



HAL
open science

Hybridization of front tracking and level set for multiphase flow simulations: a machine learning approach

Ikroh Yoon, Jalel Chergui, Damir Juric, Seungwon Shin

► **To cite this version:**

Ikroh Yoon, Jalel Chergui, Damir Juric, Seungwon Shin. Hybridization of front tracking and level set for multiphase flow simulations: a machine learning approach. *Journal of Mechanical Science and Technology*, 2023, 37 (10), pp.4749-4756. 10.1007/s12206-023-0829-3 . hal-04161702

HAL Id: hal-04161702

<https://hal.science/hal-04161702v1>

Submitted on 13 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hybridization of Front Tracking and Level Set for Multiphase Flow Simulations: A Machine Learning Approach

Ikroh Yoon¹, Jalel Chergui², Damir Juric^{2,3} and Seungwon Shin⁴

¹Korea Institute of Marine Science and Technology Promotion (KIMST), 06775 Seoul, Korea (currently works at Intergovernmental Oceanographic Commission of United Nations Educational, Scientific and Cultural Organization), ²Centre National de la Recherche Scientifique (CNRS), Laboratoire Interdisciplinaire des Sciences du Numérique (LISN), Université Paris Saclay, 91400 Orsay, France, ³Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, Centre for Mathematical Sciences, Wilberforce Road, Cambridge CB3 0WA, UK, ⁴Department of Mechanical and System Design Engineering, Hongik University, 04066 Seoul, Korea

(Received 000 0, 2020; Revised 000 0, 2020; Accepted 000 0, 2020) -please leave blank

Keywords: Multiphase flow; Numerical simulation; Front tracking; Level set; Artificial Intelligence; Machine learning

Correspondence to: Seungwon Shin / sshin@hongik.ac.kr

Abstract In the present study, a machine learning (ML) based approach is proposed to hybridize two well-established methods for multiphase flow simulations, namely the Front Tracking (FT) and the Level Set (LS) methods. Based on the geometric information of the Lagrangian marker elements which represents the phase interface in FT simulations, the distance function field which is the key feature for describing the interface in LS simulations is predicted using a ML model. The trained ML model is implemented in our conventional numerical framework, and we finally demonstrate that the FT-based interface representation can easily and immediately be switched to a LS-based representation whenever needed during the simulation period.

1. Introduction

Multiphase flow phenomena are ubiquitous not only in our everyday life (e.g., falling raindrops or free surfaces of sea waves) but also in many modern technologies such as drops in inkjet printing, cavitation in the ship industry, solidification, boiling flows in power plants to name a few examples [see Ref. 1 and references therein]. Computer simulations for multiphase flows have obviously played critical roles in understanding the complicated physics across various scientific fields, and even in realistic representation of natural phenomena in the art, computer vision, or film industries [2].

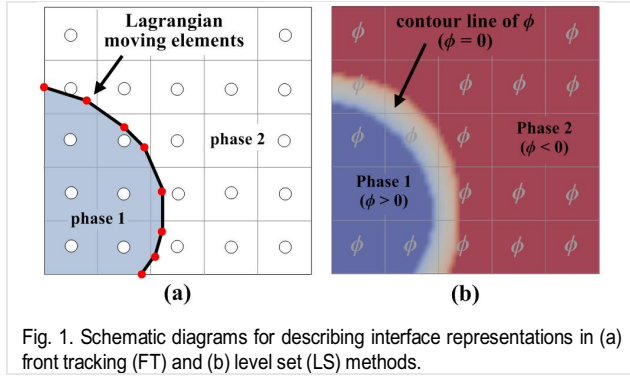
One of the most popular approaches for multiphase flow simulations is solving the governing equations for all phases simultaneously as a single field formulation on a fixed Eulerian mesh [3]. Such methods can generally be categorized into two groups: (i) front tracking and (ii) front capturing. The former utilizes an additional Lagrangian moving grid to represent the phase interface [3], while the latter uses a scalar field to describe the interface (e.g., the color function in the volume of fluid method [4] or the distance function in the level set method [5]).

Since each method has its own pros and cons, many attempts have been made to combine two (or more) different methods. For example, the particle level set method [6], the coupled level set and volume of fluid method [7], and the level contour reconstruction method [8,9] have been proposed and have shown their capability to simulate various multiphase flow phenomena.

However, such advanced hybrid methods still often require complicated geometric calculations or time-consuming processes. For example, the level contour reconstruction method (LCRM) uses an iterative scheme to obtain an exact level set function field from the Lagrangian front tracking information [9].

Data-driven strategies have recently shown their powerful ability across diverse fields including the fluid mechanics communities [10,11], and a few attempts introducing machine-learning (ML) techniques have also appeared in multiphase flow simulation fields. Qi *et al.* [12], Larios-Cárdenas and Gibou [13], and Franca and Oishi [14] utilized ML strategies to compute the interface curvature for volume of fluid (VOF), level set (LS), and front tracking (FT) simulations, respectively. Ataei *et al.* [15] also used ML to replace the conventional iterative computation for the piecewise linear interface construction (PLIC) in VOF simulation. Although a very few studies are available in literature and those are still at initial stages, they have demonstrated the potential of ML strategies in multiphase flow simulations [12-15].

