

# Height function interface reconstruction algorithm for the simulation of boiling flows

M. Magnini & B. Pulvirenti

*Dipartimento di Ingegneria Energetica,  
Nucleare e del Controllo Ambientale, Università di Bologna, Italy*

## Abstract

In this paper we present our results on numerical study of vapor bubbles growing in quiescent superheated liquid, as effect of liquid evaporation at the interface. Height Function interface reconstruction algorithm is coupled with an evaporation model based on continuum field representation of source terms. The flow solver is a finite-volume CFD code. Interface is tracked within a Volume-Of-Fluid framework. Continuum-Surface-Force method accounts for surface tension effects. Vapor bubble heat-transfer-controlled growth is simulated for three different working fluids: water, HFE-7100 and R134a. Accuracy of interface reconstruction algorithm is of maximum importance. Unbalance between pressure gradients and surface tension forces at interface leads to the growth of an unphysical velocity field which switches original only diffusive heat transfer mechanism to combined diffusive-convective one. Height Function algorithm reduces the magnitude of this unreal velocity field. Standard test cases are considered to assess the performances of implemented version, through comparison with the widely used Youngs algorithm.

*Keywords: volume of fluid, height function, evaporation, surface tension, bubbles.*

## 1 Introduction

Modern numerical implementation of surface tension driven flows, with phase change, begins with the work of Juric and Tryggvason [1]. Earlier methods used a Lagrangian approach with a moving mesh following the interface. The limit of these methods was that they could not manage large interface deformation. Juric and Tryggvason developed a Front Tracking algorithm where Lagrangian evolution of interface is tracked on a fixed grid. Welch and Wilson [2] and Son



and Dhur [3] reformulated two full Eulerian methods, called Volume-Of-Fluid (VOF) and Level-Set (LS), in order to take into account for phase change. Both methods solve a scalar field conservation equation, volume fraction for VOF and level-set function for LS. This scalar field is used to reconstruct interface shape and position. Level-set function varies smoothly through interface, giving an accurate interface shape approximation, but mass conservation is not preserved. On the contrary, Volume-Of-Fluid method preserves mass conservation, but volume fraction field varies sharply through interface, leading to poor quality interface representation. Several algorithms have been developed in order to give better interface representation on Volume-Of-Fluid background. An important issue in the simulation of boiling flows is the thermal boundary condition at the interface. Most of the methods impose interface temperature at equilibrium saturation temperature corresponding to system pressure, but different conditions may be taken at interface. A detailed review can be found in [1].

The aim of the present work is to model a two-phase flow with phase change by means of a finite-volume CFD code. VOF method is used to follow interface. Two different methods to reconstruct interface are compared: Youngs method [4] and Height Function (HF) method [5–7]. The comparison is carried out analyzing algorithms performances through standard test cases. An evaporation model based on continuum-field representation of source terms developed by Hardt and Wondra [8] is implemented. Finally, HF and evaporation models are employed to simulate the growth of a vapor bubble in quiescent superheated liquid.

## 2 Governing equations and numerical model

In the VOF method, a volume fraction  $\alpha$  is defined for every domain cell. It represents the fraction of volume cell occupied by liquid, and it can be thought as the integral of an indicator function  $I(\mathbf{x}, t)$  over the cell of volume  $V$ :

$$\alpha = \frac{1}{V} \int_V I(\mathbf{x}, t) dV \quad (1)$$

The indicator function  $I(\mathbf{x}, t)$  is a multidimensional Heavyside step function with value 1 in liquid phase and 0 in vapor phase. Interface cells can be located by those cells with volume fraction included between 0 and 1. Generic fluid property  $\Phi$  for every domain cell can be expressed in term of  $\alpha$  as follows:

$$\Phi = \Phi_v + (\Phi_l - \Phi_v)\alpha \quad (2)$$

where  $\Phi_v$  and  $\Phi_l$  are vapor and liquid specific property. Since in this work both phases are always considered incompressible, the substitution of Eq. (2) in continuity equation yields the following volume fraction conservation equation:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = \frac{S_\alpha}{\rho_l} \quad (3)$$

where  $S_\alpha$  represents mass source term due to evaporation or condensation. Through Eq. (3) interface position is advected at every time step. Momentum



conservation equation for a surface tension driven laminar flow, with constant surface tension  $\sigma$ , has the following form:

