PERGAMON

International Journal of Heat and Mass Transfer 45 (2002) 2633–2642

International Journal of HEAT and MASS TRANSFER

www.elsevier.com/locate/ijhmt

Numerical analysis of boiling on high heat-flux and high subcooling condition using MPS-MAFL

S. Heo *, S. Koshizuka, Y. Oka

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

Received 8 November 2001; received in revised form 20 December 2001

Abstract

This paper shows the numerical simulation study on the growth of the bubble in the transient pool boiling using moving particle semi-implicit with meshless advection using flow-directional local grid (MPS-MAFL) method. The growth process of a bubble with the different initial radii is calculated in a high heat-flux and high subcooling condition. The smaller initial radius is, the earlier the growth starts. The initial bubble radius has little effect on the growth initiation time and the bubble departure radius.

1. Introduction

When the control rod is unlikely to be withdrawn from the reactor core of nuclear power plant (NPP), a large reactivity insertion increases the nuclear reaction rate and the core power. Negative reactivity can be added from Doppler effect in the fuel and void generation in the reactor coolant. This is called a reactivity initiated accident (RIA).

Void formation in the coolant is a very important factor in the safety analysis of the RIA but the current licensing calculations neglect void formation because it is difficult to predict the time and the amount of void during a rapid transition condition.

A number of experimental studies are found in the literature. Iida et al. [1,2] have carried out experiments of transient boiling in high heat flux conditions for various fluids. The void formations during RIA under cold shutdown and hot standby conditions was studied by Yamada et al. [3] and Minato et al. [4]. Sugiyama et al. [5] investigated high burnup fuel behavior under RIA and Risher et al. [6] simulated the behavior of reactor core under RIA using transient three-dimensional computer code SPNOVA.

There were many efforts to simulate boiling phenomena using numerical method. Welch [7] simulated vapor bubble growth using a moving grid to trace the phase interface with the finite volume method, but it is limited to cases where the deformation of the phase interface was relatively small. Juric and Tryggvason [8] calculated a vapor shape in film boiling using a front tracking/finite difference method. Yoon [9] and Yoon et al. [10] studied sloshing of free surface, bubble rise in viscous liquids and bubble behavior in the nucleate pool boiling using moving particle semi-implicit with meshless advection using flow-directional local grid (MPS-MAFL) method. The accuracy of MPS-MAFL was tested through the calculation of natural convection in a square cavity [12].

In the present paper, MPS-MAFL method was applied to the simulation of nucleation boiling where the heat flux is very high and the coolant is highly subcooled. Bubble behavior and void fraction are calculated for the case of RIA under the cold shutdown condition of boiling water reactor (BWR) core and the results are compared with the experiment of Yamada et al.

* Corresponding author. Tel.: +81-29-287-8443; fax: +81-29-287-8488.

E-mail addresses: sunny@tokai.t.u-tokyo.ac.jp (S. Heo), koshi@tokai.t.u-tokyo.ac.jp (S. Koshizuka), oka@tokai.t.u-tokyo.ac.jp (Y. Oka).

0017-9310/02/$ - see front matter
PII: S0017-9310(02)00011-X

Nomenclature

A	area
d	number of space dimensions
g	gravity
n	number density
p	pressure
r	radius, distance
r_e	radius of interaction area
T	temperature
t	time (s)
Δt	time increment
$\vec{u}$	velocity vector
w	weight function
Greek symbols	
α	thermal diffusivity
μ	viscosity
ρ	density
σ	surface tension
Subscripts	
g	gas phase
l	liquid phase
a	arbitrary

2. Mathematical model of MPS-MAFL method

Moving particle semi-implicit (MPS) method is a particle method where thermal hydraulic problems are solved by particle interactions without the aid of grids. Convection terms are not necessary to calculate because of fully Lagrangian description. In the MPS method, differential operators, such as gradient and Laplacian, in the governing equations are modeled as particle interactions. A particle interacts with its neighbors in the interaction area covered with a weight function. The meshless advection using flow-directional local grid (MAFL) method is a gridless method developed for the calculation of convection. MPS-MAFL enables us arbitrary Lagrangian–Eulerian calculation.

In MPS-MAFL method, the calculation procedure consists of three phases: Lagrangian, re-configuration and convection (Eulerian). In the Lagrangian phase, the governing equations are solved using the particle interaction models of MPS and the computing points are moved in Lagrangian description. Next, the computing points are re-configured by considering fixed boundaries and moving interfaces. The physical quantities at new-time coordinates are interpolated through a convective (Eulerian) calculation using MAFL method.

2.1. Governing equations

The continuity, Navier–Stokes and energy equations for incompressible viscous flows are:

$$\nabla \cdot \vec{u} = 0, \tag{1}$$

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} - \vec{u}^{c}) \cdot \nabla \vec{u}\right) = -\nabla p + \mu \nabla^2 \vec{u} + \sigma \vec{\kappa} \cdot \vec{n} + \rho \vec{g} \tag{2}$$

and

$$\frac{\partial T}{\partial t} + \left(\vec{u} - \vec{u}^{c}\right) \cdot \nabla T = \alpha \nabla^2 T, \tag{3}$$

where $\vec{n}$ is the unit normal vector to the interface. In the convection terms of Eqs. (2) and (3), $\vec{u}$ and $\vec{u}^{c}$ represent the fluid velocity and the motion of a computing point that can be adaptively configured, respectively. An arbitrary calculation is allowed between fully Lagrangian calculation ($\vec{u}^{c} = \vec{u}$) and Eulerian calculation ($\vec{u}^{c} = 0$) so that a sharp fluid front is traced accurately by moving the computing points in Lagrangian coordinates while the fixed boundaries are described with Eulerian coordinates.

