Received: 7 October 2019 | Revised: 15 June 2020 | Accepted: 18 June 2020

DOI: 10.1002/fld.4886

RESEARCH ARTICLE

WILEY

Machine-learning-based surface tension model for multiphase flow simulation using particle method

Xiaoxing Liu[1] | **Koji Morita**[2] | **Shuai Zhang**[3]

[1]Sino-French Institute of Nuclear Engineering & Technology, Sun Yat-Sen University, Zhuhai City, China

[2]Department of Applied Quantum Physics and Nuclear Engineering, Kyushu University, Fukuoka, Japan

[3]School of Aeronautics and Astronautics, Zhejiang University, Hangzhou City, China

Correspondence
Xiaoxing Liu, Sino-French Institute of Nuclear Engineering & Technology, Sun Yat-Sen University, Zhuhai City, Guangdong Province 519-082, China.
Email: liuxx85@mail.sysu.edu.cn

Summary

Particle methods have shown their potential for simulating multiphase flows due to the convenience in capturing interfaces. However, when it comes to estimate the surface tension, calculation of the curvature of the interface remains challenging. Traditional methods are based on derivative models to estimate the curvature analytically from the particle number density or color function that marks different phases. It is difficult to estimate the curvature accurately in traditional derivative models. In this study, background cells are built up and are used to predict the curvature through machine learning. By training on a data set generated using circles of varying sizes, a relation function is found to predict the curvature from the particle distribution near the interface. Together with the enhanced schemes developed in our previous study, multiphase flows with surface tension are studied within the framework of the moving particle semi-implicit method.

KEYWORDS

machine learning, MPS, multiphase flow, surface tension model

1 | INTRODUCTION

Surface tension plays an important role in lots of multiphase flow phenomenon. In multiphase flow where bubbles and droplets exist, the behavior of the bubbles and droplets is largely affected by surface tension. Accurate modeling of surface tension is crucial to simulate multiphase flow with surface tension.

A majority of the multiphase flow simulation is based on Eulerian methods, such as the volume of fluid (VOF)[1,2] and level set.[3,4] In these grid-based methods, special treatment is necessary to capture or track the interface. In recent decades, there is a growing interest in developing Lagrangian methods as alternatives to traditional grid-based methods. In contrast to Eulerian methods, interfaces are represented by moving particles self-adaptively without special treatment. Popular Lagrangian methods include smoothed particle hydrodynamics (SPH),[5,6] moving particle semi-implicit (MPS)[7,8] and finite volume particle (FVP)[9,10] methods. In this article, we employ the particle method in the Lagrangian framework to study multiphase flow with surface tension.

In the context of particle methods, Nugent and Posch[11] are the first to develop a SPH model for surface tension based on interphase attractive potentials. Surface tension model based on interparticle force is then developed for MPS[12] and applied in FVP.[13,14] Although the implementation of an interphase attractive potential is straightforward, one of the difficulties is that the resulting surface tension needs to be calibrated. Inspired by the continuum surface force (CSF)[15] widely applied in Eulerian methods, Morris[6] developed the SPH formulation for the CSF model with smoothed color function.

In Morris' model, curvature is calculated by the divergence of the unit interface normal. Two major issues arise when applying CSF model in particle methods. One is the numerical error induced by smoothed color function and the other is the inaccuracy in calculating curvature. Hu and Adams[16] used a sharp color function with discontinuity at the interface. To calculate the curvature more accurately, Admi et al[17] then developed a new surface tension formulation using a high-order reproducing divergence approximation. Zhang[18] reconstructed the interface with polynomials, which allows for high-order calculation of the curvature. Similarly, Duan et al[19] obtained analytical expressions for the contours of the color function in a transition region near the interface. However, singularity caused by the small number of the neighboring different phase particles near the interface cannot be avoided totally if curvature is calculated analytically from the color function. In some serious cases, disordered particles near the interface result to interface penetration, which raises the difficulty of the curvature calculation. To remove the singularity, Zhang et al[10] calculated the surface tension on background cell based on CSF model and then extrapolated to particles.

Recently, instead of deriving mathematical formulation to calculate the curvature from the distribution of volume fraction analytically, an alternative approach based on machine learning was proposed by Qi et al[20] for VOF in their short note. A data set generated using circles of different sizes is used as training data to get the relationship between nine volume fractions in and around the target cell and the curvature. Results suggested that their approach would be a viable alternative to conventional derivations. In their short note, motion of a droplet due to the effect of surface tension is the only test by implementing the machine-learning-based surface tension model in a flow solver. The performance of this new surface tension model in simulating more complex multiphase flow problems is not demonstrated. Also, this machine-learning-based surface tension developed for VOF method is not applicable for particle-based methods.

In this study, we attempt to introduce the machine-learning-based surface tension model into particle methods. To avoid the inaccuracy of curvature calculation in traditional derivative models, our motivation is to calculate the curvature based on the particle distribution near the interface using machine learning. To get a set of well-ordered input data for training, background cells are built up and the color function at each cell center is interpolated from the color function of surrounding particles. Also, more complex multiphase flow problems such as rising bubbles in liquid are used to validate the new method.

This work is based on MPS method. During the past two decades, significant improvements have been achieved to improve the accuracy and stability of the MPS method, including corrective gradient model,[21,22] high-order Laplacian model,[8,23,24] least-square MPS,[25] and conservative pairwise-relaxing method.[26] Extensive study is conducted in multiphase flow simulation using MPS.[27-32] By introducing the dummy particle into the corrective gradient model, an improved MPS with enhanced accuracy and stability was developed in our previous study.[22] In this study, multiphase flows with surface tension are studied in the framework of this enhanced MPS.

The article is organized as follows: the MPS method is briefly described in the next section. In Section 3, the surface tension model using machine learning is described. In Section 4, numerical tests are presented and discussed. Finally, the conclusions are drawn in Section 5.

2 | MPS METHOD

In the MPS method, the following continuity equation and Navier-Stokes equations for incompressible flows are solved:

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot (\mu \nabla \mathbf{u}) + \mathbf{g} + \frac{\mathbf{f}_s}{\rho}, \tag{2}$$

where $\mathbf{u}$ is the velocity, t is the time, ρ is the density, μ is the viscosity, p is the pressure, $\mathbf{g}$ is the gravity, and $\mathbf{f}_s$ represents the surface tension.

