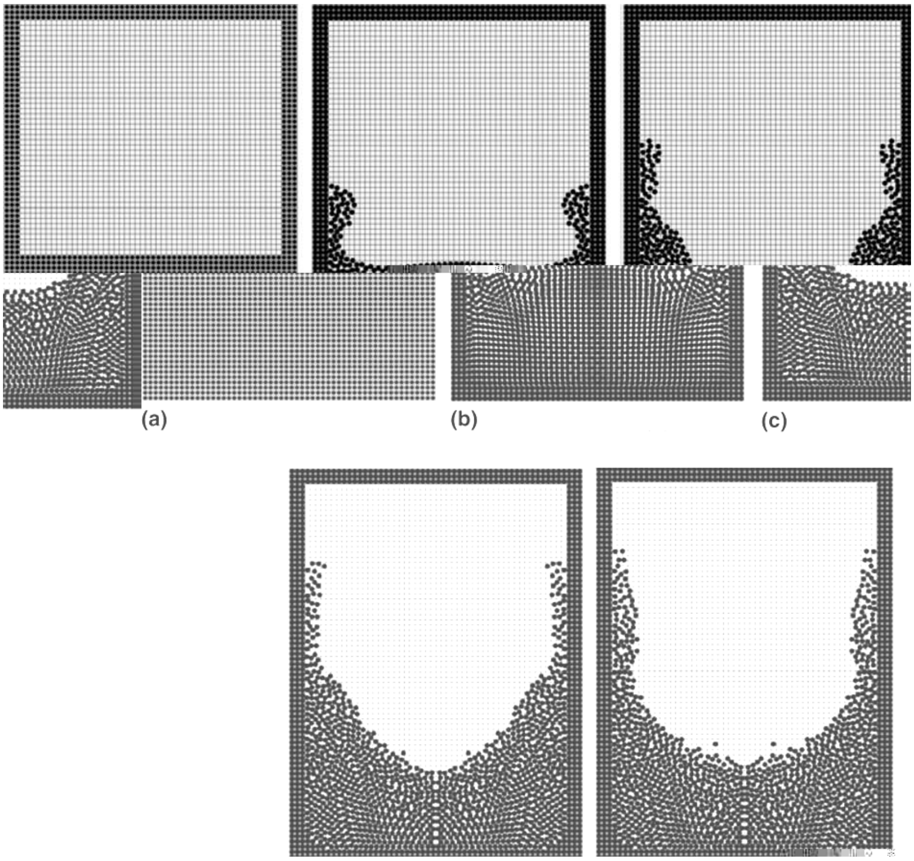

Fig. 18. Flow induced by wall adhesion for the wetting case. (a) $t = 0$ s, (b) $t = 0.5$ s, (c) $t = 1.0$ s, (d) $t = 1.5$ s, (e) $t = 2.0$ s.