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot \mathbf{u}) = -\nabla p + \nabla \cdot \left[ \mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right] + \rho \mathbf{g} + \mathbf{F}_\sigma \quad (4)$$

where  $\mathbf{F}_\sigma$  represents surface tension force. It is modelled through Continuum-Surface-Force (CSF) method developed by Brackbill *et al.* [9]. By CSF method surface tension force  $\mathbf{F}_\sigma$  can be expressed as:

$$\mathbf{F}_\sigma = \sigma \kappa \nabla \alpha \quad (5)$$

where  $\kappa$  stands for interface curvature. Therefore surface tension, a surface force, is modeled as a volume force, localized on a finite thickness interface. Interface curvature  $\kappa$  has to be computed by the interface reconstruction algorithm, in this case Youngs or HF method. The flow equations set is completed by energy conservation equation:

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla \cdot (\rho c_p \mathbf{u} T) = \nabla \cdot (\lambda \nabla T) + S_E \quad (6)$$

where  $S_E$  represents energy source term due to evaporation or condensation. Viscous heating is neglected. The task of the evaporation model is to compute correct mass and energy source terms, giving fields localized on interface cells.

Governing equations are discretized within a finite-volume framework. Volume fraction conservation equation is discretized first order in time through explicit scheme. PLIC technique by Youngs [4] is employed to compute mass transport across interface cell sides. Implicit first order scheme is used to solve momentum and energy conservation equations. Convective terms are discretized using a third order MUSCL scheme [10]. Diffusive terms are always discretized with central finite-difference scheme. For pressure-velocity coupling, PISO algorithm [11] is used.

### 3 Evaporation model

The Hardt and Wondra evaporation model [8] is implemented in order to evaluate mass and energy source terms on mass and energy equations. The model allows local interfacial temperature to deviate from saturation condition and local mass source is evaluated proportional to this interface superheating. According to Juric and Tryggvason [1], different interface temperature conditions may be taken, in order to account for entropy production due to phase change and for pressure rising due to curved interfaces. For the sake of simplicity, without loss of accuracy, entropy generation and non-equilibrium thermodynamics terms are neglected. Within these assumptions, a mass source equation was derived by Schrage study [12] of interface mass transfer. In order to express local interfacial mass flux  $\dot{m}$ , Schrage introduced the concept of local interfacial convective coefficient  $h_i$ ,



expressed as:

$$h_i = \frac{2\beta}{2 - \beta} \frac{h_{lv}^2}{\sqrt{2\pi R}} \frac{\rho_v}{T_{sat}^{3/2}} \quad (7)$$

in which  $R$  is the gas constant,  $h_{lv}$  is vaporization latent heat and  $\beta$  is a constant called accommodation coefficient, for which details can be found in [12]. Local interfacial convective coefficient  $h_i$  is connected to local interfacial mass flux as follows:

$$\dot{m} = \frac{h_i}{h_{lv}} (T_i - T_{sat}) \quad (8)$$

where  $T_i$  is the interfacial temperature. An initial mass source field is evaluated on interfacial cells, where evaporation takes place, by Eq. (8). Such a localized source term may lead to numerical instabilities. To avoid instabilities, a diffusion equation in which initial field represents initial condition is solved. The effect is to smear the initial field over some cells. Diffused mass source field is then used to evaluate mass and energy source terms. Hardt and Wondra [8] demonstrated that this evaporation model leads to correct evaporation rate, by comparing their results with some benchmarks.