Our main motivation in this study is to examine ML's capability for hybridization of the two different popular simulation methods (FT and LS). We extend our previous work [16] which directly established the LS distance function (scalar) field on the fixed Eulerian grid from the Lagrangian FT elements. Based on ML techniques instead of the conventional and numerical (iterative) approaches, we demonstrate that a FT-based interface representation can easily and immediately be switched to a LS-based representation whenever needed during the simulation period.



2. Learning and interface switching

2.1 Front tracking and level set

We first describe key features of each simulation method. In FT simulation [3], the moving interface is tracked by the Lagrangian mesh elements [see Fig.1(a)]. The use of such an additional mesh enables accurate representation of phase boundaries. However, dealing with the algorithmic connectivity among Lagrangian elements is sometimes a severe computational burden where interfacial shapes are severely deformed, merged, or broken up [3,8].

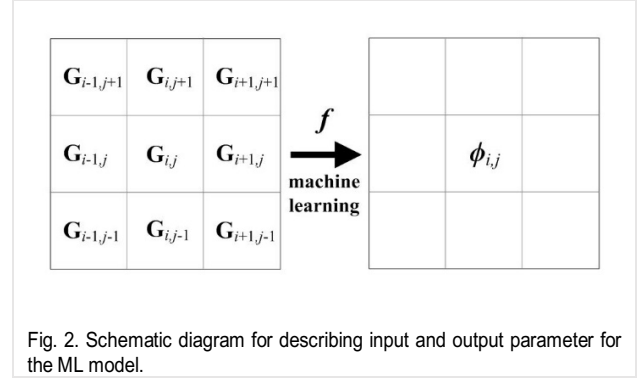
In LS simulation [5], conversely, the distance function ϕ field is used and the contour level of ϕ represents the interface. For example, $\phi > 0$ in one phase and $\phi < 0$ in the other phase, thus the contour level of $\phi = 0$ is regarded as the phase boundary. Since the scalar variable (ϕ) is used, its implementation is relatively straightforward and the topology changes can easily be handled without a Lagrangian approach. A well-known drawback is the numerical diffusion of mass during the advection of the ϕ field, which generally requires additional treatment such as re-initialization of the distance function [5,8].

Conventional iterative numerical techniques for switching from FT-based interface representation to LS-based type are (i) solving a Poisson equation to obtain the indicator function field (which is generally used for describing different physical properties in FT simulation) [3,8] or (ii) direct computation of the minimal distance between Lagrangian interfacial elements and fixed Cartesian grid cells, as in our previous work [9,16]. Since the inverse switching of interface representation (from LS to FT) can be done by linking locations where $\phi = 0$ [8,9,16], our ML approach will primarily be focused on switching from FT based to LS based representation.

2.2 ML strategy

We now describe the data preparation for training our ML model. In FT simulations, the indicator function I is the essential variable for describing material properties for different phases. I has characteristics of the Heaviside function, and can be computed by solving the following Poisson equation:

$$\nabla^2 I = \nabla \cdot \mathbf{G} \quad (1)$$



Here, \mathbf{G} contains geometric information which is directly obtained from the Lagrangian elements. The vector field \mathbf{G} is commonly calculated along the interface (Γ) comprised of many line segments as in [3,8]:

$$\mathbf{G} = \int_{\Gamma} \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_f) dA \quad (2)$$

where \mathbf{x}_f is the position of the centroid of each interface element A , \mathbf{n} is the unit normal vector to A , and dA is the differential area (length of line segments in 2D) of A . $\delta(\mathbf{x} - \mathbf{x}_f)$ is the Dirac delta function, which is nonzero only at the interface (i.e., $\mathbf{x} = \mathbf{x}_f$). Since the Lagrangian interface points, \mathbf{x}_f , do not necessarily coincide with the Eulerian grid points, $\mathbf{x}_{i,j}$, Peskin's immersed boundary method [17] is utilized to pass the information to the Eulerian grid. Thus, the vector field \mathbf{G} can be distributed on the fixed Eulerian grid as [3,8]:

$$\mathbf{G}_{i,j} = \sum_{p=1}^q \mathbf{n} D(\mathbf{x}_{i,j} - \mathbf{x}_f) dA \quad (3)$$

where q is the number of Lagrangian interface elements in the surrounding neighbor cells and p is the element index. $\mathbf{x}_{i,j}$ is the position vector of the Eulerian cell. $D(\mathbf{x}_{i,j} - \mathbf{x}_p)$ is the Dirac delta distribution which varies smoothly but with finite distance near the interface, and can be computed as [8]:

$$D(\mathbf{x}_{i,j} - \mathbf{x}_p) = \frac{\delta(x_p / \Delta x - i) \delta(y_p / \Delta y - j)}{\Delta x \Delta y} \quad (4)$$

$$\delta(r) = \begin{cases} \delta_1(r), & |r| \leq 1 \\ 1/2 - \delta_1(2 - |r|), & 1 \leq |r| \leq 2 \\ 0, & |r| \geq 2 \end{cases} \quad (5)$$

$$\delta_1(r) = \frac{3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}}{8} \quad (6)$$