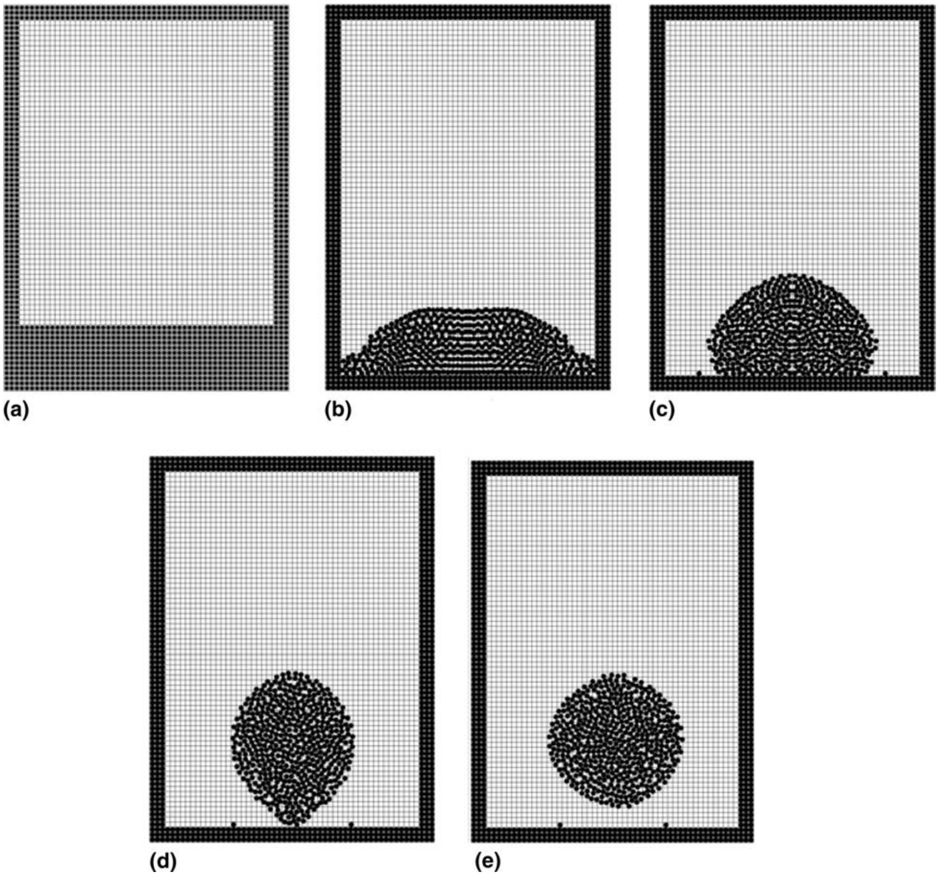

Fig. 19. Flow induced by wall adhesion for the non-wetting case. (a) $t = 0$ s, (b) $t = 0.5$ s, (c) $t = 1.0$ s, (d) $t = 2.0$ s, (e) $t = 3.0$ s.

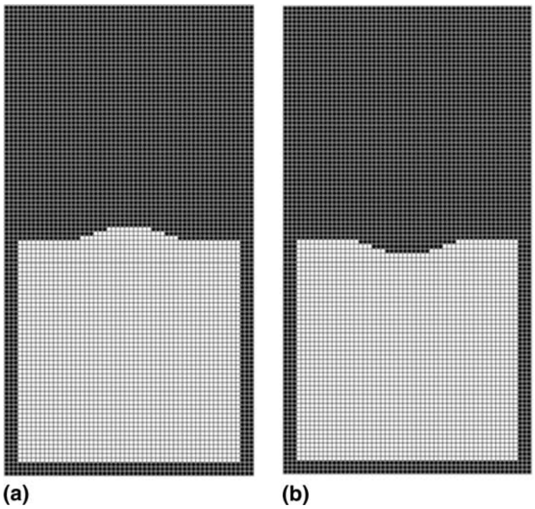

Fig. 20. Initial states of Rayleigh–Taylor instability. (a) Upward disturbance, (b) downward disturbance.

upward surface forces at contact location. From the sequence of times in Fig. 18, $t = 0$, 0.5, 1.0, 1.5, and 2.0 s, it is evident that the flow field due to the wall adhesion forces for $\theta_{eq} = 5°$ causes the water to move up the tank walls until the boundary condition in Eq. (12) is satisfied.

There exists a problem in the calculation. When the water film on the tank wall becomes very thin (for example, only 2 or 3 particles' thick), here the particle has very few neighbors and the calculation based on particle interactions is not accurate enough. For example, the calculation of gradient by Eq. (10) or divergence by Eq. (11) is an average process over the neighbors of one particle. This problem causes the interface unstable after $t = 2.0$ s in Fig. 18.

In the non-wetting case, a "ball" of water evolves from an initial shallow pool when the wall adhesion forces are specified on every boundary with $\theta_{eq} = 175°$. The water behaves in this case like mercury, wanting to separate itself from the walls since contact angle is obtuse. The shape of water is displayed at times of