2.1 | Discretization method

In the MPS method, the gradient and Laplacian operators can be approximated as

$$\langle \nabla \phi \rangle_i = \frac{d}{n^0} \sum_{j \neq i} \frac{\phi_j - \phi_i}{|\mathbf{r}_{ij}|^2} \mathbf{r}_{ij} w_{ij}, \tag{3}$$

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{\lambda n^0} \sum_{j \neq i} (\phi_j - \phi_i) w_{ij}, \tag{4}$$

where ϕ is a scalar physical quantity, d is the number of space dimensions, and n^0 is the constant particle number density. $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ is the distance vector from the ith particle to the jth particle at positions $\mathbf{r}_i$ and $\mathbf{r}_j$, respectively, and w_{ij} is the kernel function that describes the interaction of each numerical particle with its neighbors, defined as

$$w_{ij} = \begin{cases} \frac{r_e}{|\mathbf{r}_{ij}|} - 1 & 0 < |\mathbf{r}_{ij}| \leq r_e \\ 0 & |\mathbf{r}_{ij}| > r_e \end{cases}, \tag{5}$$

where r_e is the cut-off radius and is usually set between 2.0 l_0 and 4.0 l_0. In the present study, r_e is set to $3.1 l_0$. Several previous studies[8,22,33] suggest that promising results can be obtained by setting r_e to $3.1 l_0$. l_0 is the initial distance between adjacent particles. λ is defined as

$$\lambda = \frac{\sum_{j \neq i} w_{ij} |\mathbf{r}_{ij}|^2}{\sum_{j \neq i} w_{ij}}. \tag{6}$$

2.2 | Time integration

As a fractional step method, the MPS method first estimates particle's intermediate velocity and position in the explicit step without the pressure force:

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \left[\frac{\mu}{\rho} \nabla^2 \mathbf{u}^n + \frac{\mathbf{f}_s^n}{\rho} + \mathbf{g} \right], \tag{7}$$

$$\mathbf{r}^* = \mathbf{r}^n + \Delta t \cdot \mathbf{u}^*, \tag{8}$$

where Δt is the time step size and $\mathbf{u}^*$ is the intermediate velocity.

To solve the pressure field, the following PPE is solved

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^*. \tag{9}$$

In the next step, they are updated with the pressure force as

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1}, \tag{10}$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \Delta t \cdot \mathbf{u}^{n+1}. \tag{11}$$

2.3 | Advanced schemes

In our previous study,[22] advanced MPS schemes with enhanced accuracy and stability have been developed. By analyzing the accuracy of the derivative schemes in MPS, it was found numerical error was mainly caused by the irregular distribution of particles. Our idea in Reference [22] was to introduce a dummy particle to compensate the numerical error in the gradient and Laplacian models. The concept of the dummy particle is illustrated in Figure 1. The dummy particle was introduced to balance the interaction to a target particle i from neighbor particles. The gradient model is:

$$\langle \nabla \phi \rangle_i = \frac{d}{n^0} \left(\sum_{j \neq i} \frac{\phi_j - \widetilde{\phi}_i}{|\mathbf{r}_{ij}|^2} \mathbf{r}_{ij} \widetilde{M}_i w_{ij} + \frac{\phi_d - \widetilde{\phi}_i}{|\vec{r}_{id}|^2} \mathbf{r}_{id} \widetilde{M}_i w_{id} \right), \tag{12}$$

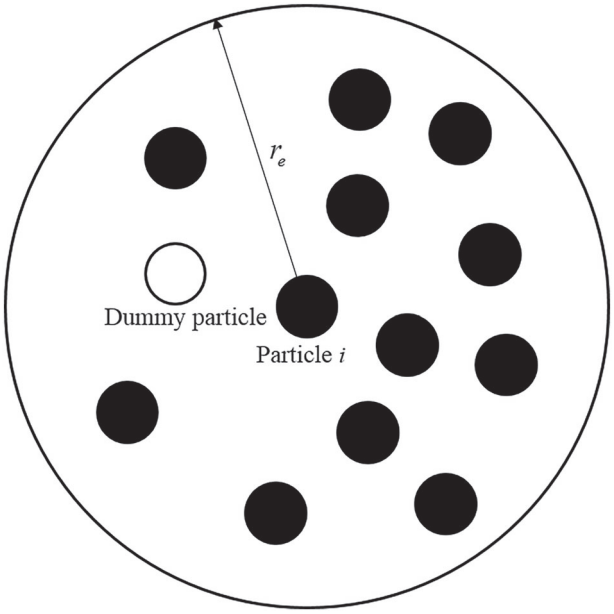

FIGURE 1 Estimated position of a dummy neighbor particle for particle i

here index d is for dummy particle (Figure 1). $\widetilde{M}_i$ is the corrective matrix:

$$\widetilde{M}_i = \begin{bmatrix} \frac{d}{n^0}\left(\sum \frac{w_{ij}x_{ij}^2}{|\mathbf{r}_{ij}|^2} + \frac{w_{id}x_{id}^2}{|\mathbf{r}_{id}|^2}\right) & \frac{d}{n^0}\left(\sum \frac{w_{ij}x_{ij}y_{ij}}{|\mathbf{r}_{ij}|^2} + \frac{w_{id}x_{id}y_{id}}{|\mathbf{r}_{id}|^2}\right) \\ \frac{d}{n^0}\left(\sum \frac{w_{ij}x_{ij}y_{ij}}{|\mathbf{r}_{ij}|^2} + \frac{w_{id}x_{id}y_{id}}{|\mathbf{r}_{id}|^2}\right) & \frac{d}{n^0}\left(\sum \frac{w_{ij}y_{ij}^2}{|\mathbf{r}_{ij}|^2} + \frac{w_{id}y_{id}^2}{|\mathbf{r}_{id}|^2}\right) \end{bmatrix}^{-1}. \tag{13}$$

The Laplacian model is:

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{n^0}\left(\sum_{j\neq i} \frac{\phi_j - \phi_i}{|\vec{r}_{ij}|^2} w_{ij} + \frac{\phi_d - \phi_i}{|\vec{r}_{id}|^2} w_{id}\right). \tag{14}$$

One can refer to Reference [22] for more details about the configuration of dummy particle.