Here, x_p and y_p are the position of the interface element p in the x - and y - directions, respectively. Δx and Δy are the dimensions

of an Eulerian grid cell. Since \mathbf{G} contains the geometric information of the Lagrangian interface, the distance function ϕ (i.e., the distance from the interface) can be approximated by a function of \mathbf{G} . Hence, we model the distance function ϕ on a given grid cell i, j as:

$$\phi_{i,j} = f \left(\begin{bmatrix} \mathbf{G}_{i-1,j+1} & \mathbf{G}_{i,j+1} & \mathbf{G}_{i+1,j+1} \\ \mathbf{G}_{i-1,j} & \mathbf{G}_{i,j} & \mathbf{G}_{i+1,j} \\ \mathbf{G}_{i-1,j-1} & \mathbf{G}_{i,j-1} & \mathbf{G}_{i+1,j-1} \end{bmatrix} \right) \quad (7)$$

where the unknown function f is now modeled using an ML technique as illustrated in Fig.2. Note that we consider a two-dimensional simulation in the present study for simplicity, but our strategy can be extended to a full three-dimensional simulation in a straightforward way.

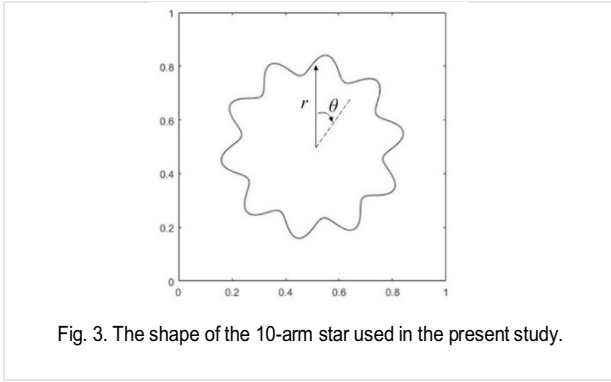


Fig. 3. The shape of the 10-arm star used in the present study.

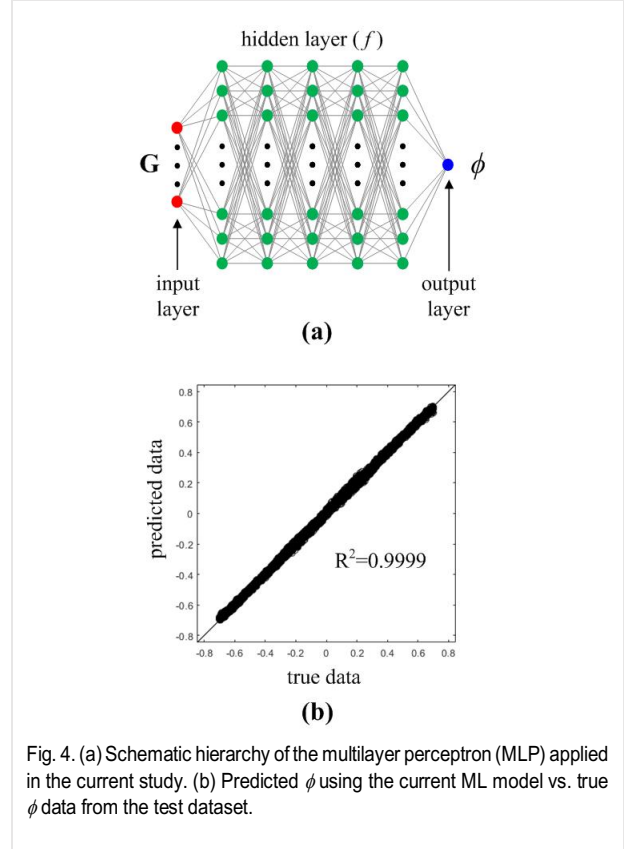


Fig. 4. (a) Schematic hierarchy of the multilayer perceptron (MLP) applied in the current study. (b) Predicted ϕ using the current ML model vs. true ϕ data from the test dataset.

To prepare a synthetic dataset, we use the 10-arm star-like shape (see Fig.3) using the following equation:

$$r(\theta) = R_0 + 0.25R_0 \cos(5\theta)\sin(5\theta) \quad (8)$$

In a 1×1 unit computational domain resolved by a 128×128 grid, the 10-arm star is rotated using the velocity field:

$$\mathbf{u} = \frac{\pi}{314} \begin{bmatrix} 0.5 - y \\ x - 0.5 \end{bmatrix} \quad (9)$$

where the axis of rotation is placed at $(0.5, 0.5)$. To obtain the accurate ϕ fields as the true data from the Lagrangian FT interface elements shown in Fig.3, we use the direct computation and distribution algorithm [16] with the high-order iterative method [9]. Note that this shape contains local curvature variations along both convex and concave interfacial shapes. Note also that \mathbf{G} and ϕ are normalized by the grid size, thus one resolution is sufficient.