## 4 Evaluation of surface tension effects: HF and Youngs comparison

The task of the reconstruction algorithm is to reconstruct interface geometry, meaning normal vector  $\mathbf{n}$  and curvature, which is implicit in volume fraction field. VOF based algorithms compute local curvature through derivatives of a chosen color function  $c$ , by the following relation:

$$\kappa = -\nabla \cdot \mathbf{n} = -\nabla \cdot \frac{\nabla c}{|\nabla c|} \quad (9)$$

where the first equation derives from geometrical considerations, see [9] for reference. Youngs [4] is one of the widest reconstruction algorithm used on VOF background, for this reason it was chosen as comparison. It works also with unstructured grids. It employs volume fraction  $\alpha$  as color function to compute curvature, but volume fraction has a very sharp variation through interface, leading to poor accuracy computation. Height Function method derives a smoother color function field through local integration of volume fraction field, then derivatives are more accurate. It only works with Cartesian grids. The algorithm implemented is a merge of Malik *et al.* [6] and Hernandez *et al.* [5] schemes. It follows a brief summary of the algorithm:

1. For each interface cell, volume fraction field on a local stencil is considered. Stencil orientation, vertical or horizontal, depends on interface orientation, determined by normal vector direction, computed as  $\mathbf{n} = \nabla \alpha / |\nabla \alpha|$ ;
2. Stencil extension, from  $3 \times 3$  up to  $7 \times 3$  cells, is chosen depending on volume fraction field variation, width is always 3 cells;



3. Stencil volume fraction field is adjusted to obtain a monotonic variation along height direction, in order to avoid errors when more than one interface cuts the stencil;
4. Volume fractions are summed columnwise, obtaining a local discrete height function  $H$  field;
5. If central column  $H$  value is not included on the cell for which curvature is being computed, curvature is taken equal to adjacent cell for which this condition is satisfied. Otherwise, curvature is computed through the extended form of Eq. (9):

$$\kappa = -\frac{H^{(2)}}{[1 + (H^{(1)})^2]^{3/2}} \quad (10)$$

where  $H^{(1)}$  and  $H^{(2)}$  denote first and second order derivatives of height function, computed through central difference scheme.

For axisymmetric simulations, curvature  $\kappa$  in Eq. (5) refers to total curvature, obtained summing principal curvatures *radii*  $\kappa_1$  and  $\kappa_2$ . On a cylindrical coordinate reference frame  $(r, \phi, z)$ ,  $\kappa_1$  denotes interface curvature on  $r - z$  plane, computed as reported from 1 to 5 steps. Curvature  $\kappa_2$  refers to second principal curvature of the surface obtained through revolution of interface around  $z$  axis. It lies on a plane normal to interface and perpendicular to  $r - z$  plane. Denoting interface profile in  $r - z$  plane as  $f(z)$ , second curvature can be expressed as:

$$\kappa_2 = -\left(\frac{f'(z)}{|f'(z)|}\right) \frac{1}{f(z)[1 + (f'(z))^2]^{1/2}} \quad (11)$$

The discrete version of first order derivative of interface profile  $f'(z)$  follows from HF algorithm, while discrete interface position  $f(z)$  can be computed for each interface cell, given cell volume fraction and normal vector.

#### 4.1 Test case 1: inviscid static droplet

A circular droplet of radius  $R$  is centered on a  $L = 4R$  side square domain. Different uniform mesh sizes  $\Delta x$  are tested, in order to check convergence rate of the methods. The coarsest mesh has  $20 \times 20$  elements, with  $R/\Delta x = 5$ . The most refined mesh has  $160 \times 160$  elements, with  $R/\Delta x = 40$ . Surface tension and phases' densities are set as unity. Viscous and gravity effects are neglected. Pressure is given at all boundaries as boundary condition. For such a flow, momentum equation reduces to the form:

$$\frac{D(\rho \mathbf{u})}{Dt} = -\nabla p + \sigma \kappa \nabla c \quad (12)$$