To treat the discontinuity in multiphase flows, following,[8,22] harmonic mean of the viscosity coefficient and density is used. The viscosity term is calculated by:

$$\langle \nabla \cdot \mu(\nabla \mathbf{u}) \rangle_i = \frac{2d}{n^0}\left(\sum_{j\neq i} \frac{2\mu_i\mu_j}{\mu_i + \mu_j} \frac{\mathbf{u}_j - \mathbf{u}_i}{|\vec{r}_{ij}|^2} w_{ij} + \mu_i \frac{\mathbf{u}_d - \mathbf{u}_i}{\rho_i |\vec{r}_{id}|^2} w_{id}\right). \tag{15}$$

Left-hand side of PPE is discretized by

$$\left\langle \nabla \cdot \left(\frac{\nabla p}{\rho}\right) \right\rangle_i = \frac{2d}{n^0}\left(\sum_{j\neq i} \frac{\rho_i + \rho_j}{2\rho_i\rho_j} \frac{p_j - p_i}{|\vec{r}_{ij}|^2} w_{ij} + \frac{p_d - p_i}{\rho_i |\vec{r}_{id}|^2} w_{id}\right). \tag{16}$$

Also, following Duan et al's work in Reference [19], density is smoothed to avoid the instability caused by sharp-varied density.

$$\rho_i = \frac{\sum_{j\neq i} \rho_i w_{ij}}{\sum_{j\neq i} w_{ij}}. \tag{17}$$

3 | SURFACE TENSION MODEL USING MACHINE LEARNING

In the CSF model, to calculate the surface tension a color function C is introduced:

$$C_i = \begin{cases} 0 & i \text{ is in the specified phase} \\ 1 & i \text{ is in the other phase} \end{cases}. \tag{18}$$

And the surface tension is calculated by

$$\mathbf{f}_s = \sigma \cdot \kappa \cdot \nabla C, \tag{19}$$

here σ is the surface tension coefficient and κ is the curvature of the interface. Calculation of the curvature is the most challenging issue in CSF model. In References [6,16,17], curvature is calculated from the unit interface normal $\mathbf{n}$: $\kappa = -\nabla \cdot \mathbf{n}$. As $\mathbf{n}$ is expressed by the first derivative of the color function, accuracy of the curvature calculation depends on the second derivative of the color function, which is difficult to estimate accurately in particle methods. In References [18,19], interface is reconstructed and the curvature is calculated analytically from the interpolation polynomial or reconstructed contour function. However, second derivative of the reconstructed polynomial and contour function is nevertheless necessary to calculate the curvature. In summary, all the above methods are based on derivative models to estimate the curvature from the color function and it is difficult to estimate the curvature accurately. In Reference [20], a different approach based on machine learning to predict curvature from the volume fraction in VOF was developed. Explicit analytical function between the volume fraction and the curvature is unnecessary in their machine-learning-based model. Trained from a data set generated using well-defined shapes where the curvature and volume fractions are known beforehand, a functional relationship is found:

$$h\kappa_{i,j} = \psi\left(\begin{bmatrix} V_{i-1,j+1} & V_{i,j+1} & V_{i+1,j+1} \\ V_{i-1,j} & V_{i,j} & V_{i+1,j} \\ V_{i-1,j-1} & V_{i,j-1} & V_{i+1,j-1} \end{bmatrix}\right), \tag{20}$$

here h is the cell size and $V_{i,j}$ is the volume fraction of cell (i, j). ψ is the function between volume fraction and curvature. The nine volume fractions are input and the output is nondimensional curvature. Benefited from the good order of the cells in structural grid, a well-trained function ψ is obtained using machine learning. The main challenge of applying the machine-learning technique in particle methods is the unstructured feature of particles and how to define suitable input for machine learning.

In this study, by overlaying a temporal grid on the particle calculation region, we apply the machine-learning-based surface tension model to this temporal grid. Our surface tension model based on machine learning is described as the followings. Firstly, color function is smoothed as

$$\widetilde{C}_i = \frac{\sum_j w_{ij} C_j}{\sum_j w_{ij}}. \tag{21}$$

To obtain ordered input, 3×3 Cartesian cells are configured centered on the interface particle as shown in Figure 2. Color function for each cell is calculated by weighted sum of the color function of particles around each cell center:

$$D_m = \frac{\sum_j w_{mj} \widetilde{C}_j}{\sum_j w_{mj}}, \quad m = 1, 2...9, \tag{22}$$

here D_m is the color function of cell m. Particles located within each cell are used in the calculation of the color function of the cell. Effects of different cell sizes on the model accuracy will be shown later.

Using the nondimensional curvature $\kappa_i \cdot r_e$, our goal is to find the function ψ:

$$\kappa_i \cdot r_e = \psi([D_1, D_2 \ \ldots \ D_9]). \tag{23}$$

A data set containing curvature and color functions is generated using circles of different sizes as in Reference [20]. A 1×1 domain is resolved by 1000×1000 particles and the interface with shape of varied circles is configured. The radius of circles ranges from 0.005 to 0.5 with the increment equal to 0.001. The data set of color functions and curvature are generated near the interface along each circle. In total, 472 340 sets of color function and curvature data are generated. Among these data sets, 70% is used for training, 15% for validation, and 15% for test.

We use *KERAS*, an open source library to build the neural network. A sequential model is employed. Input is a nine-dimensional array of the color function. Several tests using different cell sizes, hidden layers and epochs are

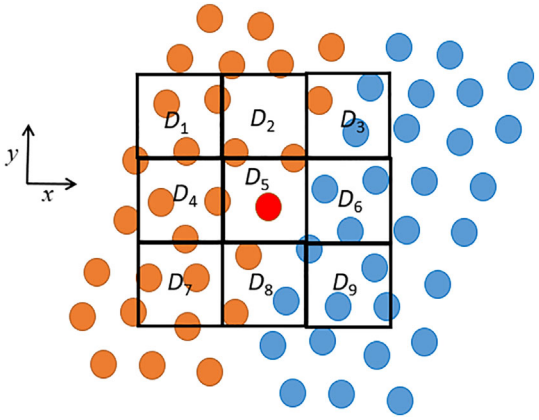

FIGURE 2 Configuration of the background cells in the machine-learning-based surface tension model [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Result of the tests using different cell sizes