A total of 28 cases are used to obtain \mathbf{G} and ϕ data by varying R_0 from 0.1 to 0.3, and an exponential function is applied for varying R_0 to sufficiently sample the smaller R_0 (high curvature) cases [12]. For each R_0 case \mathbf{G} and ϕ data are extracted until θ reaches $\pi/5$, at which the rotated shape becomes equal to the initial shape. 500 time-steps are used to resolve this period ($\theta = 0 - \pi/5$). A total of 10,229,936 data samples are finally obtained, and normalized curvature $h\kappa$ (where h and κ are the grid size

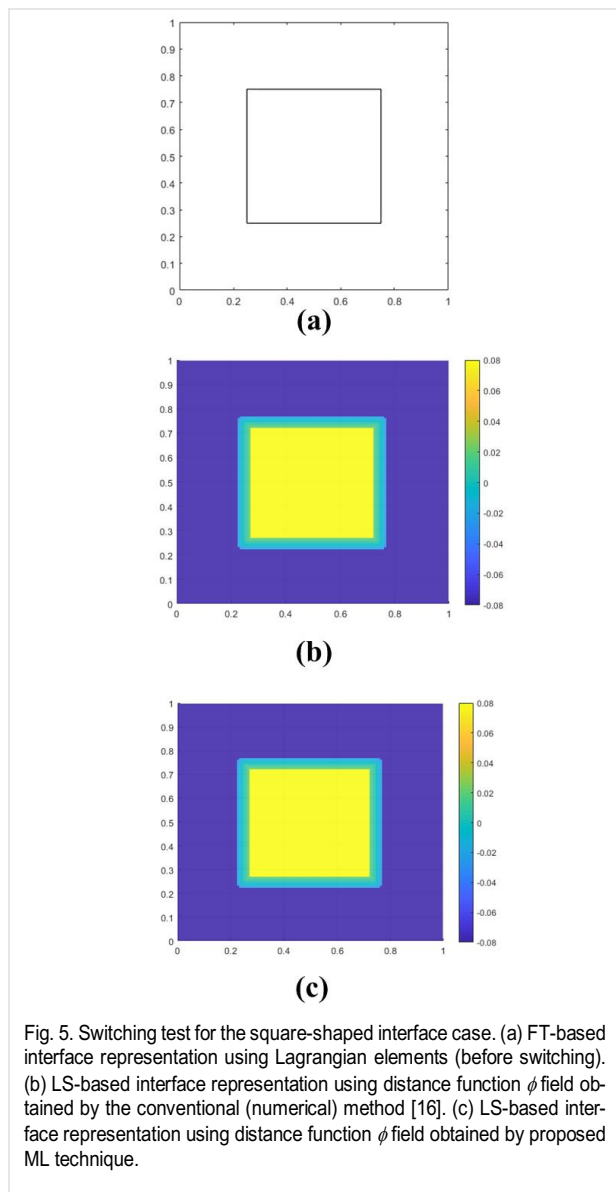


Fig. 5. Switching test for the square-shaped interface case. (a) FT-based interface representation using Lagrangian elements (before switching). (b) LS-based interface representation using distance function ϕ field obtained by the conventional (numerical) method [16]. (c) LS-based interface representation using distance function ϕ field obtained by proposed ML technique.

and the curvature, respectively) is varied from -1.0 to 1.0. Note that this range sufficiently covers the realistic range of the normalized interfacial curvature (0.002–0.5) which appears in multiphase flow simulations [18].

To model the unknown function f , the functional relationship between \mathbf{G} and ϕ , the multilayer perceptron (MLP, also known as multilayer neural network) [19] is applied. Fig.4(a) illustrates the schematic hierarchy of a typical MLP structure, which consists of one input layer, multiple hidden layers, and one output layer. The input features (i.e., \mathbf{G}) are first provided from the input layer to the first hidden layer. Then, linear combinations of the inputs are constructed and forwarded to the next hidden layer after nonlinearization. Such a feedforward procedure is performed over all the hidden layers, and the output layer provides the final output value (ϕ) without nonlinearization. The output value of the m^{th} neuron in the n^{th} layer is computed as:

$$a_m^n = g \left(\sum_{k=1}^{N^{n-1}} w_{mk}^n a_k^{n-1} + b_m^n \right) \quad (10)$$

where N is the number of nodes of each layer, and w_{ml} is the weight between the m^{th} node of the current layer and l^{th} node of the previous layer. b is the bias, and g is the activation function for nonlinearization. w and b are automatically updated during the learning procedure of MLP training by the backpropagation algorithm [20]. After manual search (trial and error approach),

we found that using 3 hidden layers, $N = 60$, and the ReLU function as the activation function work well in terms of accuracy and simplicity. More details on the fundamentals of MLP, its feedforward procedure, and optimization techniques can also be found in Ref.19 and Ref.21.