If pressure gradient is balanced by surface tension force, left hand side term is zero at each time step. Any unbalance leads to the growth of an unphysical velocity field defined "spurious velocity" or "parasitic current". Due to the absence of



external forces, the accuracy of the surface tension model and curvature calculation algorithms is attested by the comparison of unphysical velocity fields magnitude, in terms of the following non-dimensional error norms:

$$L_2(|\mathbf{u}^*|) = \frac{1}{U} \sqrt{\sum_{i=1}^n \frac{|\mathbf{u}_i|^2}{n}} \quad (13)$$

$$L_\infty(|\mathbf{u}^*|) = \frac{1}{U} \cdot \max(|\mathbf{u}_i|) \quad \text{for } i = 1, \dots, n \quad (14)$$

where  $n$  is the number of domain cells and  $U$  is a velocity scale defined as  $U = (\sigma/2\rho R)^{1/2}$ . Velocity error norms convergence history is reported in Fig. 1. Youngs method results do not converge with mesh refinement. HF results show second order convergence for  $R/\Delta x \leq 10$ , then for higher resolutions convergence order is in the range  $[1, 2]$ . Simulation time step is hold constant at  $\Delta t = 5 \cdot 10^{-7}$ , while capillary time step constraint decreases as mesh is refined. For  $R/\Delta x > 10$  it gets too close to simulation time step, yielding a worsening of convergence rate. The following non-dimensional pressure jump error norm is defined:

$$L_2(\Delta p^*) = \frac{1}{\Delta p_{ex}} \sqrt{\sum_{i=1}^m \frac{(\Delta p_i - \Delta p_{ex})^2}{m}} \quad (15)$$

where  $m$  is the number of interior droplet cells and  $\Delta p_{ex} = \sigma/R$  is the exact value of pressure jump across the interface. Pressure convergence history is reported in Fig. 1. Youngs results show convergence (first order) only for  $R/\Delta x \leq 10$ . HF error norm shows convergence rate in the range  $[1, 2]$  for all mesh sizes.

#### 4.2 Test case 2: isothermal bubble rising in viscous liquid

The second test case is the simulation of a gas bubble rising in a viscous liquid due to buoyancy forces. The effect of the combination of inertial, viscous and surface forces on bubble numerical terminal shape and velocity are compared with Bhaga and Weber experimental results [13]. Bhaga and Weber performed several test cases with air bubbles rising in a quiet water-sugar solution. Acting on sugar concentration, they could vary liquid density and viscosity to span a wide range of Eötvös (E) and Morton (M) numbers, obtaining different bubble shapes and terminal velocities, expressed as bubble Reynolds (Re) number. Surface tension variations were negligible. Non-dimensional numbers are defined as follows:

$$E = \frac{\rho_l g D^2}{\sigma}, \quad M = \frac{g \mu^4}{\rho_l \sigma^3}, \quad Re = \frac{\rho_l U D}{\mu_l}$$

where  $D$  is bubble initial diameter and  $U$  is terminal bubble velocity, when the rise reaches a steady state. Numerical and experimental results are compared for 4 different cases characterized by the same Eötvös number. Variation of Morton number was possible in the simulations by changing only liquid viscosity. It was