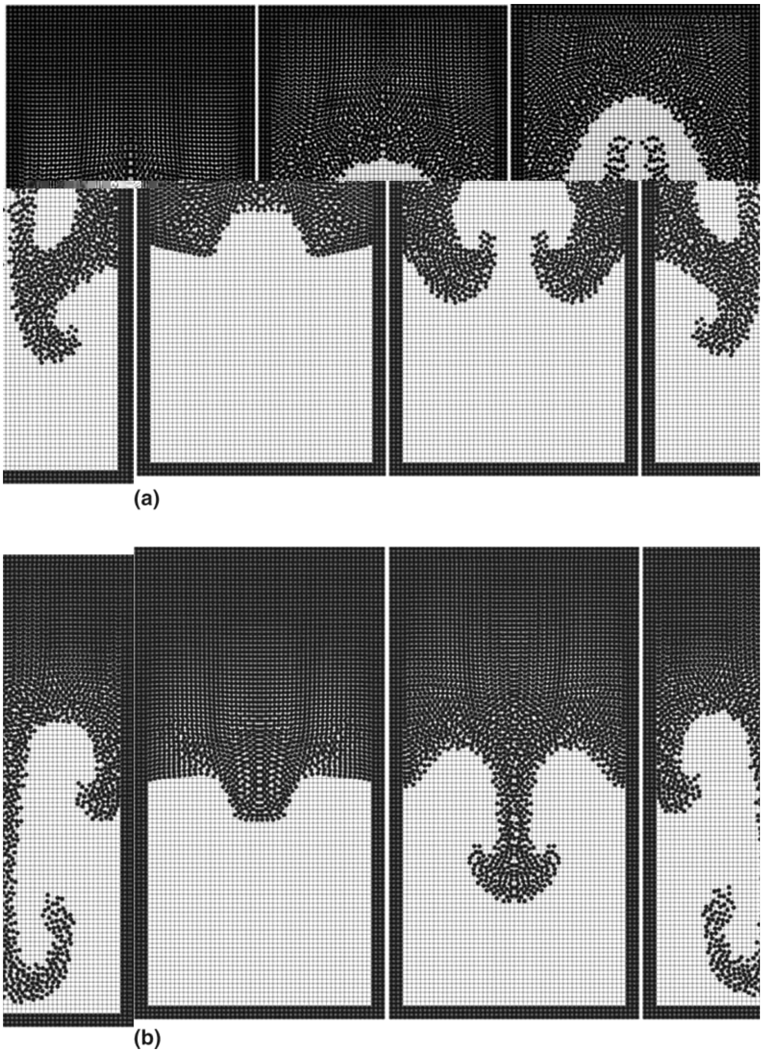

Fig. 21. Evolutions of R–T instabilities ($t = 0.4$, 0.8, and 1.2) without surface tension. (a) Upward disturbance case, (b) downward disturbance case.

$t = 0$, 0.5, 1.0, 2.0, and 3.0 s in Fig. 19. A net upward momentum, evident in Fig. 19(e), is imported to the water ball by the wall adhesion force, eventually causing it to separate from the bottom wall.

3.5. Rayleigh–Taylor instability

The Rayleigh–Taylor (R–T) instability is created when a heavy fluid is put upon a lighter one and when the equilibrium state is perturbed. As a basic tow-phase flow phenomenon, the R–T instability is numerically simulated by using the present method. In the initial state of the phenomenon, the heavier fluid $\rho_1 = 3$ lies above the lighter one $\rho_2 = 1$ in a channel with width 1 and height 2. The gravity acceleration is given as $g = 10$. Both fluids have the same kinematic viscosity $v_1 = v_2 = 0.01$. Two disturbance cases, one is upward and another is downward, are given by the sinusoidal interfaces with the amplitude 0.06 as shown in Fig. 20. The heavy fluid is represented by particles with distance $D = 0.02$, and the lighter one is defined on mesh ($\Delta x = \Delta y = 0.02$). All quantities mentioned above are non-dimensional.

The various behaviors of the two-phase system are observed at equivalent times and the effects of surface tension are illustrated. Fig. 21 shows the evolutions ($t = 0.4$, 0.8 and 1.2) of instabilities of the two cases without surface tension ($\sigma = 0$). In upward disturbance case, the lighter fluid moves upward and a bubble is formed. While in downward disturbance case, a jet of heavier fluid falls down. Further developments of the phenomenon mix the fluids to the dispersed state, because there is no surface tension which would limit the dispersion.

The growth rate of the R–T instability is given by [20]

$$n^2 = kg\left[A - \frac{k^2\sigma}{g(\rho_1 + \rho_2)}\right], \tag{31}$$

where k is the wave number of the perturbation, g is the gravitational acceleration perpendicular to the interface, and $A = (\rho_2 - \rho_1)/(\rho_2 + \rho_1)$ is the Atwood number. The perturbation of interface will grow exponentially in time as $\exp(nt)$ for small amplitudes. The effect of fluid viscosity on the development of the R–T instability, which plays a stabilizing role, has been ignored in Eq. (31). The correction to the growth rate that takes the viscous effect into account is given by the positive root of [22]

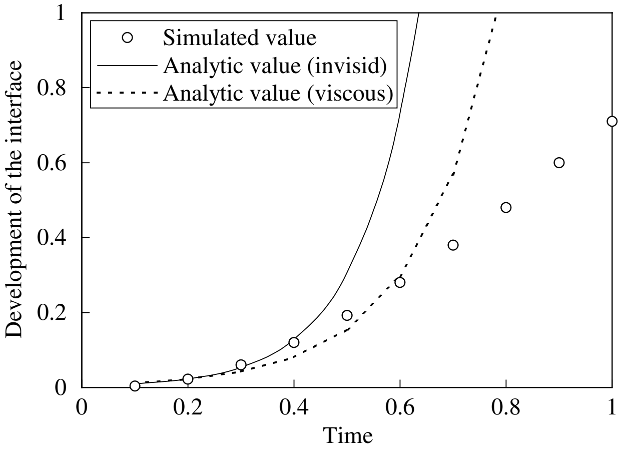

Fig. 22. Development of the interface in R–T instabilities without surface tension.

$$n^2 + 2vk^2n - kgA = 0, \tag{32}$$

where v is the kinematic viscosity.