The prepared dataset is randomly divided into three subsets: (i) the training dataset (70 %, 7,160,956 data samples), (ii) the validation dataset (15 %, 1,534,490 data samples), and (iii) the test dataset (15 %, 1,534,490 data samples). After training of our MLP model, we evaluate the prediction accuracy using the mean square error (MSE):

$$\text{MSE} = \frac{1}{M} \sum_{k=1}^M (\phi_{\text{true}} - \phi_{\text{predicted}})^2 \quad (11)$$

where M is the number of data samples of the test dataset.

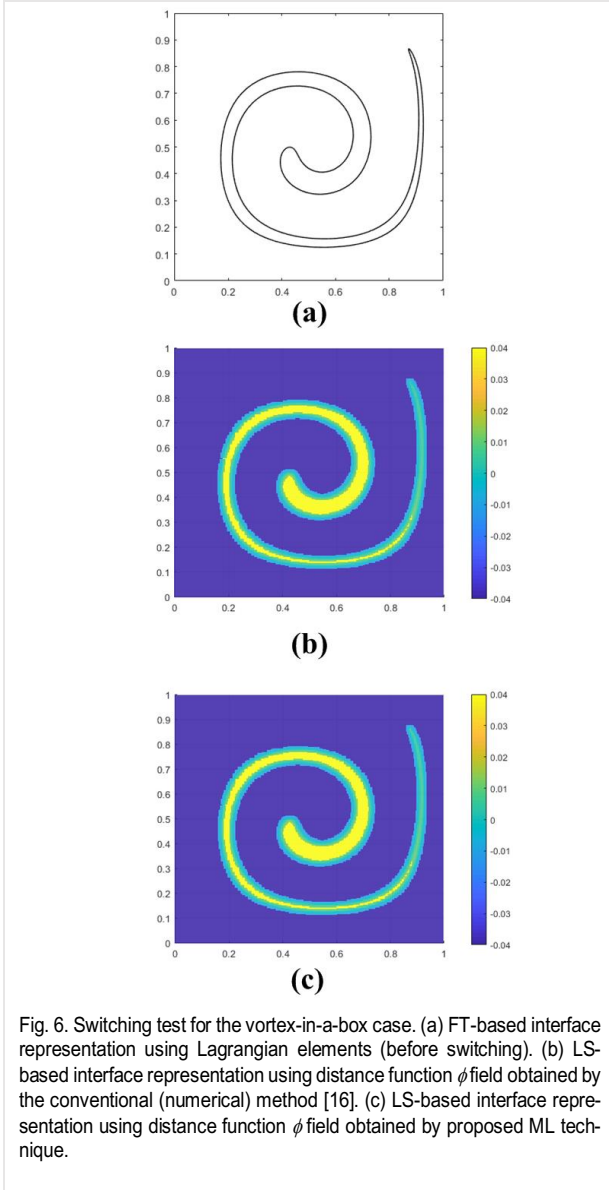


Fig. 6. Switching test for the vortex-in-a-box case. (a) FT-based interface representation using Lagrangian elements (before switching). (b) LS-based interface representation using distance function ϕ field obtained by the conventional (numerical) method [16]. (c) LS-based interface representation using distance function ϕ field obtained by proposed ML technique.

Fig.4(b) shows the prediction accuracy of our final MLP model evaluated from the test dataset. As seen, the trained ML model can predict ϕ very accurately using G . Note that the final MSE and R^2 value are 1.8×10^{-5} and 0.9999, respectively, demonstrating its excellent prediction capability.

2.3 Interface switching based on ML model

We now test the ML-based switching capability from FT based interface representation (i.e., using Lagrangian mesh elements) to LS based (i.e., using the distance function ϕ field). In order to enable switching the interface representation anytime during simulation periods, the feedforward procedure of our trained ML model described above is implemented into our existing in-house FT simulation code.

Fig.5 shows the test result for the square-shaped interface case. A 1×1 unit computational domain resolved by a 128×128

mesh is applied for this test. The length of a side of the square is set to 0.5. Note that this shape contains very extreme local curvatures (a right angle and a straight line). In Fig.5(a), (b), and (c), three different interface representations for the same shape, i.e., FT-based representation (before switching), LS-based representation (after switching, but using conventional iterative technique), LS-based representation (after switching using our proposed ML technique) are depicted.

As seen, the ML-based ϕ field [Fig.5(c)] clearly represents the original FT-based interface representation (see where $\phi = 0$), and also shows good agreement with the ϕ field obtained by the existing method [16]. Note that the practical region where the ϕ field should be precisely evaluated is only near the interface in this type of hybrid simulation because the primary purpose for using the ϕ field is to describe the interface in terms of scalar values. For regions far from the interface, only the sign of ϕ with sufficiently high (or low) values of ϕ are sufficient to distinguish each phase [16].