2.2. Numerical scheme

The calculation procedure of MPS-MAFL method consists of three phases: Lagrangian, re-configuration, and convection (Eulerian).

In the Lagrangian phase, the right-hand side of Navier–Stokes equation (Eq. (2)) is solved explicitly and the temporal velocity, $\vec{u}^{*}$, is obtained by

$$\vec{u}^{*} = \vec{u}^{n} + \frac{\Delta t}{\rho}\left[\mu \nabla^2 \vec{u}^{n} + \sigma(\kappa \cdot \vec{n})^{n} + \rho g\right], \tag{4}$$

where superscript n refers to the time step. The temporal location of the computing point becomes

$$\vec{r}^{*} = \vec{r}^{n} + \vec{u}^{*} \Delta t. \tag{5}$$

The pressure is calculated implicitly using Poisson's equation

$$\nabla^2 P^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \vec{u}^{*}. \tag{6}$$

The temporal velocity and coordinates are updated by the pressure gradient term as

$$\frac{\vec{u}^{L} - \vec{u}^{*}}{\Delta t} = -\frac{1}{\rho} \nabla P^{n+1} \tag{7}$$

and the position of the computing point is updated by

$$\vec{r}^{L} = \vec{r}^{n} + \vec{u}^{L} \Delta t, \tag{8}$$

where superscript 'L' stands for Lagrangian description. The temperature at the new position of the computing point, T^{L}, is obtained by solving the energy equation (Eq. (3)) explicitly as

$$T^{\mathrm{L}} = T^n + \alpha\nabla^2 T^n \cdot \Delta t. \tag{9}$$

The particle interaction models of the MPS method, which will be presented in the next section, are applied to the differential operators, such as gradient, Laplacian and divergence, in Eqs. (4), (6), (7) and (9).

In the re-configuration phase, the position of computing point at new-time step, $\vec{r}^{n+1}$, is determined arbitrarily and the velocity of the computing point, $\vec{u}^{\mathrm{c}}$, is given by

$$\vec{u}^{\mathrm{c}} = \frac{\vec{r}^{n+1} - \vec{r}^{n}}{\Delta t}. \tag{10}$$

From Eqs. (8) and (10), the arbitrary convection velocity $\vec{u}^{\mathrm{a}}$ ($\equiv \vec{u}^{\mathrm{L}} - \vec{u}^{\mathrm{c}}$) becomes

$$\vec{u}^{\mathrm{a}} = \frac{\vec{r}^{\mathrm{L}} - \vec{r}^{n}}{\Delta t} - \frac{\vec{r}^{n+1} - \vec{r}^{n}}{\Delta t} = -\frac{\vec{r}^{n+1} - \vec{r}^{\mathrm{L}}}{\Delta t}. \tag{11}$$

The position changes associated with the velocities, $\vec{u}^{\mathrm{L}}$, $\vec{u}^{\mathrm{c}}$ and $\vec{u}^{\mathrm{a}}$, which are illustrated in Fig. 1.

At the convection phase, the convection terms of Navier–Stokes and energy equations, Eqs. (2) and (3) are solved and velocity and temperature at the new time-step are obtained by means of interpolation of values at $\vec{r}^{\mathrm{L}} - \Delta t\vec{u}^{\mathrm{a}}$ from the profiles of $\vec{u}^{\mathrm{L}}$ and T^{L} as

$$\vec{u}^{n+1} \equiv \vec{u}\left(\vec{r}^{n+1}\right) = \vec{u}\left(\vec{r}^{\mathrm{L}} - \Delta t\vec{u}^{\mathrm{a}}\right) \tag{12}$$

and

$$T^{n+1} \equiv T\left(\vec{r}^{n+1}\right) = T\left(\vec{r}^{\mathrm{L}} - \Delta t\vec{u}^{\mathrm{a}}\right). \tag{13}$$

The gridless convection scheme MAFL is applied to Eqs. (12) and (13).

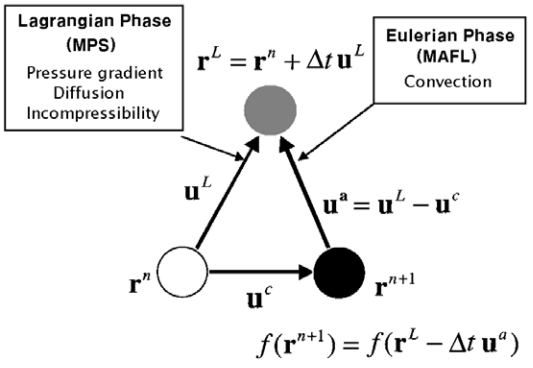

Fig. 1. Schematic diagram of numerical algorithm.

2.3. Lagrangian phase

Differential operators in the governing equations are calculated using the particle interaction models of MPS method. The gradient and Laplacian models were proposed by Koshizuka et al. [13] and the divergence model was developed for MPS-MAFL method by Yoon et al. [14]. In these models, a particle interacts with its neighboring particles with a weight function $w(r, r_{\mathrm{e}})$, where r is the distance between two particles and r_{e} is the radius of interaction area. The following function is employed in this study:

$$w(r, r_{\mathrm{e}}) = \begin{cases} -(2r/r_{\mathrm{e}})^2 + 2 & (0 \leqslant r < 0.5r_{\mathrm{e}}), \\ (2r/r_{\mathrm{e}} - 2)^2 & (0.5r_{\mathrm{e}} \leqslant r < r_{\mathrm{e}}), \\ 0 & (r \geqslant r_{\mathrm{e}}). \end{cases} \tag{14}$$

Since the interaction area of the weight function is bounded by r_{e}, a particle interacts with a finite number of neighboring particles in $r < r_{\mathrm{e}}$. The radius of the interaction area varies in space and time so that the number of particles within r_{e} is kept constant.