To examine the accuracy of the present numerical method, the growth rate of R–T instability in the downward disturbance case without surface tension is compared with the analytic values given by Eqs. (31) and (32). Here the Atwood number is $A = 0.5$, and the wave number k can be estimated by the wavelength λ as $k = 2\pi/\lambda$. The growth rate by Eq. (31) is about $n_1 = 8.68$, and the growth rate by Eq. (32) is $n_2 = 6.50$. The comparison of the interface developments between the simulation result and the analytic values calculated by n_1 and n_2 is given in Fig. 22. The comparison shows that, for small amplitudes, the simulation result agrees well with the analytic growth rate $n_2 = 6.50$, which validates the damping effect of the viscosity. When the amplitudes become large, the heavy fluid rolls up with a slight downward velocity.

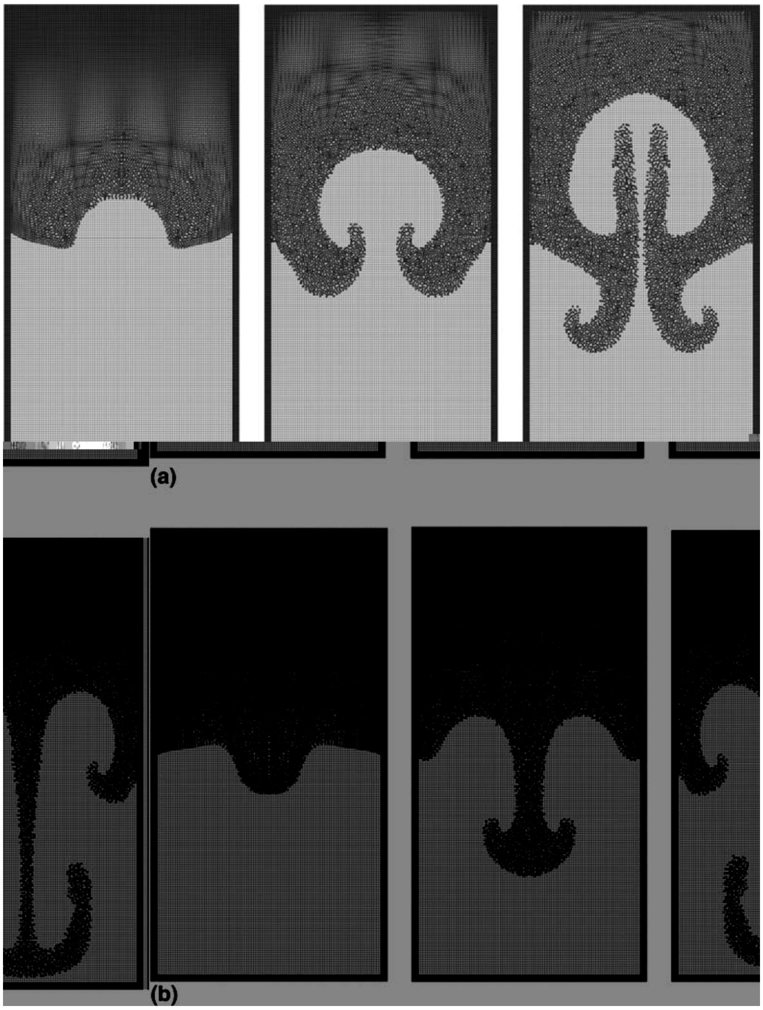

Fig. 23. Evolutions of R–T instabilities (t = 0.4, 0.8, and 1.2) with fine mesh. (a) Upward disturbance case, (b) downward disturbance case.

Fig. 23 shows the evolutions ($t = 0.4$, 0.8, and 1.2) of above instabilities with a fine mesh ($\Delta x = \Delta y = 0.01$ m). Compared with coarse mesh in Fig. 21, the outlines of the interfaces become clear, although the amplitudes of the jet show a weak dependence on resolution.

Fig. 24 shows the developments of instability of the two cases with a surface tension ($\sigma = 0.127$) at times of $t = 0.5$, 1.0, and 1.5. The effect of surface tension is effectively observed that the surface force tends to make the interface as flat as possible. The curvatures of the interfaces are less than those without surface tension. The growth rate of R–T instability in the downward disturbance case by Eq. (31) with surface tension ($\sigma = 0.127$) is about $n = 4.91$. The comparison of the interface developments between the simulation result and the analytic value calculated by $n = 4.91$ is given in Fig. 25. The comparison shows that, for small amplitudes, the simulation result agrees well with the analytic value.