Fig.6 depicts the test result for the well-known vortex-in-a-box problem [22]. In this case, a circular interface of diameter 0.4 is initially placed at (0.5,0.75), then starts its deformation by the vortical velocity field given as:

$$\mathbf{u} = \begin{bmatrix} -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos(\pi t / 8) \\ 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x)\cos(\pi t / 8) \end{bmatrix} \quad (12)$$

Note that this case contains a very severely deformed interfacial shape, thus the 1×1 unit computational domain is now resolved with a 256×256 mesh. The interfacial shapes at $t = 1.875$ are compared in Fig.6. The LS distance function ϕ field obtained by the ML model [Fig.6(c)] still clearly represents the original FT-based interface boundaries (see where $\phi = 0$) and also shows good agreement with the ϕ field computed by the existing iterative scheme [16], even for such a highly stretched case. To quantitatively check the switching accuracy, we further compare volume loss from the initial circular interface for those two cases [Fig.6(b) and (c)]. The volume losses due to the switching operations are 0.050% [Fig.6(b)] and 0.053% [Fig.6(c)], showing very good agreement again.

We finally test our ML-based switching capability incorporating the flow solvers, thus the governing equations for incompressible two-phase flows are now considered together. The numerical procedure and solution techniques for the governing equations, surface tension force, discretization and other detailed information can be found in Shin and Juric [8,9].

A 2D drop oscillation problem in zero gravity is considered. In a doubly periodic domain of $[-10,10]^2$ resolved by a 128×128 mesh, the initial shape of the droplet is given as $x^2/9 + y^2/4 = 1$. The density of the droplet phase and the ambient phase are set to $\rho_1 = 1.0$ and $\rho_2 = 0.01$, respectively, whereas the viscosities are $\mu_1 = 0.01$ and $\mu_2 = 5 \times 10^{-5}$. The surface tension coefficient is set to $\sigma = 1$. Note that all variables are nondimensionalized, and more detailed simulation settings and conditions for this test can also be found in Ref.8 and Ref.23.

To check its switching capability more quantitatively and also to examine its applicability to practical multiphase flow simulations, we now reconstruct the Lagrangian FT interface elements using the ϕ field obtained by both the conventional iterative numerical method [16] and the ML technique during the simulation period. Since the contour level $\phi = 0$ denotes the phase interface in the LS simulation, FT elements can be reconstructed by linking those points where $\phi = 0$ based on the existing LCRM algorithm [8,9]. Therefore, comparison between the original FT elements (before switching from FT to LS) and the reconstructed elements (after inverse-switching from LS to FT) can measure how accurately the ML-based switching works because those two interfaces should ideally be identical to each other.

The initially stretched droplet starts retracting due to the presence of the surface tension force. Then, the droplet continues oscillatory motions until all kinetic energy is dissipated by viscous damping. The simulation is performed until $t = 200$, and 100 switching-reconstruction operations are performed during this simulation period ($0 \leq t \leq 200$). In Fig.7(a), the droplet interfacial shapes are plotted for two different time instants ($t = 6.0$ and 11.0) where the droplet reaches the first maximal stretching state in the vertical and the horizontal directions, respectively. As seen, the reconstructed interface using the ϕ field obtained by the proposed ML-based simulation (see red lines) is sufficiently identical to the original FT interface (see black lines) as well as the interface reconstructed using the conventional iterative method (see blue lines) [16]. Those three interfaces are practically very difficult to distinguish since they are almost exactly superposed, thus showing excellent switching capability of the proposed ML model.

In Fig.7(b), we plot the droplet kinetic energies obtained by three simulations, i.e., the case where no switching procedure is performed (pure FT simulation) and two cases where 100 switching-reconstruction procedures are performed using the current ML technique and the conventional iterative method [16]. The computed kinetic energies from these droplets are almost identical. In particular, the results of two hybrid simulations (red and blue lines) show excellent agreement. A minor deviation from the original FT simulation is presumably caused by the interpolation for very frequent switching-reconstruction operations (100 times).