The differential operator vector ∇ in the pressure gradient term of Navier–Stokes equation, Eq. (2) is expressed in terms of scalar quantities of each particle with the weight function of Eq. (14). A gradient vector between two particles i and j possessing scalar quantities ϕ_i and ϕ_j at $\vec{r}_i$ and $\vec{r}_j$ is simply defined by $(\phi_j - \phi_i)(\vec{r}_j - \vec{r}_i)/|\vec{r}_j - \vec{r}_i|^2$. The gradient vector at the particle i is given as the weighted average of the gradient vectors between the particle i and its neighboring particles j,

$$\langle\nabla\phi\rangle_i = \frac{d}{n_i}\sum_{j\neq i}\left[\frac{\phi_j - \phi_i}{|\vec{r}_j - \vec{r}_i|^2}\left(\vec{r}_j - \vec{r}_i\right)w\left(|\vec{r}_j - \vec{r}_i|, r_{\mathrm{e},ij}\right)\right], \tag{15}$$

where the normalization factor is defined by

$$n_i \equiv \sum_{j\neq i} w\left(|\vec{r}_j - \vec{r}_i|, r_{\mathrm{e},ij}\right) \tag{16}$$

and $r_{\mathrm{e},ij} = (r_{\mathrm{e},i} + r_{\mathrm{e},j})/2$.

The Laplacian operator ∇^2 representing diffusion is modeled by distribution of a quantity from a particle to its neighboring particles by use of the weight function as follows:

$$\left\langle\nabla^2\phi\right\rangle_i = \frac{2d}{\lambda n_i}\sum_{j\neq i}\left[\left(\phi_j - \phi_i\right)w\left(|\vec{r}_j - \vec{r}_i|, r_{\mathrm{e},ij}\right)\right]. \tag{17}$$

By the central limit theorem, Eq. (17) converges to exact solution if λ is determined so that the variance increase, $\sigma^2(\Delta t)$, during Δt is equal to $2dv/\Delta t$ which is the theoretical variance increase of ϕ, where r is the diffusion coefficient.

$$\lambda = \sum_{j\neq i}\left[|\vec{r}_j - \vec{r}_i|^2 w\left(|\vec{r}_j - \vec{r}_i|, r_{\mathrm{e},ij}\right)\right]. \tag{18}$$

This Laplacian model is conservative because the quantity lost by a particle is just obtained by its neighboring particles.

The divergence operator is modeled like the gradient operator in Eq. (15). The velocity divergence between two particles i and j is defined by $(\vec{u}_j - \vec{u}_i) \cdot (\vec{r}_j - \vec{r}_i)/|\vec{r}_j - \vec{r}_i|^2$. The velocity divergence at the particle i is given by the weighted average of the individual velocity divergences.

$$\left\langle \nabla \cdot \vec{u} \right\rangle_i = \frac{d}{n_i} \sum_{j \neq i} \left[\frac{(\vec{u}_j - \vec{u}_i) \cdot (\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^2} w\left(|\vec{r}_j - \vec{r}_i|, r_{e,ij}\right) \right]. \tag{19}$$

The right-hand side of Eq. (6) is calculated using Eq. (19) and the left-hand side is calculated using the Laplacian model of Eq. (17). Then we get simultaneous equations expressed by a linear symmetric matrix. This matrix equation is solved by the incomplete Cholesky conjugate gradient (ICCG) method. Since this incompressibility model is based on the velocity divergence, the particle number density need not be constant and particles are allowed to be concentrated locally for higher resolution.

2.4. Re-configuration phase

The fully Lagrangian method is more complicated for describing inlet and outlet flow boundaries as well as irregular distribution of particles may occur by particle motion. Thus, the computing points, which are particles in Lagrangian context, are re-configurated.

After the locations of computing points are modified from $\vec{r}^n$ to $\vec{r}^L$ in Lagrangian phase, the computing points are redistributed considering the shape of boundaries. The computing points belonging to the fixed boundary and inlet or outlet boundary go back to their original position, $\vec{r}^n$, for a fully Eulerian calculation. The moving boundary can be traced through the Lagrangian motion of the computing points describing the free surface without calculating the convection term of the governing equation so that $\vec{r}^{n+1}_{\text{surface}} = \vec{r}^{L}_{\text{surface}}$. However, in practice, the computing points on the moving boundary are likely to be apart or cluster each other when the fully Lagrangian calculation is applied repeatedly for a number of time steps. To avoid this situation, the coordinates $\vec{r}^{n+1}_{\text{surface}}$ are adjusted to keep an equal distance between the computing points. After the coordinates of boundary point have been determined, the computing point of the inside region is configured considering the geometry of boundaries.

The re-configuration of computing points corresponds to the mesh re-generation in the conventional mesh-based method. Although it is somewhat laborious task to configure the computing points for the complex geometry, it is much easier in MPS-MAFL method because only computing points are located. The number of computing points need not be constant and can be spatially concentrated for the higher resolution.

2.5. Convection phase

MAFL method is proposed by Yoon et al. [14] for an accurate gridless calculation of convection. A multi-dimensional convection problem is regarded as a one-dimensional problem if a computational grid is generated along the flow direction. Considering the flow direction at each computing point ($\vec{u}^a$ in Eqs. (12) and (13)), a one-dimensional local grid is temporary generated as shown in Fig. 2 where Δr shows the distance between local grid points and $r_{e,i}$ is the radius of interpolation area. The positions of local grid points are denoted by $\langle \vec{r} \rangle_k$. Locations and the number of local grid points are determined based on the difference scheme that will be applied. For example, in Fig. 2, there are three local grid points, two in the upstream and the other in the downstream of the computing point, for the application of a second-order upwind scheme.