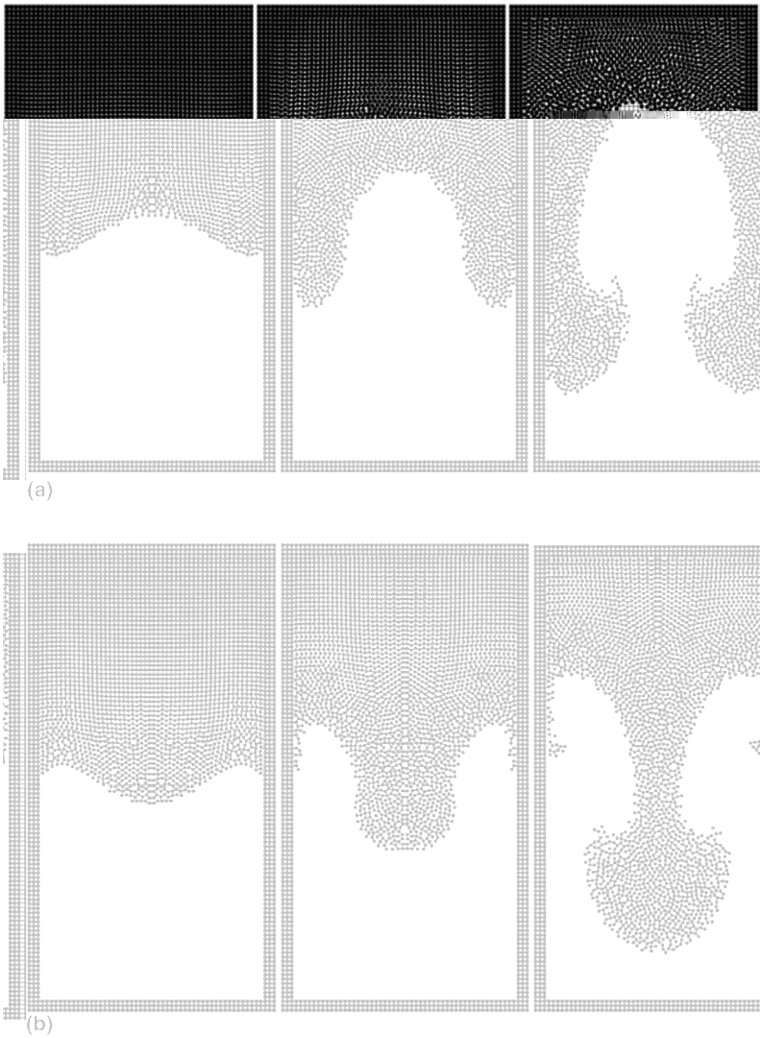

Fig. 24. Evolutions of R–T instabilities ($t = 0.5$, 1.0, and 1.5) with surface tension ($\sigma = 0.127$). (a) Upward disturbance case, (b) downward disturbance case.

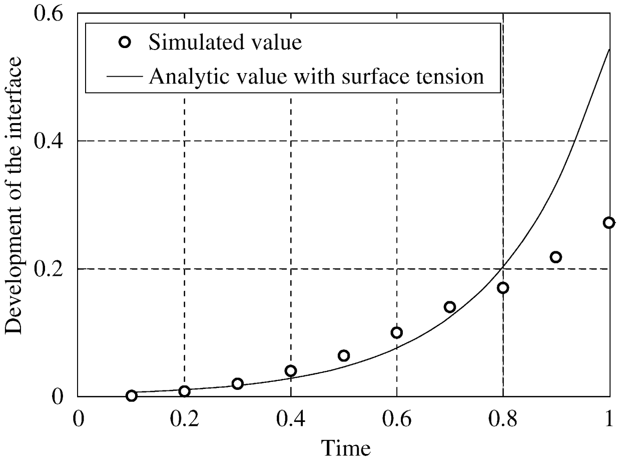

Fig. 25. Development of the interface in R–T instabilities with surface tension ($\sigma = 0.127$).

A critical surface tension, σ_c, can be calculated from Eq. (31) for which the growth rate becomes zero ($n = 0$). If it is assumed that the wavelength equals to the width of the channel, the critical surface tension is about $\sigma_c = 0.253$. The R–T instabilities of the two disturbance cases are simulated with this critical surface tension. Fig. 26 gives the evolutions of instability at times of $t = 1.0$, 2.0, and 3.0, which show the deformations of the interface in a, different way. Although the disturbance initiates the instability, the surface tension prohibits it from developing. Due to the numerical error, the symmetric state of the system is lost, which results in another disturbance on the balance of interface. The perturbation develops and finally the heavy fluid flows downward near one sidewall of tank. The phenomena can be explained that the perturbation tends to develop under a larger wavelength than the width of the channel. Therefore, the critical surface tension increases with larger wavelength and the stability parameter $\Phi = \sigma/\sigma_c$ becomes less than 1.0.