Cell size	Layer	Epoch	Correlation coefficient
3 l_0	2	500	.8271
4 l_0	2	500	.9722
5 l_0	2	500	.9554
6 l_0	2	500	.9623

TABLE 2 Result of the tests using different layers and epochs

Layer	Epoch	Correlation coefficient
1	500	.9574
2	500	.9722
3	500	.9738
2	250	.9696
2	1000	.9734

TABLE 3 Result of the tests with smooth and without smooth

Smooth	Layer	Epoch	Correlation coefficient
Yes	2	500	.9722
No	2	500	.8552

conducted and the results of correlation coefficient between the test data and the data predicted by machine learning are summarized in Tables 1 and 2. Using a cell size equal to $4l_0$, two hidden layers with 500 epochs would give enough accuracy. A test is conducted as well to demonstrate the advantage of using smoothed color function in calculating the color function of the cells and the result is summarized in Table 3. The accuracy is improved significantly using smoothed color function. Figure 3 shows the structure of the network. Two hidden layers with 100 neurons are used. Output is the normalized nondimensional curvature. *ReLU* activation function is used for the hidden layer and linear activation function is used for the output.

4 | NUMERICAL TESTS

4.1 | Computation of the curvature of a circle and sine wave

To test the curvature estimation using machine learning, the curvature along a circle and the curve defined by a sine wave: $y(x) = A\sin(x)$ is calculated and compared with the analytical result. Calculations using derivative model are also

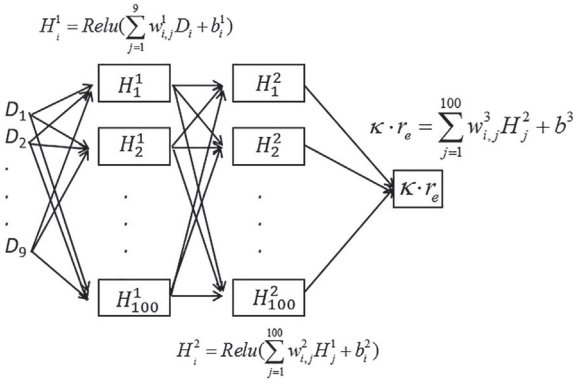

FIGURE 3 Structure of the network used in training the machine learning model

conducted for comparison. In the derivative model, we use the divergence model with corrective matrix[21] to calculate the curvature from unit interface normal:

$$\kappa = -\frac{d}{n^0} \sum_{j \neq i} \frac{\mathbf{n}_j - \mathbf{n}_i}{|\mathbf{r}_{ij}|^2} \cdot M_i \mathbf{r}_{ij} w_{ij}, \tag{24}$$

where M_i is the corrective matrix:

$$M_i = \begin{bmatrix} \frac{d}{n^0} \sum \frac{w_{ij} x_{ij}^2}{|\mathbf{r}_{ij}|^2} & \frac{d}{n^0} \sum \frac{w_{ij} x_{ij} y_{ij}}{|\mathbf{r}_{ij}|^2} \\ \frac{d}{n^0} \sum \frac{w_{ij} x_{ij} y_{ij}}{|\mathbf{r}_{ij}|^2} & \frac{d}{n^0} \sum \frac{w_{ij} y_{ij}^2}{|\mathbf{r}_{ij}|^2} \end{bmatrix}^{-1}. \tag{25}$$

In the calculation of the curvature of a circle, radius of the circle is set to 1 and initial particle distance is set to 0.1. Calculation results using three different particle resolutions are compared in Figure 4A,B. The derivative model shows large fluctuation in predicting the curvature. The curvature predicted using machine learning agrees very well with the analytical result. Calculation result in Figure 4B suggest that machine learning cannot guarantee convergence under particle refinement, which is also concluded in Reference [20].

As discussed in the previous section, background cell size is set to $4l_0$. This is equivalent to an influence radius of 6.0 l_0, about double of the influence radius in the derivative model, which is 3.1 l_0. Here the influence radius is used to distinguish with the cut-off radius in the MPS calculation and generally the influence radius in the surface tension calculation is equal to the cut-off radius. To check the effect of influence radius, the influence radius in the derivative model is increased to 6 l_0 for a fair comparison. Comparison results are shown in Figure 4C. The calculation using larger influence radius shows comparable accuracy compared with machine-learning-based model. However, using larger influence radius results to more interaction particles in the calculation and increased calculation time. Using derivative model with influence radius equal to 3.1 l_0, the calculation time for single surface tension calculation is 69 milliseconds, while it increases to 206 milliseconds while using derivative model with influence radius equal to 6.0 l_0. Using machine-learning-based model, the surface tension calculation only takes 6 milliseconds. All the tests are carried out using the same CPU processor (Intel Xeon Processor E5-2643 v2, 3.5 GHz). Although derivative model with influence radius equal to 6.0 l_0 gives better accuracy, it is not practical to use it in actual calculation due to its high-computation cost. Machine-learning-based model shows both promising accuracy and efficiency. In the following numerical tests, derivative model with influence radius equal to radius l_0 is used for comparison.

To compare the machine-learning-based model with another advanced derivative model, the contoured continuum surface force (CCSF) model developed by Duan et al,[19] the curvature of a circle of radius = 0.1 is also calculated using different initial particle size. The *L2* norm of the total error of the calculated curvature is summarized in Table 4. It can be seen that the CCSF model demonstrates good convergence property when the initial particle distance is refined from the 0.1/5 to 0.1/20. Although the machine-learning-based model is not convergent, the results suggest that the machine-learning-based model is more accurate than the CCSF when large particle size is adopted.