Last but not least, we discuss the fundamental characteristics and expected benefits of the proposed ML strategy. Computational efficiency for the switching-reconstruction procedure using the ML technique can be greatly improved compared to the conventional iterative numerical method. For the problem shown in Fig.5 above, the computational time for 100 switching-reconstruction operations by the ML technique is measured as 1.9 seconds which accounts for only 12.4 % of that from the conventional iterative method (15.4 seconds). Also, for the problem shown in Fig.6, the computational time is reduced to 27.3 % if the ML technique is used. Those computational times have been evaluated using the Fortran intrinsic function "cpu_time" and a system equipped with a 4-core Intel® Core(TM) i7-7700

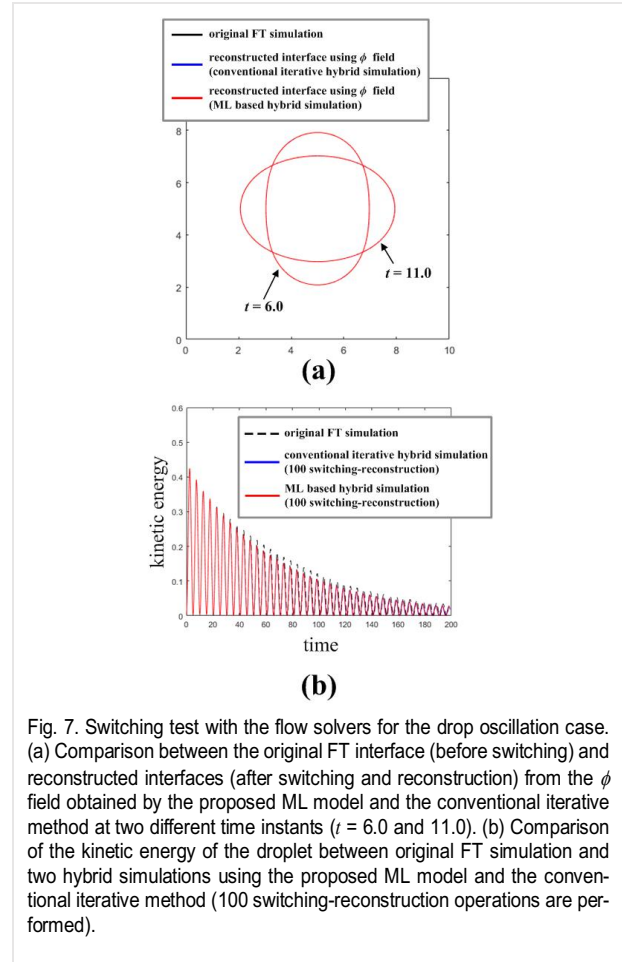


Fig. 7. Switching test with the flow solvers for the drop oscillation case. (a) Comparison between the original FT interface (before switching) and reconstructed interfaces (after switching and reconstruction) from the ϕ field obtained by the proposed ML model and the conventional iterative method at two different time instants ($t = 6.0$ and 11.0). (b) Comparison of the kinetic energy of the droplet between original FT simulation and two hybrid simulations using the proposed ML model and the conventional iterative method (100 switching-reconstruction operations are performed).

CPU 3.60GHz processor. Although a typical comparison of those two approaches based on the number of numerical additions and multiplications per cell is not straightforward since the two grid systems (Eulerian and Lagrangian) should be considered together, it is evident that the proposed ML technique is much more efficient. In our ML approach, only the Eulerian variable G is used and only simple matrix multiplications and additions are needed in the feedforward procedure to calculate the LS distance function ϕ field, whereas the conventional iterative method should perform numerous iterative operations and complicated geometric calculations dealing with information exchange between two different grid systems [9,16]. This characteristic of the ML approach (i.e., use of only the Eulerian variable G) can enable much easier implementation and ideal load balancing for parallel computations as well.

We also expect that the proposed ML model can be applied to the typical grid resolution levels of FT and similar types of simulations because our dataset sufficiently covers a practical range of interfacial curvature which usually appears in multiphase flow simulations. However, if flow phenomena are beyond the typical grid resolution levels used in FT simulations (e.g., very small bubbles having sizes comparable to the Eulerian grid), neither the current ML approach nor the general FT simulation may be sufficient and different simulation methods

should be considered.

3. Conclusions

In this study, the machine learning (ML) strategy is proposed to hybridize two well-established methods for multiphase flow simulations: (i) the Front Tracking (FT), and (ii) the Level Set (LS) methods. The simple ML model is found to predict the LS distance function ϕ field very easily and accurately using the geometric information vector \mathbf{G} of the FT simulation. It is further demonstrated that the FT-based interface representation can easily and immediately be switched to LS-based representation during the simulation period and its inverse switching operation can also be incorporated using the existing numerical algorithm.

Although the present study is still confined to 2D simulations, the key features of the current strategy can be extended to 3D in a straightforward way (we are currently working on this issue). In addition, more detailed analysis of the switching capability using ML techniques such as (global and local) switching accuracy, its dependence on the grid resolution, and computational efficiency depending on various simulation cases, can be further investigated to examine more practical applicability.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (2020R1A2C1003822) and support through computing time at the Institut du Developpement et des Ressources en Informatique Scientifique (IDRIS) of the Centre National de la Recherche Scientifique (CNRS), coordinated by GENCI (Grand Equipement National de Calcul Intensif) Grant 2022 A0142B06721.