3.6. Kelvin–Helmholtz instability

The Kelvin–Helmholtz instability is a fundamental instability of incompressible fluid flow, which will arise, for example, from two parallel fluids of different densities moving at different velocities. The likelihood of Kelvin–Helmholtz instability can be evaluated by means of the Richardson's number (Ri) [23]:

$$\text{Richardson number} = \frac{\text{Stabilizing Buoyancy}}{\text{Destabilizing Shear}}. \tag{33}$$

The instability occurs when the destabilizing shear is strong enough to break up the stable layer across the interface between two fluids.

Here a simple way is considered to generate the Kelvin–Helmholtz instability. In a long narrow tank a lighter freshwater rests on a heavier saltwater with a stable boundary between the two fluids. If the tank is tilt at a angle, the saltwater flows down and the freshwater flows upward, causing a current speed difference at the boundary of the two types of water. When the current speed difference exceeds a certain level, the current becomes unstable. The nonlinear growth and evolution of this interface are governed by only the competing effects of the buoyancy and shear.

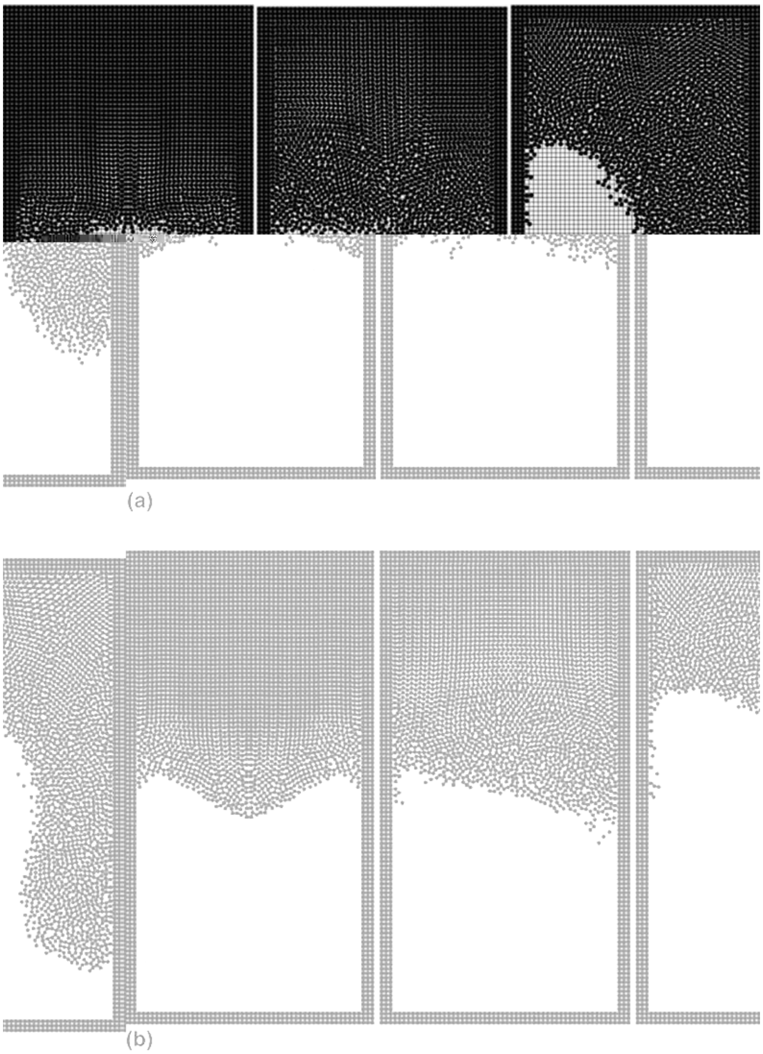

Fig. 26. Evolutions of R–T instabilities ($t = 1.0$, 2.0 and 3.0) with surface tension ($\sigma = 0.253$). (a) Upward disturbance case, (b) downward disturbance case.

The calculation is carried out on a two-dimensional geometry shown in Fig. 27(a) with the present method. To tilt the tank is equivalent to give the gravity in the direction different from downward vertically (with an angle $\theta = 30°$). The densities of freshwater and saltwater are 1000 and 1050 kg/m^3, respectively. The gravity acceleration is given as $g = 9.8$ m/s^2. Surface tension is neglected. The saltwater is represented by particles with distance $D = 0.002$ m, and the freshwater is defined on mesh ($\Delta x = \Delta y = 0.002$ m), shown in Fig. 27(b).

Fig. 28 shows the evolution of instability at times of $t = 1.2$, 1.4, 1.6, 1.8, and 2.0. The interface rolls up and forms a wave which grows quickly. In the present calculations, the freshwater and saltwater flow to either side without mixing. However, in the experiments [23] the two kinds of fluid mix violently and the mixing finally ends up with a turbulent state.