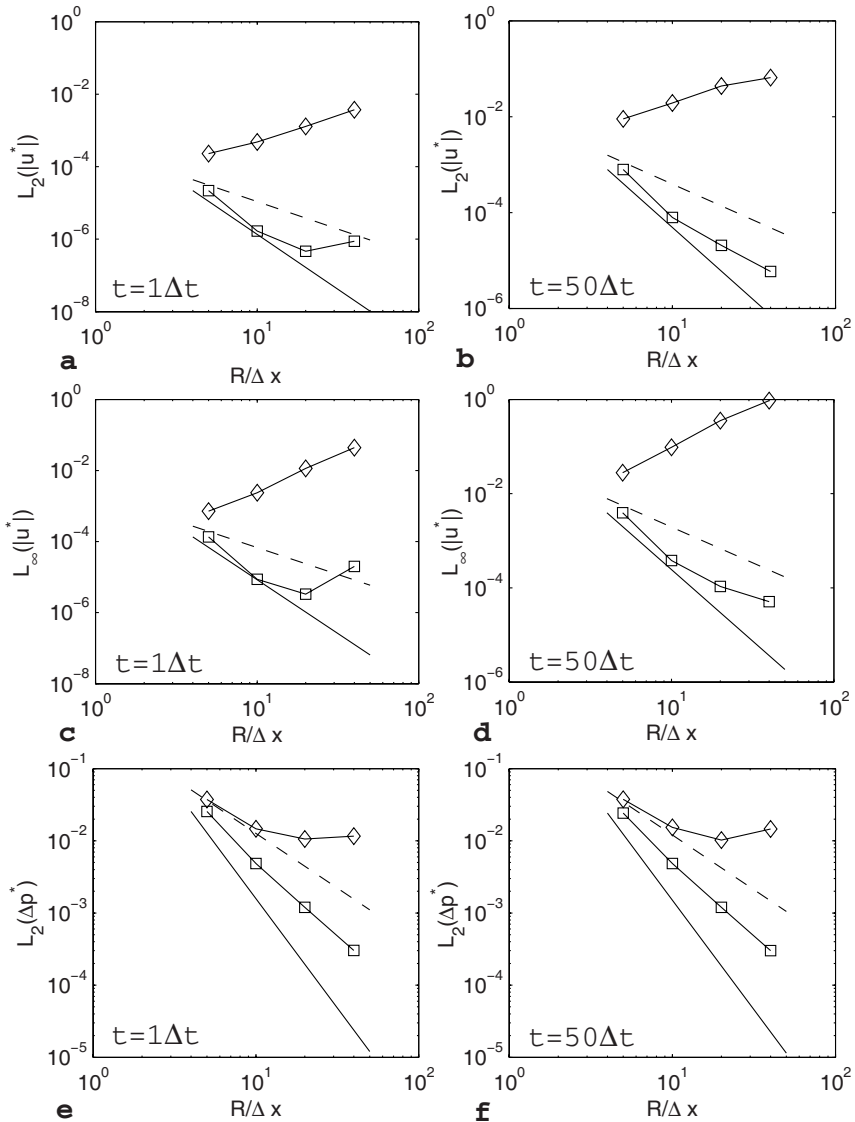


Figure 1:  $L_2(|\mathbf{u}^*|)$  (a,b),  $L_\infty(|\mathbf{u}^*|)$  (c,d) and  $L_2(\Delta p^*)$  (e,f) error norms after one and fifty time steps. White squares represents HF and black diamonds are Youngs error norms. Dashed lines are first order and solid lines are second order curves.

proved that setting density  $\rho_l/\rho_g$  and viscosity  $\mu_l/\mu_g$  ratios as in the experiments leads to numerical errors. Since similitudes with experiments are guaranteed by Eötvös and Morton numbers, which involve only liquid properties, gas properties are set in order to fix aforementioned ratios to  $\rho_l/\rho_g = 1000$  and  $\mu_l/\mu_g = 100$ . Bhaga and Weber tested that wakes behind the bubbles are closed and symmetric until  $Re < 110$ . Since the simulations are performed on a 2D axisymmetrical domain, the cases chosen satisfy this condition.

A bubble of diameter  $D$  is centered at  $(0, 2D)$  of a  $[0, 4D] \times [0, 12D]$  axisymmetric rectangular domain, with  $x = 0$  being the revolution axis. Domain sizes are set in order to avoid boundary influence, following Hua *et al.* sensitivity analysis [14]. Hua *et al.* show also that mesh resolution  $D/\Delta x \geq 20$  ensures grid independence, then a  $80 \times 240$  grid is chosen. Constant pressure is set on top and bottom boundaries. Free-slip condition is imposed on the domain side.