At the local grid points, the physical properties, $\langle f \rangle_k$, are interpolated from those of neighboring computing points, f_j^L, using a weight function as follows:

$$\langle f \rangle_k = \frac{\sum_j f_j^L w\left(|\vec{r}_j^L - \left\langle \vec{r} \right\rangle_k|, r_{e,k}\right)}{\sum_j w\left(|\vec{r}_j^L - \left\langle \vec{r} \right\rangle_k|, r_{e,k}\right)}, \quad k = -2, -1, 1. \tag{20}$$

The weight function in Eq. (14) is used. The interpolation region is limited by a circle of the radius of interpolation area, $r_{e,k}$ and the grid lies vertical to the

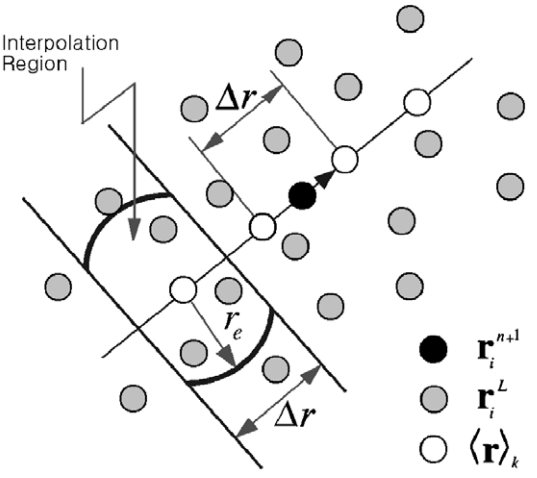

Fig. 2. Generation of local grid.

flow-directional local grid as shown in Fig. 2. If all the particles within the radius of $r_{e,k}$ are used for the interpolation, the interpolation regions will overlap each other and this will cause more numerical diffusion.

Any difference scheme can be applied easily since a one-dimensional gird has been employed along the flow direction. In this study, a first-order upwind scheme is applied as

$$f_i^{n+1} = f_i^{\mathrm{L}} - q\left(f_i^{\mathrm{L}} - \langle f\rangle_{-1}\right), \tag{21}$$

where $q = |\vec{u}^{\mathrm{a}}|\Delta t/\Delta r$. When the number and locations of the computing points are largely changed during the re-configuration phase, the profile of the convection velocity is severely distorted and a higher-order scheme is likely to cause numerical instability.

2.6. Surface tension model

For the analysis of two-phase interface problems where the curvature of the interface is large, a calculation model for the surface tension force is used. In general, the pressure difference between gas and liquid is

$$\Delta p = \sigma\kappa. \tag{22}$$

In the present method, a continuous curvilinear surface is obtained by connecting the computing points on phase interface. Then the curvature κ is calculated from the radius of circumference whose center is determined to be the intersection points of two perpendicular bisectors. It is showed in Fig. 3. This model is verified through the calculation of volume decrease of a gas bubble in stagnant liquid due to surface tension [9].

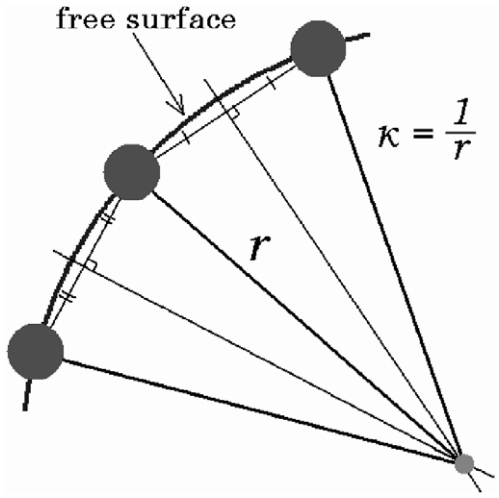

Fig. 3. Determination of curvature of curvilinear surface.

2.7. Coupling of energy and momentum equations

The energy and momentum equations of Eqs. (2) and (3) are solved with the following calculation steps:

(1) the diffusion term of the energy equation is calculated explicitly with the boundary condition, $T_{\text{bubble surface}} = T_{\text{sat}}(P_{\mathrm{g}})$,
(2) the evaporation rate and the amount of evaporation, Q, is determined by calculating the heat transfer rate at the phase interface,
(3) the new-time vapor pressure is guessed considering the change in the amount of vapor

$$P^* = P^n \frac{V^n + \Delta V}{V^n}, \tag{23}$$

where $\Delta V = Q/h_{\mathrm{fg}}\rho_{\mathrm{g}}$,
(4) momentum equation is solved and the coordinates, velocity vectors and temperatures are updated,
(5) for the new-time coordinates the vapor volume, V^{n+1}, is calculated, and
(6) the vapor pressure is modified considering the change in the vapor volume

$$P^{n+1} = P^* \frac{V^n + \Delta V}{V^{n+1}}. \tag{24}$$

The ideal gas law is applied for the calculation of the vapor pressure.

2.8. Boundary condition models

In a gridless mothod, the physical quantity at the computing points beyond the boundary $\langle f\rangle_{\text{out}}$ does not coincide with boundary value f_{BC}. When a strict condition is needed at the boundary, we have to solve a set of simultaneous equations for the quantities at the computing points beyond the boundary. Instead of solving a matrix, we assign boundary values explicitly to the computing points beyond the boundary so that the estimation at the boundary, $\langle f\rangle_{\mathrm{BC}}$, approximates f_{BC}. For example, in a free-slip condition, the velocity vector of an outside computing point is determined so that the component parallel to the wall is equal to that of the nearest inner point while the vertical component is zero. In a non-slip condition, the velocity distribution of the outside point is given to be opposite to that of the inner points.