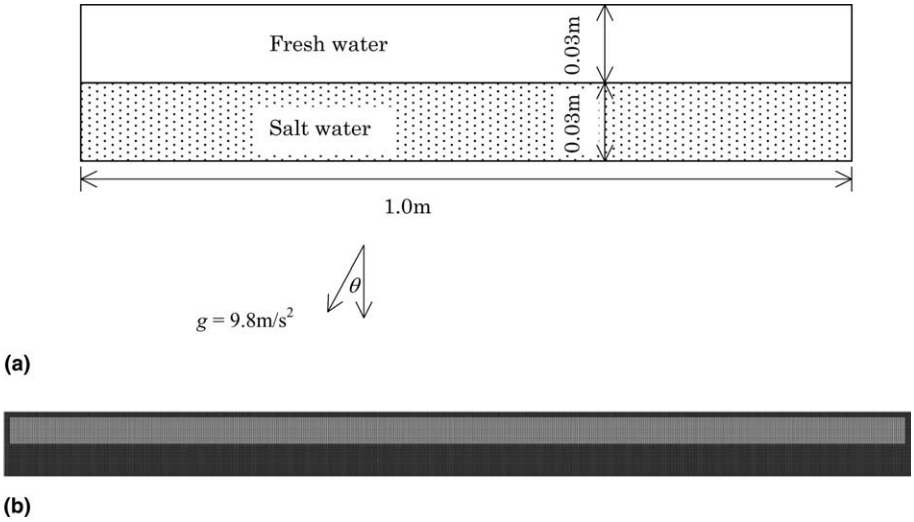

Fig. 27. Calculation condition for Kelvin–Helmhotz instability. (a) Calculation geometry, (b) initial setup of mesh and particles.

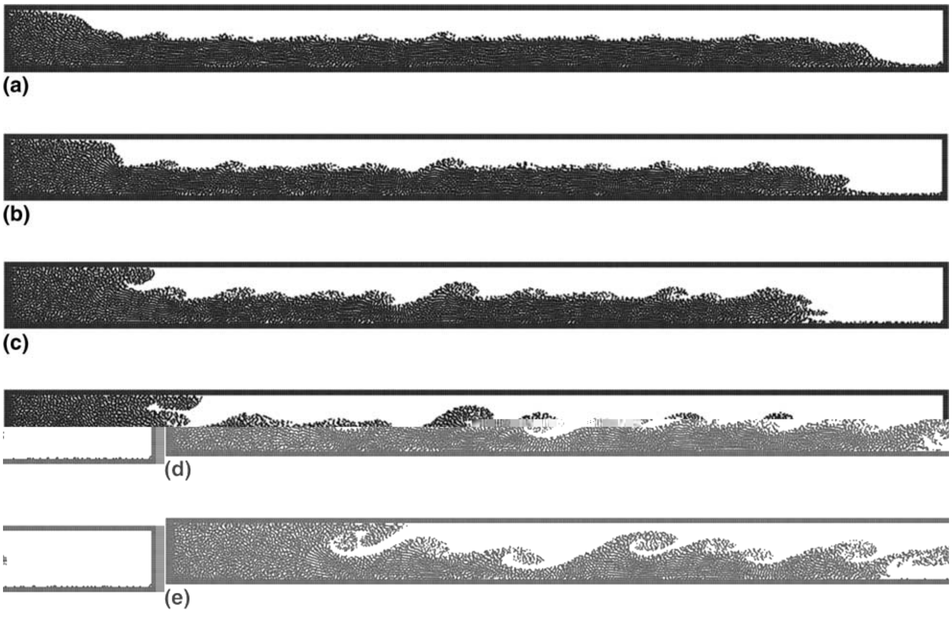

Fig. 28. Evolution of Kelvin–Helmhotz instability. (a) $t = 1.2$ s, (b) $t = 1.4$ s, (c) $t = 1.6$ s, (d) $t = 1.8$ s, (e) $t = 2.0$ s.

4. Conclusions

A hybrid numerical method that coupled the mesh and particle has been developed for the unsteady, incompressible, multiphase flows. One phase (usually the liquid phase) is represented by the particles and another phase (usually the gas phase) is defined on mesh. The formulation of discontinuous interface problem as a continuum problem is realized by extrapolating the density and viscosity of particles onto mesh. The conservation equations are solved on the stationary mesh with finite-volume method, and the sharp interface is automatically tracked by the moving particles. This prevents both numerical diffusion and oscillations and allows surface tension to be incorporated in a natural way. The original Eulerian mesh is retained through the simulation. A simple SOR method is used in solving the pressure equation, which is successful in large density ratio problems. Since a conservative finite volume scheme is used, no special treatment is needed near the interface. The use of a regular fixed mesh makes coding the method relatively simple. Several test cases suggest that the method is both robust and computationally efficient.