Nomenclature


| | |
|--------------|---|
| \mathbf{G} | : Geometric information vector in FT simulation |
| ϕ | : Distance function in LS simulation |

References

- [1] C. E. Brennen, *Fundamentals of Multiphase Flows*, Cambridge University Press, Cambridge, UK (2005).
- [2] F. Gibou, D. Hyde, and R. Fedkiw, Sharp interface approaches and deep learning techniques for multiphase flows, *Journal of Computational Physics*, 380 (2019) 442-463.
- [3] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, A Front-Tracking Method for the Computations of Multiphase Flow, *Journal of Computational Physics*, 169 (2) (2001) 708-759.
- [4] C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics*, 39 (1) (1981) 201-225.
- [5] S. Osher and R. P. Fedkiw, Level Set Methods: An Overview and Some Recent Results, *Journal of Computational Physics*, 169 (2) (2001) 463-502.
- [6] D. Enright and R. Fedkiw, J. Ferziger, and I. Mitchell, A Hybrid Particle Level Set Method for Improved Interface Capturing, *Journal of Computational Physics*, 183(1) (2002) 83-116.
- [7] M. Sussman and E. G. Puckett, A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows, *Journal of Computational Physics*, 162 (2) (2000) 301-337.
- [8] S. Shin and D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, *Journal of Computational Physics*, 180 (2002) 427-470.
- [9] S. Shin and D. Juric, High-order level contour reconstruction method, *Journal of Mechanical Science and Technology*, 21 (2) (2007) 311-326.
- [10] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine Learning for Fluid Mechanics, *Annual Review of Fluid Mechanics*, 52 (2020) 477-508.
- [11] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence Modeling in the Age of Data, *Annual Review of Fluid Mechanics*, 51 (2019) 357-377.
- [12] Y. Qi, J. Lu, R. Scardovelli, S. Zaleski, and G. Tryggvason, Computing curvature for volume of fluid methods using machine learning, *Journal of Computational Physics*, 377 (2019) 155-161.
- [13] L. Á. Larios-Cárdenas and F. Gibou, A Deep Learning Approach for the Computation of Curvature in the Level-Set Method, *SIAM Journal on Scientific Computing*, 43 (3) (2021) A1754-A1779.
- [14] H. L. França and C. M. Oishi, A machine learning strategy for computing interface curvature in Front-Tracking methods, *Journal of Computational Physics*, 450 (2022) 110860.
- [15] M. Ataei, M. Bussmann, V. Shaayegan, F. Costa, S. Han, and C. B. Park, NPLIC: A machine learning approach to piecewise linear interface construction, *Computers & Fluids*, 223 (2021) 104950.
- [16] S. Shin and D. Juric, A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques, *International journal for numerical methods in fluids*, 60 (2009) 753-778.
- [17] C. S. Peskin, Numerical analysis of blood flow in the heart, *Journal of Computational Physics*, 25(3) (1977) 220-252.
- [18] H.V. Patel, A. Panda, J.A.M. Kuipers, and E.A.J.F. Peters, Computing interface curvature from volume fractions: A machine learning approach, *Computers & Fluids* 193 (2019) 104263.
- [19] F. Rosenblatt, The Perceptron, a perceiving and recognizing automaton project para, Report Vol. 85, Nos. 460-461, *Cornell Aeronautical Laboratory*, 1957.
- [20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Neurocomputing: Foundations of Research, Learning Representations by Back-propagating Errors, *MIT Press*, Cambridge, USA (1988).
- [21] M. A. Nielsen, Neural Network and Deep Learning, *Determination Press* (2015).

- [22] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for two-phase flow consisting of separate compressible and incompressible regions. *Journal of Computational Physics* 85 (1989) 257-283.
- [23] D.J. Torres and J.U. Brackbill, The Point-Set Method: Front-Tracking without Connectivity, *Journal of Computational Physics*, 165(2) (2000) 620-644.

Author information

| | |
|---|---|
|  | <p>Ikroh Yoon received his Ph.D. degree from Hongik University in Seoul, Korea, in 2020. Dr. Yoon is currently a principal researcher and a programme specialist at Korea Institute of Marine Science and Technology Promotion (KIMST) and United Nations Educational, Scientific and Cultural Organization, respectively.</p> |
|  | <p>Jalel Chergui received his Ph.D. degree from Paris-Sorbonne University at Paris VI, France. Dr. Chergui is currently a senior Research Engineer in Computer Methods and Applied Fluid Mechanics at Centre National de la Recherche Scientifique (LISN/CNRS).</p> |
|  | <p>Damir Juric received his Ph.D. degree from the University of Michigan in 1996. Dr. Juric is currently a senior researcher at the Centre National de la Recherche Scientifique (CNRS) in France as well as a Visiting Fellow at Churchill College, University of Cambridge in the Department of Applied Mathematics and Theoretical Physics (DAMTP).</p> |
|  | <p>Seungwon Shin received his Ph.D. degree from Georgia Tech in 2002. Dr. Shin is currently a Professor at the School of Mechanical and System Design Engineering at Hongik University in Seoul, Korea.</p> |