The Neumann boundary condition is applied to the implicit calculation of pressure Poisson equation. This pressure boundary condition is easily implemented through the Laplacian model in Eq. (17). If there are N inside particles, we have N simultaneous linear equations expressed by $N \times N$ symmetric matrix where the coefficients of the matrix are given by Eq. (17). The Neumann boundary condition is satisfied if the pressure of a particle j beyond the boundary equals to that of

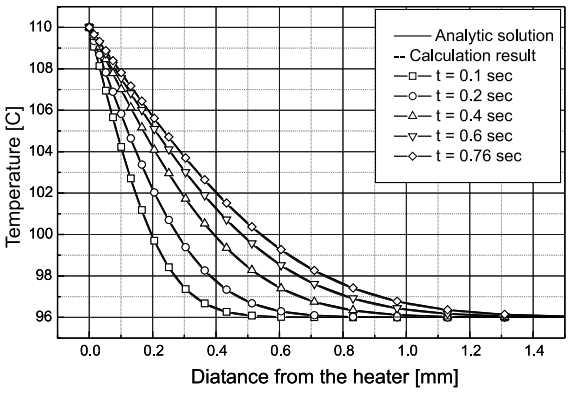

Fig. 4. Verification of boundary condition model (constant temperature case, $T_{\text{wall}} = 110$ °C and $T_{\text{bulk}} = 96$ °C).

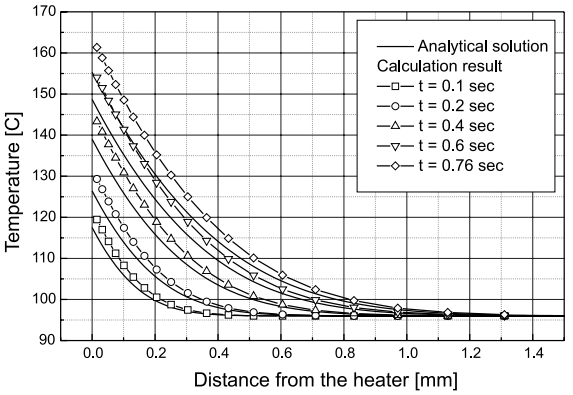

Fig. 5. Verification of boundary condition model (constant heat flux case, $q''_{\text{heater}} = 100$ kW/m^2 and $T_{\text{bulk}} = 96$ °C).

boundary particles, i.e., $P_j = P_i$. This may be approximated by an alternative expression of Eq. (17)

$$\langle \nabla^2 P \rangle_i = \frac{2d}{\lambda n_i} \sum_{j \neq i, j \neq \text{out}} \left[(P_j - P_i) w\left(|\vec{r}_j - \vec{r}_i|, r_{e,ij} \right) \right]. \quad (17a)$$

For the constant temperature boundary condition, the temperature of computing points beyond heater wall is assigned a value at the end of convection phase of each time step. For the constant heat flux case, it is determined by considering the temperature of the computing points near the heater and heat flux

$$\begin{aligned} T^{n+1}_{\text{heater}} &= \left\langle T^{n+1}_{\text{liquid}} \right\rangle + \langle \Delta T \rangle_{\text{heat flux}} \\ &= T^{n+1}_{\text{nearest}} + \frac{q''}{k} \Delta y_{\text{nearest}} \end{aligned} \quad (25)$$

or

$$T^{n+1}_{\text{heater}} = \frac{1}{n_i} \sum_{j=\text{liquid}} \left[\left(T^{n+1}_j + \frac{q''}{k} \Delta y_j \right) w\left(|\vec{r}_j - \vec{r}_i|, r_{e,ij} \right) \right]. \quad (25a)$$

The temperature profile without phase change nor flow motion was calculated using this model. The result is shown in Figs. 4 and 5 with that of heat conduction equation and two results show a good agreement.

3. Simulation condition

Simulation conditions are based on the experiment of Yamada et al., which considered the RIAs under cold shutdown condition. The simulation is x–y two-dimensional, while the experiment is three-dimensional. The effect of space dimension on the bubble growth was investigated [9] and no significant difference is observed between x–y and r–z two-dimensions. The heating wall is horizontal in the simulation though it is vertical in a rectangular pipe in the experiment. The effect of buoyancy is not large because the growth time of bubble is very short (shorter than 0.2 s).

The simulation conditions which are used in this study are summarized in Table 1. It is assumed that the initial shape of bubble is spherical and the contact angle between the vapor–liquid interface and wall is 45°.

The heat flux is 2 MW/m^2 which is the same as the experiment. The radius of an initial bubble is varied from 30×10^{-6} to 300×10^{-6} m. Very small bubbles are difficult to be handled because extremely many computing points are needed.

Fig. 6 shows the configuration of computing points. A space of 0.01×0.015 m^2 divided into four regions so that computing points are fitted to the liquid–vapor interface and are concentrated near the interface. In region (a), the computing points are fitted to the interface having a constant spacing. Region (b) surrounding region (a) is filled with the computing points in a rectangular array. The spacing of the computing points is wider as the distance from the interface increases. The average distance between the neighboring computing points, which is the growth ratio of the grid size in mesh-based methods, increases by the ratio of 1.15 for each layer. The computing points are uniformly distributed in region (c). In region (d), which is placed in the upper side of regions (b) and (c), horizontal layers of the computing points are configured by the growth ratio of 1.15.