All the simulations are run until a steady state condition for the rising bubble is achieved, for both Height Function and Youngs algorithms. Then, bubble terminal velocity is computed and Reynolds number is evaluated for each case. Bubbles terminal shape for HF and Youngs cases are reported in Tab. 1, together with experimental and numerical Reynolds numbers. For cases I, II and III, both methods show similar performances, with errors in Reynolds number included below 5%. HF method performs slightly better than Youngs. Bubble shapes are very similar for both methods, close to experiments. Case IV is characterized by the highest bubble terminal velocity, then by the highest difference in gas-liquid phases velocity. Interface profile is highly deformed, the most in the bubble back, where interface changes its orientation in really a sharp way. Then, interface is poorly solved by the grid and HF algorithm computes wrong local geometry, leading to errors in capillary effects calculation. Wrong forces computation leads to the detachment of little parts of gas at bubble trail in the simulations. Then, the smaller bubble main body moves with a lower velocity with respect to experiments.

## 5 Vapor bubble growing in superheated liquid




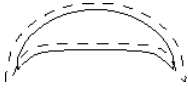
Height Function interface reconstruction algorithm and evaporation model are implemented in order to simulate the growth of a spherical vapor bubble in an infinitely extended superheated liquid. In accord to Plesset and Zwick analysis [15], bubble growth process can be temporally split in two stages. A first stage, called inertia-controlled growth, which starts at bubble formation. At this stage, growth is governed by momentum interaction between bubble and surrounding liquid. Later, an asymptotic stage is reached, called heat-transfer-controlled growth. In this stage, characterized by a growth rate lower than first stage, growth is limited by heat transport to the interface. This asymptotic stage is the object of this study. Scriven [16] has derived an analytical solution for this stage, neglecting viscous and surface tension effects and considering the interface at saturation temperature. He obtained analytical bubble radius  $R$  as a function of time  $t$ :

$$R(t) = 2\beta\sqrt{\gamma t} \quad (16)$$





Table 1: Comparison of experimental and numerical Reynolds number. Figures represents bubble terminal shapes computed at  $\alpha = 0.5$  contours. Solid line is HF shape, dashed line is Youngs shape. Errors between parenthesis are computed as  $\text{err} = |\text{Re} - \text{Re}_{\text{num}}|/\text{Re}$ .

Test case	Experiments [13]	Simulations		
		Shapes	$\text{Re}_{HF}$	$\text{Re}_Y$
I	E = 116 M = 848 Re = 2.47		2.37 (4%)	2.37 (4%)
II	E = 116 M = 41.1 Re = 7.16		7 (2.2%)	6.94 (3.1%)
III	E = 116 M = 1.31 Re = 20.4		19.66 (3.7%)	19.55 (4.2%)
IV	E = 116 M = 0.103 Re = 42.2		37.8 (10.4%)	39 (7.6%)

where  $\beta$  is a growth constant which details can be found in [16] and  $\gamma$  is liquid thermal diffusivity. This solution is used to validate numerical results.

The growth of a bubble of initial radius  $R_0 = 0.1$  mm is simulated. The bubble is centered at  $(4R_0, 0)$  of a  $[0, 8R_0] \times [0, 4R_0]$  axisymmetric rectangular domain, with  $y = 0$  being the revolution axis. A uniform mesh size is chosen, with  $1\mu\text{m}$  element size. Such a fine grid is necessary in order to solve the thin

thermal boundary layer surrounding bubble interface. As boundary conditions, pressure is fixed at all boundaries except for the axis. The fluid flow is laminar and gravity effects are neglected. All vapor and liquid properties are considered constant at saturation temperature. Initial bubble size is large enough to neglect vapor saturation temperature rising due to pressure jump across interface, then saturation temperature is equal for both phases. Initial pressure and velocity fields are zero. Initial temperature is saturation temperature for the bubble, while liquid is superheated at a temperature  $T_\infty = T_{sat} + 5^\circ\text{C}$ .