In the calculation of the curvature of a sine wave, the calculation condition is set the same as that in Reference [20]. $A = 1.0$ is used. The calculation domain is $2\pi \times 2\pi$ domain divided by the sine wave. 100×100 particles are used. Comparison of the predicted result using derivative model, machine learning and analytical result is shown in Figure 5. The

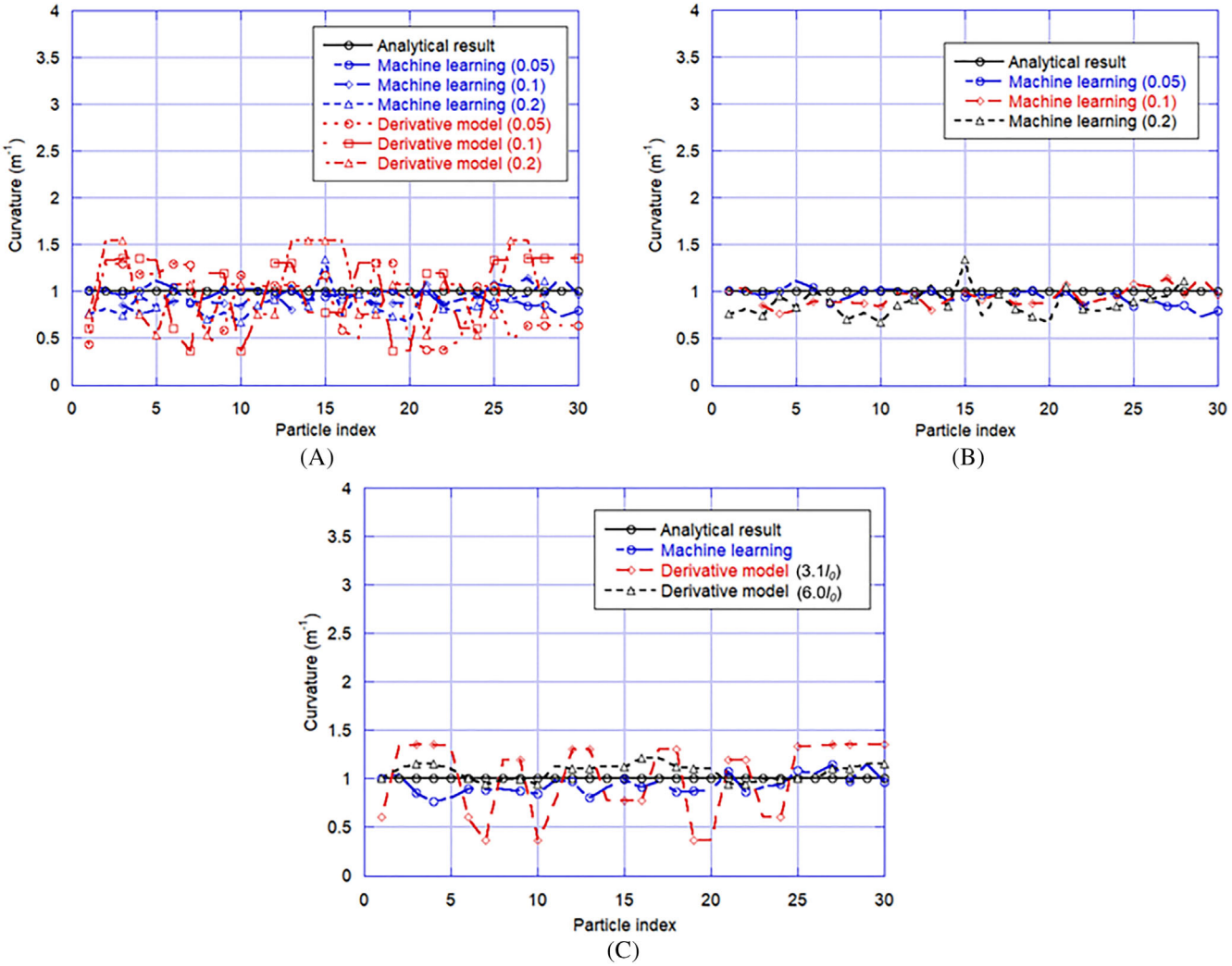

FIGURE 4 Comparison of the curvature predicted by derivative model, machine learning and analytical result of a circle [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 *L2*-norm of the total error of calculated curvature for a circle of radius = 0.1

l_0	CCSF (Duan et al, 2015)	Machine learning
0.1/5	3.730	1.534
0.1/10	1.632	1.010
0.1/20	0.633	1.600

derivative model shows large fluctuation while using machine learning comparable result is obtained. Our result is not as good as that in Reference [20], this is because color functions of the background cells used in our study are smoothed color functions, while in Reference [20] the volume fraction is calculated more accurately using VOF.

4.2 | Static drop in equilibrium

To further validate our model, a test case of static drop in equilibrium is considered. The exact pressure jump across the drop in two-dimensional space is given by

$$\Delta p = \frac{\sigma}{R}. \tag{26}$$

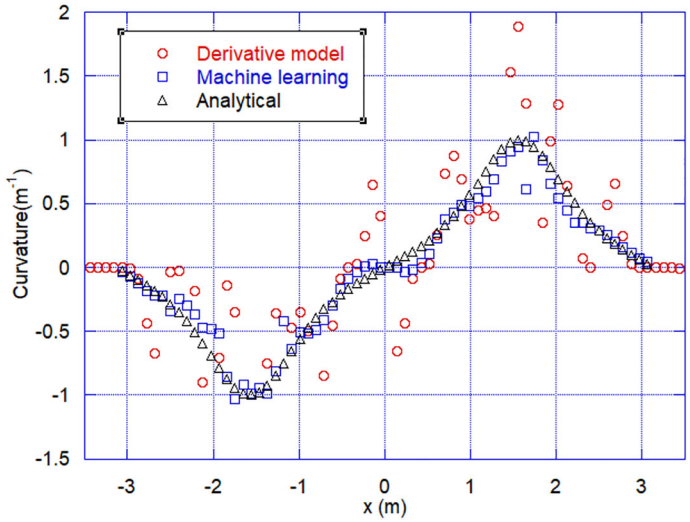

FIGURE 5 Comparison of the curvature of the sine wave predicted by machine learning and analytical result [Colour figure can be viewed at wileyonlinelibrary.com]

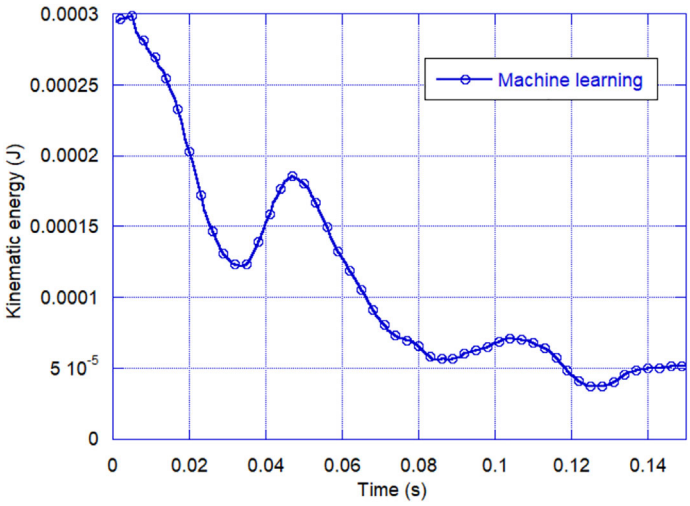

FIGURE 6 Time history of the kinematic energy in the calculation of static drop in equilibrium [Colour figure can be viewed at wileyonlinelibrary.com]

In the calculation, $R = 0.2$ and σ=1.0 are used and initial particle distance is set to 0.01.