Table 1
Initial and boundary conditions

Temperature of water	27 °C
Radius of bubble	30–300×10^{-6} m
Contact angle of bubble	45°
Temperature in the bubble	100 °C
Pressure of bubble	1 atm
System pressure	1 atm (at the top of water)
Heat flux at the heater wall	2 MW/m^2

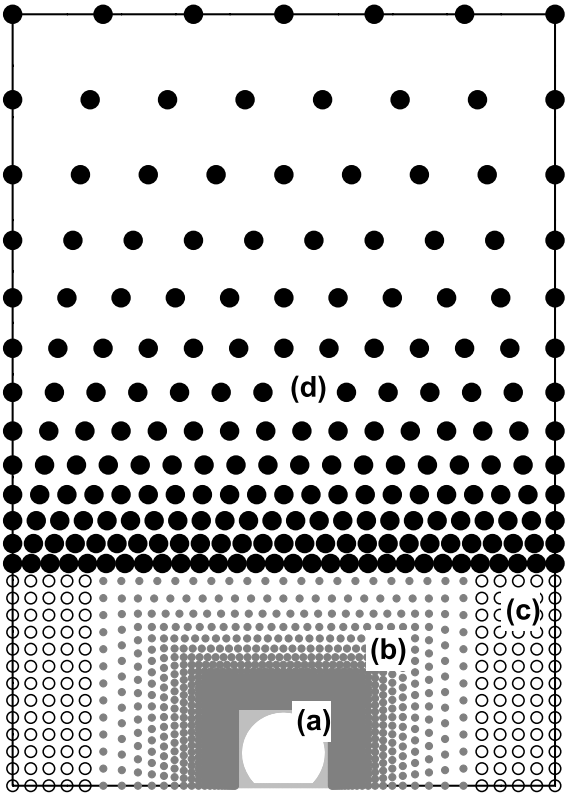

Fig. 6. Configuration of computing points.

The spacing between the computing points on the interface, which is the distance of computing points on the first layer of region (a), varies from 2×10^{-6} to 15×10^{-6} m at initial state depending on the radius of bubble. The spacing increases as the radius of bubble increases and it is about 60×10^{-6} m at the departure point of a bubble. The number of computing points at the bubble surface is about from 80 to 180 and the total number of computing points is from 2500 to 8000.

When the average temperature of water at the bubble surface is lower than the initial temperature of vapor, the heat flow at the bubble surface and the movement of computing points, that is, the movement of the interface between bubble and water is not calculated and it prevents the collapse of bubble.

A non-slip boundary condition is applied at the wall and the constant heat flux model (Eq. (25)) applied to the heater wall. The upwind scheme is used for the convection calculation.

4. Results and consideration

Fig. 7 shows the shape change of the bubble according to time. The initial radius of the bubble is 50×10^{-6} m and another conditions are same with what presented at Table 1. A dot presents the location of a computing point and the color of dot presents the temperature at the point. The white area of the center represents the vapor bubble.

At the start, water surrounding the bubble is subcooled at 27 °C. As time goes on, the average temperature of water around the bubble rises and reaches the saturation temperature. The net heat flux at the interface flows from water to the bubble and the bubble grows at 0.0162 s. Growth of the bubble is slow because of the low superheating of water, inertia of water and surface tension at the interface.

The bubble growth is accelerated with increase of the superheat of water. A larger bubble size also enhances the bubble growth speed due to reduction of surface tension. When the bubble becomes larger, however, the bubble is condensed by bulk water. Finally, the bubble departs from the heater wall, bubble is rapidly condensed. This explanation is confirmed by Figs. 8 and 9 showing the radius of the bubble and the net heat flux at the bubble surface, respectively.

To examine the effect of the initial radius of bubble nuclei, the bubble growth is searched for the several values of the initial radius and some results are shown in Fig. 10. It becomes clear that the bubble growth times taken from the start of bubble growth to the departure are almost same and the smaller bubble is, the earlier it starts to grow. These results agree with the observational results that many small bubbles cover the entire surface of heater in the case of high heat flux and high subcooling while large bubbles are often observed in the case of low heat flux and low subcooling. When a heat flux is very high and liquid is highly subcooled, the temperature difference between the region nearest heater and the other part of liquid is very high and the thickness of thermal layer is very small. The nearer region to heater the bubble is located on, the more energy it receives from the heater and the less energy it looses at the upper part of the bubble surface, therefore the nucleus of the smaller radius grows earlier than larger nucleus.

The calculation result is compared with the experimental data of Yamada et al. The volume of bubble was estimated from the shape of bubble surface in two-dimension using the volume integral method as shown in Fig. 11. It is assumed that the shape of the bubble is symmetric to the center line. A nucleation site density is necessary to get a void fraction, but there is no information about the active nucleation site density under the present condition. It is assumed that the pitch between the nucleation sites is about four times the departure diameter of the bubble, then the nucleation site density is 4000/m^2. The void fraction of this simulation is shown in Fig. 12 with that of Yamada et al. and these show good agreement with each other.