The CSF models used to calculate the surface tension effects at the interfaces and the wall adhesion. The surface tension model has been validated on both static and dynamic interfaces having surface tension. It has been applied successfully to a number of fluid flows driven by surface tension.

There is no limitation of the method for three-dimensional simulations. Furthermore, the interface is determined by the distribution of particles, no special algorithm is necessary to deal with interface interactions. The mass conservation of different phases is also preserved by the Lagrangian particles.

The method could be improved to achieve higher accuracy and faster convergent rate by the use of more sophisticated interactions between particles and mesh. A smoothing technique is required to compute the surface tension. Some high order upwind schemes will be used for the advection terms at high Reynolds numbers.

Although the present method has been tested extensively on two-dimensional problems, the major purpose of developing this method is to simulate fully three-dimensional multiphase flows. The test cases presented here suggest that large simulations involving interacting interfaces are entirely feasible. The implementation of the method in three dimensions appear to be straightforward and the method would be extended to many new and physically interesting problems.

Acknowledgement

This research was partly funded by Japan Society for the Promotion of Science (FY2002 JSPS Postdoctoral Fellowship for Foreign Researchers, ID No. P02121).

References

[1] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182.

[2] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201.

[3] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, J. Comput. Phys. 169 (2) (2001) 556.

[4] D. Jacqmin, Calculation of two-phase Navier-Stokes flows using phase-field modeling, J. Comput. Phys. 155 (1999) 96.

[5] J.M. Floryan, H. Rasmussen, Numerical methods for viscous flows with moving boundaries, Appl. Mech. Rev. 42 (12) (1989) 323.

[6] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25.

[7] J. Glimm, J.W. Grove, X.L. Li, K.M. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM J. Sci. Comput. 19 (3) (1998) 703.

[8] J. Glimm, J.W. Grove, X.L. Li, D.C. Tan, Robust computational algorithms for dynamic interface tracking in three dimensions, SIAM J. Sci. Comput. 21 (6) (2000) 2240.

[9] A.A. Amsden, The particle-in-cell method for the calculation of the dynamics of compressible fluids, Los Alamos Scientific Laboratory Report LA-3466, 1966.

[10] R.A. Gingold, J.J. Monaghan, Kernel estimates as a basis for general particle, methods in hydrodynamics, J. Comput. Phys. 46 (1982) 429.

[11] S. Koshizuka, K. Okamoto, K. Furuta, Development of computational techniques for nuclear engineering, Prog. Nucl. Energy 32 (1/2) (1998) 209.

[12] S. Koshizuka, A. Nobe, Y. Oka, Numerical analysis of breaking waves using the moving particle semi-implicit method, Int. J. Numer. Methods Fluids 26 (1998) 751.

[13] K. Nomura, S. Koshizuka, Y. Oka, H. Obata, Numerical analysis of droplet breakup behavior using particle method, J. Nucl. Sci. Technol. 38 (12) (2001) 1057.

[14] H.Y. Yoon, S. Koshizuka, Y. Oka, Direct calculation of bubble growth, departure, and rise in nucleate pool boiling, Int. J. Multiphase Flow 27 (2001) 277.

[15] S. Heo, S. Koshizuka, Y. Oka, Numerical analysis of boiling on high heat-flux and high subcooling condition using MPS-MAFL, Int. J. Heat Mass Transfer 45 (2002) 2633.

[16] H. Ikeda, S. Koshizuka, Y. Oka, H.S. Park, J. Sugimoto, Numerical analysis of jet injection behavior for fuel-coolant interaction using particle method, J. Nucl. Sci. Technol. 38 (3) (2001) 174.

[17] S. Koshizuka, Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nucl. Sci. Eng. 123 (1996) 421.

[18] S. Koshizuka, H. Ikeda, Yoshiaki Oka, Numerical analysis of fragmentation mechanisms in vapor explosions, Nucl. Eng. Des. 189 (1999) 423.

[19] Y. Chikazawa, S. Koshizuka, Y. Oka, A particle method for elastic and visco-plastic structure and fluid–structure interactions, Comput. Mech. 27 (2001) 97.

[20] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335.

[21] D.E. Fyfe, E.S. Oran, M.J. Fritts, Surface tension ad viscosity with Lagrangian Hydrodynamics on a triangular mesh, J. Comput. Phys. 76 (1988) 349.

[22] P.F. Velazquez, D.O. Gomez, G.M. Dubner, G. Gimenez de Castro, A. Costa, Study of the Rayleigh–Taylor instability in Tycho's supernova remnant, Astron. Astrophys. 334 (1998) 1060.

[23] I.P.D. De Silva, H.J.S. Fernando, F. Eaton, D. Hebert, Evolution of Kelvin–Helmholtz billows in nature and laboratory, Earth Planet. Sci. Lett. 143 (1996) 217.