Time history of the kinematic energy in the calculation is shown in Figure 6. The steady state of kinematic energy is achieved after 0.1 second with order of $O(10^{-5})$. Pressure drop is shown in Figure 7 and the calculation result using machine learning shows better agreement with the analytical result compared to the derivative model.

4.3 | Single-rising bubble

Rising bubbles in liquid, as a more complex multiphase flow problem, is used to study the surface tension model. Bubbles rising through stagnant liquid are one of the most common gas-liquid flow phenomena and bubble behavior is largely affected by the surface tension. We calculate the rise of a bubble in liquid considered by Hysing et al.[34] Hysing et al[34] reported their comprehensive studies on this problem studying in particular two-dimensional benchmark test cases requiring bubble simulations. The initial configuration of the rising bubble is shown in Figure 8. The physical parameter values used in the two test cases are listed in Table 5. The gravity acceleration is set to 0.98 m/s^2 in both cases. In the simulation, initial particle distance of 0.01 m is used.

Snapshots of bubble shapes at $t = 3.0$ seconds in case 1 are shown in Figure 9. In case 1 surface tension is strong and the shape of the bubble turns into ellipsoidal shape at $t = 3.0$ seconds. Both the bubble shapes obtained by MPS and derivative model and MPS and machine learning model agree well with the reference result. Rising velocity of the bubble is compared in Figure 10. MPS and machine learning model shows slight improvement compared with MPS and the derivative

FIGURE 7 Pressure drop profile in in the calculation of static drop in equilibrium [Colour figure can be viewed at wileyonlinelibrary.com]

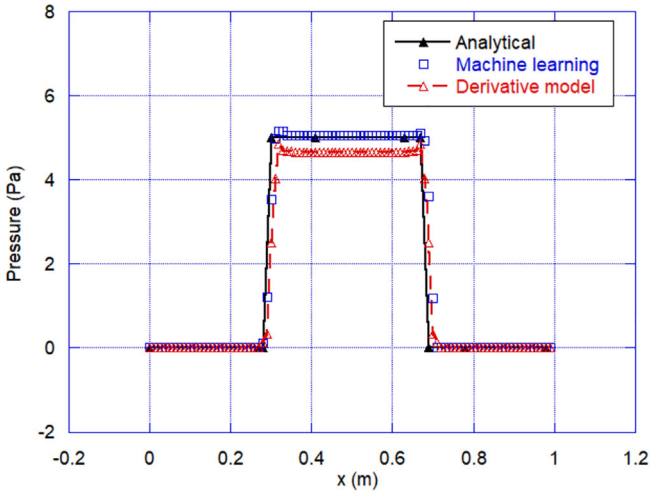

FIGURE 8 Initial configuration of the rising bubble in a liquid

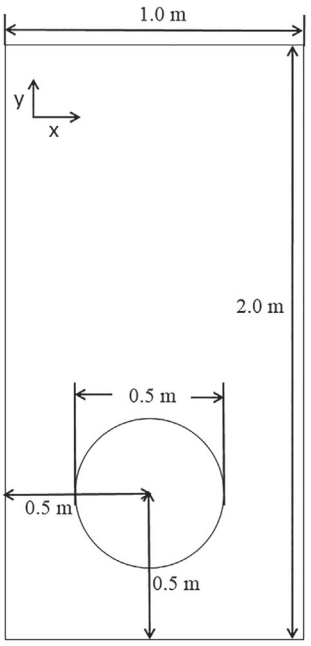

TABLE 5 Physical parameter values defined in the rising bubble tests

	Liquid density (kg/m^3)	Gas density (kg/m^3)	Surface tension coefficient (N/m)	Liquid viscosity (Pa·s)	Gas viscosity (Pa·s)
Case 1	1000	100	24.5	1.0	1.0
Case 2	1000	1	1.96	10.0	0.1

model. The time required to compute 100 iterations is compared using the derivative model and machine-learning-based model, which are 171 and 135 seconds, respectively. The computation time difference is not as large as that in the curvature calculation of a circle. This is because in the single bubble calculation, most of the computation cost is occupied by solving the PPE. Nonetheless, the MPS and machine-learning-based model is still approximately 20% faster than the MPS and derivative model.

Snapshots of bubble shapes at $t = 3.0$ seconds in case 2 are shown in Figure 11. In case 2 surface tension is much smaller than case 1 and the bubble breaks up while rising upwards. Both the two simulation results are comparable with the reference result. Rising velocity of the bubble is compared in Figure 12. Both the two simulation results indicate some underestimation of the rising velocity. This can be attributed to the numerical viscosity of the MPS method. In case 1, surface tension is strong and the effect of numerical viscosity of the MPS is relatively small compared with surface tension

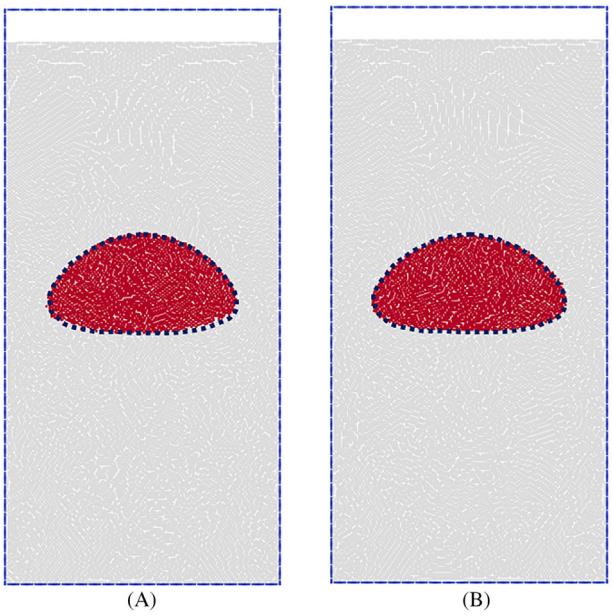

FIGURE 9 Snapshots of bubble shape at $t = 3$ seconds: A, MPS-machine learning based model and, B, MPS-derivative model in case 1 [Colour figure can be viewed at wileyonlinelibrary.com]