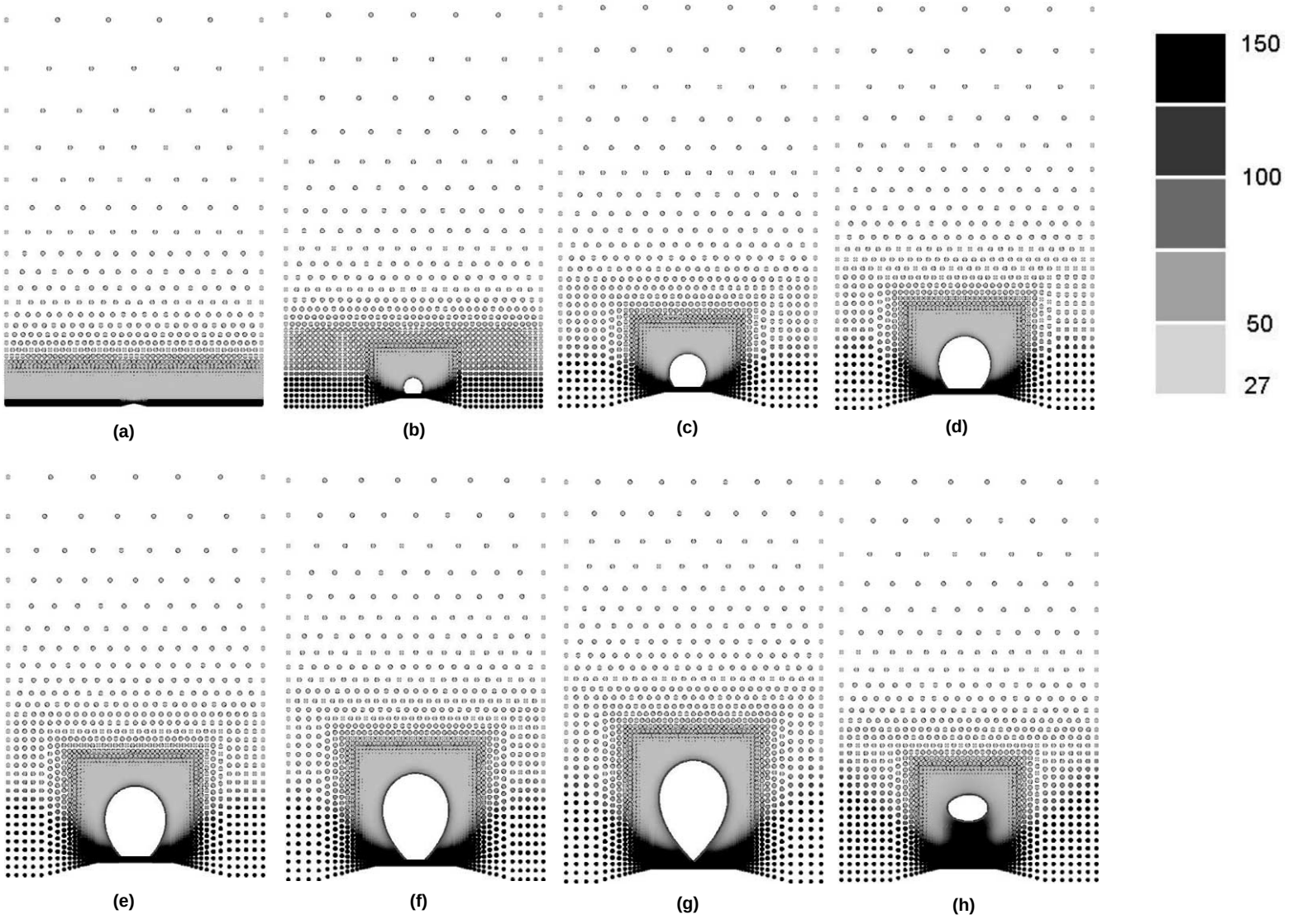

Fig. 7. Sequence of bubble growth: (a) 0.0162 s; (b) 0.0362 s; (c) 0.0562 s; (d) 0.0762 s; (e) 0.0962 s; (f) 0.1162 s; (g) 0.1363 s; (h) 0.1463 s.

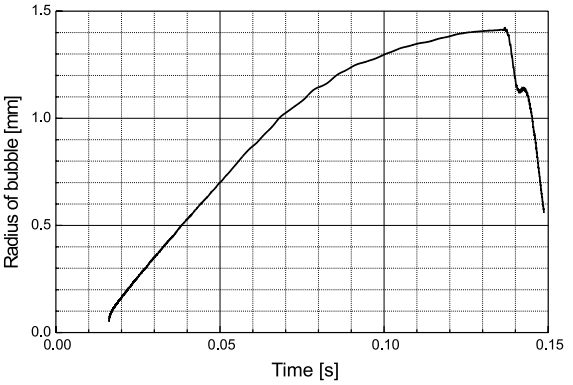

Fig. 8. Radius of vapor bubble.

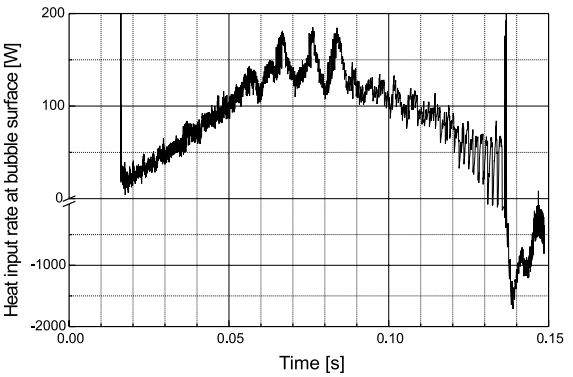

Fig. 9. Total heat flow at the bubble surface.

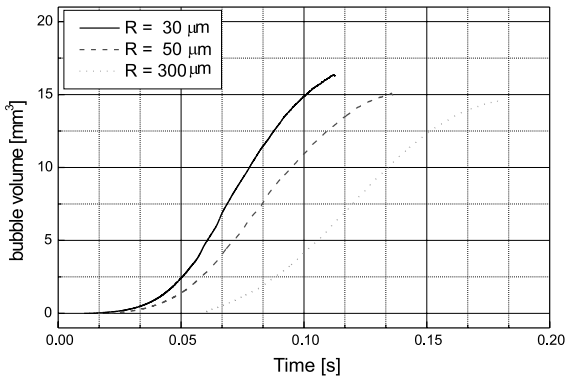

Fig. 10. Effect of the initial bubble radius.