A thin thermal boundary layer is placed around the interface on liquid side. Since the simulation starts at a  $t = t_0$ , when  $R(t_0) = R_0$ , a thermal boundary layer has been developing around the bubble since the beginning of heat-transfer-controlled growth stage. Temperature field around the bubble at  $t = t_0$  can be extrapolated from Scriven solution as a function of spatial coordinate  $r$  and time. The thickness of initial thermal layer  $\delta_T$  is defined as:

$$\delta_T = r(T = T_{sat} + 0.99(T_\infty - T_{sat})) - R_0 \quad (17)$$

Great attention has to be paid to initial thermal boundary layer position. Analytical initialization suggests thermal layer to begin at  $r = R_0$ . Nevertheless, in those cells which interface is less aligned with grid (close to  $\pi/4$  and  $3/4\pi$ ), cells centroids are located at  $r > R_0$ . Then, in those cells, thermal layer intersects bubble interface, leading to a faster initial growth rate than analytical. To avoid this effect, thermal boundary layer is initialized with a bit of misplacement, about 1-2 cells, outside bubble interface.

Three different fluids are tested. Water at atmospheric pressure, HFE-7100 at 0.52 bar, both with  $\beta = 15.1$  and  $\delta_T = 7 \mu\text{m}$  and R134a at 0.84 bar, with  $\beta = 9.34$  and  $\delta_T = 11 \mu\text{m}$ . The choice of each system pressure is done in order to have similar growth constants for the fluids.

During the whole simulation, bubble shape remains spherical. At initial stage of this work, this was proved not to happen using Youngs method for evaluating interface curvature. Moreover, growth was too fast due to high convective heat transfer led by unreal velocities related to errors in curvature calculation. Figure 2 shows bubble radius evolution obtained through HF method compared to analytical solutions, for all the fluids. Numerical data show good agreement with analytical results. For each fluid, bubble numerical growth rate follows a  $\sqrt{t}$  proportional law, as it should be from Eq. (16). However, this does not happen during initial growth phase, in a more evident way for HFE-7100 and R134a. This initial phase can be meant as a settlement phase, in which thermal boundary layer arranges to fit interface position. This settlement is reflected on numerical growth rates lower than analytical ones at the beginning of the simulations. As detected by Kunkelmann and Stephan [17], liquid thermal conductivity is the parameter that rules the length of this thermal layer settlement phase. The higher is liquid thermal conductivity, the faster is thermal layer arrangement. For this reason numerical bubble growth rate deviation from analytical curve is more evident for refrigerant fluids and the most for HFE-7100, which has the lowest thermal conductivity.



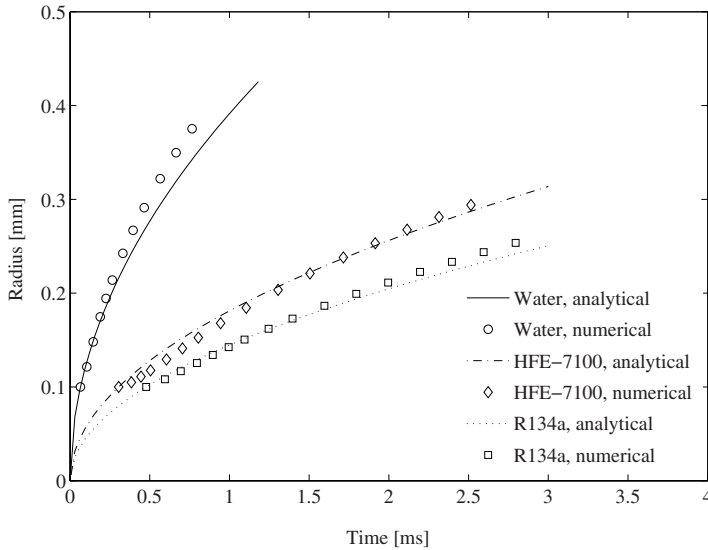


Figure 2: Vapor bubble radius over time for analytical and numerical solutions.