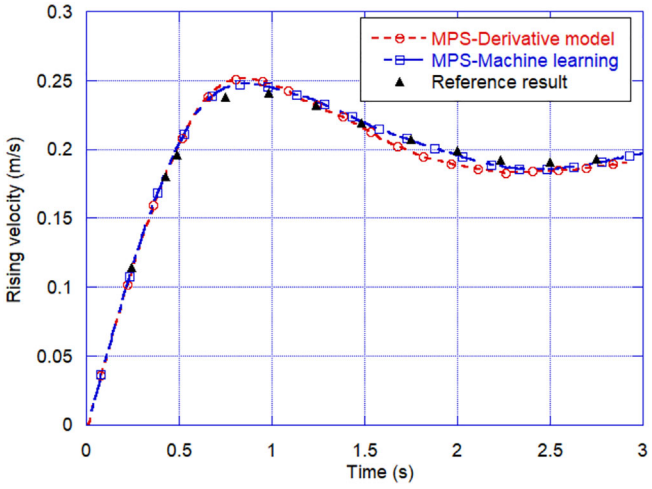

FIGURE 10 Rising velocity of the bubble in case 1 [Colour figure can be viewed at wileyonlinelibrary.com]

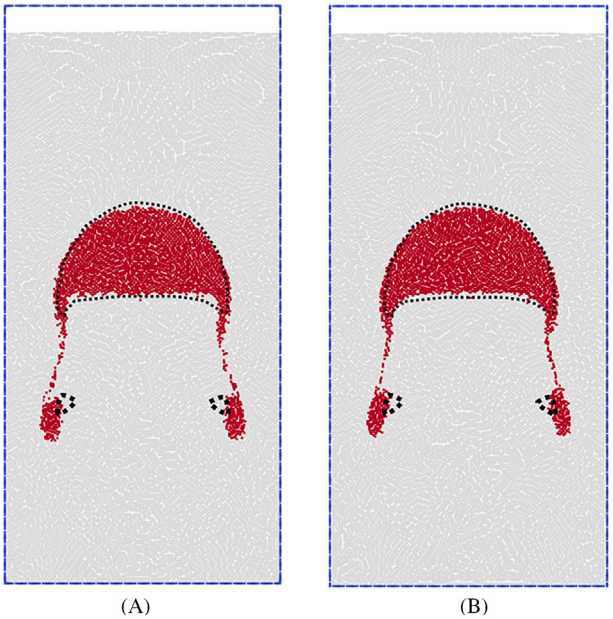

FIGURE 11 Snapshots of bubble shape at $t = 3$ seconds: A, MPS-machine learning and, B, MPS-derivative model in case 2 [Colour figure can be viewed at wileyonlinelibrary.com]

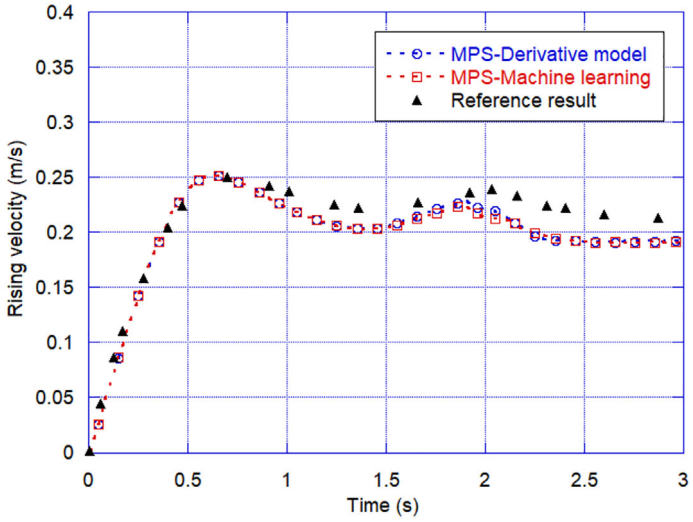

FIGURE 12 Rising velocity of the bubble in case 2 [Colour figure can be viewed at wileyonlinelibrary.com]

effect. In case 2, the effect of numerical viscosity of the MPS is dominant as the surface tension is much smaller. Rising velocity of the bubble is underestimated by MPS even using different surface tension models.

5 | CONCLUSION

In the present study, a machine-learning-based framework is developed for particle methods to calculate the curvature of the interface in multiphase flow simulation. Following Qi et al's work in Reference [20], our machine-learning model is trained on a data set generated using circles of different sizes. Background cells are configured to help produce ordered input for the neural network. Calculation results of the curvature of a circle and sine wave suggest that in the context of particle method, our machine-learning-based model is capable of predicting the curvature of complex interfaces accurately and efficiently, surpassing the classical derivative model. Implementing the surface tension model in the MPS method, static drop in equilibrium and single-rising bubble are simulated. Pressure drop is well reproduced in the simulation result of static drop in equilibrium. In both the two cases of single-rising bubble, both the shape change and rising velocity are well predicted. Especially in case 1 where the bubble motion is dominant by surface tension, the machine-learning model shows obvious improvement compared to the classical derivative model.

In the present study, the size of the background cell is fixed. It might be suitable to use adaptive size of the background cell to handle more complex interfacial phenomenon such as bubble merging in the future. It is expected the MPS method together with the machine-learning-based surface tension model can be applied in solving a wide range of multiphase flow problems. Extension of the present machine-learning-based surface tension model for free-surface flow will also be one of the future works.