It is important that when the vapor starts to grow and how much time will be taken before the amount of void generation reaches the amount whose reactivity is equivalent to the reactivity of one control rod at the moment of RIA in reactor core. The shape of void fraction of this study is very similar to that of Yamada's

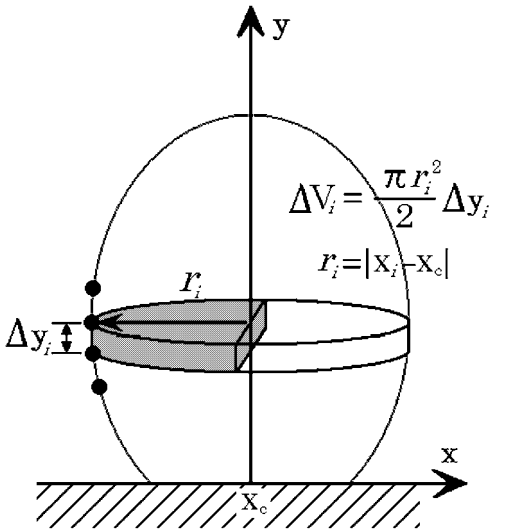

Fig. 11. Volume element of a computing cell.

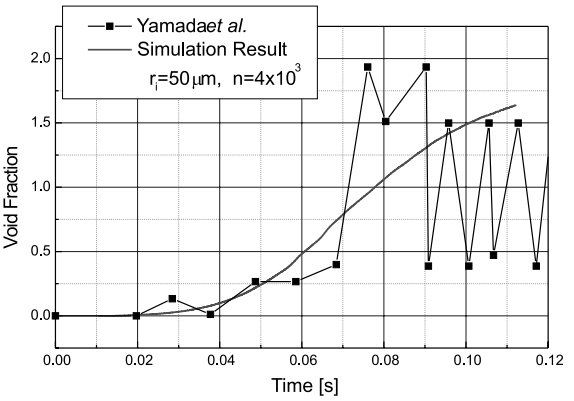

Fig. 12. Comparison of void fraction with that of Yamada's experiment.

experiment during the nucleate and transient boiling regime.

5. Conclusions

The growth of the bubble on the condition of high heat flux and high subcooling such as RIAs at the NPP was studied and was simulated using MPS-MAFL method. The growth process of bubbles with the different initial radius is calculated and it becomes clear that as smaller initial radius is, as earlier the growth starts. The initial bubble radius has little effect on the growth initiation time and the bubble departure radius. The calculated void fraction was compared with that of the experiment of Yamada et al. and it becomes certain that

the MPS-MAFL code well-estimates the void fraction on the high heat flux and high subcooling condition.

References

[1] Y. Iida, K. Okuyama, K. Sakurai, Peculiar bubble generation on a film heater submerged in ethyl alcohol and imposed a high heating rate over 10^7 K/s, Int. J. Heat Mass Transfer 36 (1993) 2699–2701.

[2] Y. Iida, K. Okuyama, K. Sakurai, Boiling nucleation on a very small film heater subjected to extremely rapid heating, Int. J. Heat Mass Transfer 37 (1994) 2771–2780.

[3] T. Yamada, F. Kato, H. Asaka, Y. Kukita, Transient boiling and void formation during postulated reactivity-initiated accident in BWR: Experimental simulation, in: Proceedings of the second Japan–Korea Symposium on Nuclear Thermal Hydraulics and Safety, Fukuoka, Japan, October 2000, 2000, pp. 196–203.

[4] A. Minato, M. Abe, K. Nishida, K. Moriya, Y. Masuhara, Experimental of subcooled boiling by high heat flux in a short period, in: International Workshop on Current Status and Future Directions of Boiling Heat Transfer and Two-phase Flow, Osaka, Japan, October 2000.

[5] T. Sugiyama, T. Fuketa, K. Ishijima, Mechanical energy generation during high burnup fuel failure under reactivity initiated accident conditions, in: Proceedings of the seventh International Conference on Nuclear Engineering, Tokyo, Japan, April 1999.

[6] D.H. Risher, U. Bachrach, B.J. Johansen, R. Milanova, S. Stelletta, M. Subra, Advanced methods for analysis of the reactivity insertion accident for pressurized water reactors, in: Proceedings of the fifth International Conference on Nuclear Engineering, Nice, France, May 1997.

[7] S.W.J. Welch, Direct simulation of vapor bubble growth, Int. J. Heat Mass Transfer 41 (1998) 1655–1666.

[8] D. Juric, G. Tryggvason, Computations of boiling flows, Int. J. Multiphase flow 24 (1998) 387–410.

[9] H.Y. Yoon, A particle-gridless hybrid method for gas–liquid two-phase flows, Submitted in partial fulfillment of the requirement for the Ph.D., University of Tokyo, 1999.

[10] 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) 277–298.

[12] S. Koshizuka, H.Y. Yoon, D. Yamashita, Y. Oka, Numerical analysis of natural convection in a square cavity using MPS-MAFL, Comput. Fluid Dynamics J. 8 (2000) 485–494.

[13] S. Koshizuka, H. Tamako, Y. Oka, A particle method for incompressible viscous flow with fluid fragmentation, Comput. Fluid Dynamics J. 4 (1996) 29.

[14] H.Y. Yoon, S. Koshizuka, Y. Oka, A particle-gridless hybrid method for incompressible flows, Int. J. Numer. Meth. Fluid 30 (1998) 407–424.