## 6 Conclusions

The main objective of this work was to implement the Hardt and Wondra [8] evaporation model in a finite-volume framework, in order to simulate a vapor bubble growing in superheated liquid for some working fluids. Bubble shape is computed in implicit way through VOF scheme and an interface reconstruction algorithm has the task to compute interface curvature. Accurate curvature computation is fundamental in order to correctly account for surface tension effects, leading to physical consistent interface shapes. This objective is reached implementing Height Function algorithm, which evaluate curvatures through derivatives of a color function obtained integrating volume fraction field. Inviscid static drop test case assesses HF performances compared to the widely used Youngs algorithm. Inaccuracies in terms of errors in velocity fields and pressure jump over interface scale with second order to mesh element size. Isothermal bubble rising test cases show good agreement of HF numerical results with experiments for  $Re \leq 20$ , with errors in numerical Reynolds number lower than 5%. HF algorithm coupled with evaporation model leads to excellent agreement of vapor bubble numerical growth rate compared to analytical solutions. The success of the simulations is related to correct treatment of initial thermal boundary layer. Fine grid resolution and proper boundary layer placement are necessary to obtain exact bubble growth rate.



## References

- [1] Juric, D. & Tryggvason, G., Computations of boiling flows. *Int J of Multiphase Flow*, **24**, pp. 387–410, 1998.
- [2] Welch, S.W.J. & Wilson, J., A volume of fluid based method for fluid flows with phase change. *J of Computational Physics*, **160**, pp. 662–682, 2000.
- [3] Son, G. & Dhir, V.K., Numerical simulation of film boiling near critical pressures with a level set method. *J of Heat Transfer*, **120**, pp. 183–192, 1998.
- [4] Youngs, D.L., Time-dependent multi-material flow with large fluid distortion. *Numerical Methods for Fluid Dynamics*, eds. K.W. Morton & M.J. Baines, Academic Press, pp. 273–285, 1982.
- [5] Hernandez, J., Lopez, J., Gomez, P., Zanzi, C. & Faura, F., A new volume of fluid method in three dimensions-Part I: Multidimensional advection method with face-matched flux polyhedra. *Int J for Numerical Methods in Fluids*, **58**, pp. 897–921, 2008.
- [6] Malik, M., Sheung-Chi Fan, E. & Bussmann, M., Adaptive VOF with curvature-based refinement. *Int J for Numerical Methods in Fluids*, **55**, pp. 693–712, 2007.
- [7] Cummins, S.J., Francois, M.M. & Kothe, D.B., Estimating curvature from volume fractions. *Computers and Structures*, **83**, pp. 425–434, 2005.
- [8] Hardt, S. & Wondra, F., Evaporation model for interfacial flows based on a continuum-field representation of the source terms. *J of Computational Physics*, **227**, pp. 5871–5895, 2008.
- [9] Brackbill, J.U., Kothe, D.B. & Zemach, C., A continuum method for modeling surface tension. *J of Computational Physics*, **100**, pp. 335–354, 1992.
- [10] van Leer, B., Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J of Computational Physics*, **32**, pp. 101–136, 1979.
- [11] Issa, R.I., Solution of the implicitly discretized fluid flow equations by operator-splitting. *J of Computational Physics*, **62**, pp. 40–65, 1985.
- [12] Schrage, R.W., *A theoretical study of interphase mass transfer*. Columbia University Press: New York, 1953.
- [13] Bhaga, D. & Weber, M.E., Bubbles in viscous liquid: shapes, wakes and velocities. *Journal of Fluid Mechanics*, **105**, pp. 61–85, 1981.
- [14] Hua, J., Stene, J.F. & Lin, P., Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method. *Journal of Computational Physics*, **227**, pp. 3358–3382, 2008.
- [15] Plesset, M.S. & Zwick, S.A., The growth of vapor bubbles in superheated liquids. *J of Applied Physics*, **25**, pp. 493–500, 1954.
- [16] Scriven, L.E., On the dynamics of phase growth. *Chemical Engineering Science*, **10**, pp. 1–13, 1959.
- [17] Kunkelmann, C. & Stephan, P., CFD simulation of boiling flows using the Volume-of-Fluid method within OpenFOAM. *Numerical Heat Transfer, Part A*, **56**, pp. 631–646, 2009.