ORCID

Xiaoxing Liu https://orcid.org/0000-0002-5941-0640

REFERENCES

1. Hirt CW, Nichols BD. Volume of fluid (VOF) methods for the dynamics of free boundaries. *J Comput Phys*. 1981;39:201-225.
2. Rider W, Kothe D. Reconstructing volume tracking. *J Comput Phys*. 1998;141(2):112-152.
3. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *J Comput Phys*. 1994;114(1):146-159.
4. Sussman M, Almgren A, Bell J, Colella P, Howell L, Welcome M. An adaptive level set approach for incompressible two-phase flows. *J Comput Phys*. 1999;148(1):81-124.
5. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon Not Royal Astron Soc*. 1977;181:375-389.
6. Morris JP. Simulating surface tension with smoothed particle hydrodynamics. *Int J Numer Methods Fluids*. 2000;33(3):333-353.
7. Koshizuka S, Oka Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl Sci Eng*. 1996;123(3):421-434.

8. Duan G, Koshizuka S, Yamaji A, Chen B, Li X, Tamai T. An accurate and stable multiphase moving particle semi-implicit method based on a corrective matrix for all particle interaction models. *Int J Numer Methods Eng*. 2018;115(10):1287-1314.
9. Yabushita K, Hibi S. A finite volume particle method for an incompressible fluid flow. *Proc Comp Eng Conf*. 2005;10:419-421. in Japanese.
10. Zhang S, Morita K, Fukuda K, Shirakawa N. A new algorithm for surface tension model in moving particle methods. *Int J Numer Methods Fluids*. 2007;55(3):225-240.
11. Nugent S, Posch HA. Liquid drops and surface tension with smoothed particle applied mechanics. *Phys Rev E*. 2000;62(4):4968-4975.
12. Kondo M, Koshizuka S, Suzuki K, Takimoto M. Surface tension model using inter-particle force in particle method. Paper presented at: Proceedings of the ASME/JSME 2007 5th Joint Fluids Engineering Conference; 2007:93-98; San Diego, CA.
13. Guo L, Kawano Y, Zhang S, Suzuki T, Morita K, Fukuda K. Numerical simulation of rheological behavior in melting metal using finite volume particle method. *J Nucl Sci Technol*. 2010;47(11):1011-1022.
14. Liu X, Aramaki Y, Guo L, Morita K. Numerical simulation of gas–liquid–solid three-phase flow using particle methods. *J Nucl Sci Technol*. 2015;52(12):1480-1489.
15. Brackbill JU, Kothe DB, Zemach C. A continuum method for modelling surface tension. *J Comput Phys*. 1992;100:335-354.
16. Hu XY, Adams NA. A multi-phase SPH method for macroscopic and mesoscopic flows. *J Comput Phys*. 2006;213(2):844-861.
17. Adami S, Hu XY, Adams NA. A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation. *J Comput Phys*. 2010;229(13):5011-5021.
18. Zhang M. Simulation of surface tension in 2D and 3D with smoothed particle hydrodynamics method. *J Comput Phys*. 2010;229:7238-7259.
19. Duan G, Koshizuka S, Chen B. A contoured continuum surface force model for particle methods. *J Comput Phys*. 2015;298:280-304.
20. Qi Y, Lu J, Scardovelli R, Zaleski S, Tryggvason G. Computing curvature for volume of fluid methods using machine learning. *J Comput Phys*. 2019;377:155-161.
21. Khayyer A, Gotoh H. Enhancement of stability and accuracy of the moving particle semi-implicit method. *J Comput Phys*. 2011;230(8):3093-3118.
22. Liu X, Morita K, Zhang S. An advanced moving particle semi-implicit method for accurate and stable simulation of incompressible flows. *Comput Methods Appl Mech Eng*. 2018;339:467-487.
23. Khayyer A, Gotoh H. A higher order Laplacian model for enhancement and stabilization of pressure calculation by the MPS method. *Appl Ocean Res*. 2010;32(1):124-131.
24. Liu X, Morita K, Zhang S. A stable moving particle semi-implicit method with renormalized Laplacian model improved for incompressible free-surface flows. *Comput Methods Appl Mech Eng*. 2019;356:199-219.
25. Tamai T, Koshizuka S. Least squares moving particle semi-implicit method. *Comput Part Mech*. 2014;1(3):1-29.
26. Liu X, Morita K, Zhang S. An ALE pairwise-relaxing meshless method for compressible flows. *J Comput Phys*. 2019;387:1-13. accepted.
27. Liu X, Ogawa R, Kata M, Morita K, Zhang S. Accuracy and stability enhancements in the incompressible finite-volume-particle method for multiphase flow simulations. *Comput Phys Commun*. 2018;230:59-69.
28. Tian W, Ishiwatari Y, Ikejiri S, Yamakawa M, Oka Y. Numerical computation of thermally controlled steam bubble condensation using moving particle semi-implicit (MPS) method. *Ann Nucl Energy*. 2010;37(1):5-15.
29. Chen R, Tian W, Su GH, Qiu S, Ishiwatari Y, Oka Y. Numerical investigation on bubble dynamics during flow boiling using moving particle semi-implicit method. *Nucl Eng Des*. 2010;240(11):3830-3840.
30. Duan G, Chen B, Koshizuka S, Xiang SH. Stable multiphase moving particle semi-implicit method for incompressible interfacial flow. *Comput Methods Appl Mech Eng*. 2017;318:636-666.
31. Shimizu Y, Gotoh H, Khayyer A. An MPS-based particle method for simulation of multiphase flows characterized by high density ratios by incorporation of space potential particle concept. *Comput Math Appl*. 2018;76(5):1108-1129.
32. Khayyer A, Gotoh H, Shimizu Y. A projection-based particle method with optimized particle shifting for multiphase flows with large density ratios and discontinuous density fields. *Comput Fluids*. 2019;179:356-371.
33. Liu X, Morita K, Zhang S. Comparative study of two corrective gradient models in the simulation of multiphase flows using moving particle semi-implicit method. *Int J Numer Methods Fluids*. 2020;92(6):573-586.
34. Hysing S, Turek S, Kuzmin D, et al. Quantitative benchmark computations of two–dimensional bubble dynamics. *Int J Numer Methods Fluids*. 2009;60:1259-1288.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Liu X, Morita K, Zhang S. Machine-learning-based surface tension model for multiphase flow simulation using particle method. *Int J Numer Meth Fluids*. 2021;93:356–368. https://doi.org/10.1002/fld.4886