Rising-Bubble PINN Simulation
Progress Report on 2024 Summer

Woojeong Kim

August 15, 2024

1 Problem Setting

In this report, we will introduce several strategies to enhance the approximating capability of the Physics Informed Neural Network(PINN). We want to derive the result by applying the the enhanced PINN to solve more complex bubble dynamics.
Despite significant advances in the numerical simulation of incompressible interfacial flows, it is surprising that no rigorous, quantitative numerical benchmark has been proposed for validating and comparing interfacial flow codes. Every year, new and improved methods are developed for simulating mixtures of immiscible fluids undergoing complex changes. This lack of benchmarks contrasts sharply with other areas of computational fluid dynamics (CFD), where dedicated benchmarks are well-established and widely accepted.
As our aim, we simulate and study the behavior of the Abels–Garcke–Grün Navier–Stokes–Cahn–Hilliard diffuse-interface model and the sharp-interface model for binary-fluid flows as the diffuse-interface thickness approaches zero. The interface-thickness parameter represents the transverse length scale of the transition layer between the two fluid components, which is expected to collapse into a manifold of co-dimension one in the sharp-interface limit ($\epsilon \to 0$).
Mathematical and physical models for binary-fluid flows generally are generally classified as one of two categories: sharp-interface or diffuse-interface models. In sharp-interface models, the surface that separates the two fluid components is explicitly represented by a manifold of co-dimension one. This manifold carries kinematic and dynamic interface conditions, acting as boundary conditions on the initial boundary-value problems of the two contiguous fluid components, and determines the evolution of the manifold. Therefore, sharp-interface models are of the free-boundary type.

In diffuse-interface models, the interface between the two fluid components is represented as a thin but finite transition layer, where the two components are mixed in a proportion that varies continuously and monotonically between the two pure species across the layer. The strength of diffuse-interface models lies in their intrinsic ability to account for topological changes of the fluid-fluid interface due to coalescence or break-up of droplets or wetting, such as the propagation of the fluid-fluid front along a (possibly elastic) solid substrate.
Diffuse-interface models for two immiscible incompressible fluid species are generally described by the Navier–Stokes–Cahn–Hilliard (NSCH) equations. NSCH models invariably contain three parameters related to the diffuse interface: an interface-thickness parameter ϵ, a mobility parameter m, and a surface-tension parameter σ. The mobility parameter is responsible for the rate at which phase diffusion occurs in the vicinity of the diffuse interface. However, the details of how the diffuse-interface solution approaches the sharp-interface limit solution for various admissible scalings of the mobility are not currently known. Different scalings have been proposed in the literature, particularly in the context of numerical simulation approaches.

For the diffuse-interface model to approach a classical sharp-interface model in the limit $\epsilon \to 0$, the mobility parameter m in the diffuse-interface model must scale appropriately with the interface-thickness parameter ϵ.

Numerous articles tried to address the open question of the optimal scaling of the mobility parameter in terms of the convergence rate of the diffuse-interface solution to the sharp-interface solution.
In this report, we will use the "picture norm", which is commonly used to validate interfacial flow codes by qualitatively comparing interface shapes to other numerical experiments or reference diagrams. However, this approach is insufficient to confirm the accuracy of numerical solutions to the

Navier-Stokes equations. For instance, bubble shapes generated by different codes with the same problem setup should ideally be identical, but often they are not. Therefore, more precise metrics are needed to directly measure convergence and ensure correctness.

The simulation of two-phase fluid systems using the Navier-Stokes-Cahn-Hilliard (NSCH) equations presents significant challenges, particularly when dealing with moving boundaries. One prominent example is the rising-bubble system in two fluids, where the density discontinuity at the moving boundary can cause blow-up gradients in the neural network forward pass, complicating the training process of Physics-Informed Neural Networks (PINNs). We have several strategies to overcome the discontinuity difficult. This report explores the difficulties encountered in such simulations and proposes a method to overcome these challenges using gradient clipping and adaptive loss weighting. For the adaptive loss weighting, we use the level set method, embedding the bubble boundary interface, which is a curve in 2D or a surface in 3D, within a higher-dimensional function.

Level set methods are extensively used for capturing the evolution of interfaces, particularly during extreme topological changes such as merging or pinching off. Their application spans a wide range of fields, including fluid mechanics, combustion, computer vision, and materials science. These uses are thoroughly discussed in recent review articles as [4].

The success of level set methods, and other Eulerian methods, is largely due to the role of curvature in numerical regularization. However, as highlighted in the paper[7], the level set function can quickly deviate from being a signed distance function, especially during extreme topological changes.

1.1 Governing Equations

We will analyse a binary-fluid system in which both fluids are characterized as incompressible, isothermal, immiscible, and Newtonian with finite viscosity.

1.1.1 Diffuse-interface Governing Equations

Let us call droplet fluid as fluid 2 and background fluid as fluid 1. The two immiscible fluids in diffuse-interface models are separated by a droplet surface layer of finite thickness made up of a mixture of the two fluids. This layer contains a gradual transition between fluid 1 and fluid 2. Time interval is defined as $(0, t_{fin})$ within the positive real numbers and a spatial domain is defined as a simply connected, fixed subset Ω within R^d (where $d = 2$ or 3).

We take order parameter ϕ in an NSCH model to linearly describe changes of density over time. The order parameter ϕ can range from -1 to 1; it is 1 for pure fluid 2, -1 for pure fluid 1, and between -1 and 1 for a mixture of the two fluids. This model also includes the velocity and pressure of the fluid mixture. From the paper by Abels, Garcke, and Grün (See [1]), we use the following NSCH model.

$$\begin{cases} \partial_t(\rho u) + \nabla \cdot (\rho u \otimes u) + \nabla \cdot (u \otimes J) + \nabla p - \nabla \cdot \tau_v + \nabla \cdot \zeta = \rho g, \\ \nabla \cdot u = 0, \\ \partial_t \phi + \nabla \cdot (\phi u) - \nabla \cdot (m \nabla \mu) = 0, \\ \mu + \sigma\epsilon\Delta\phi - \dfrac{\sigma}{\epsilon}\Psi' = 0, \end{cases} \tag{1}$$

when the unknown variables in the model are the volume-averaged velocity u, the pressure p, the order parameter ϕ, and the chemical potential μ. The parameter g is the external gravitational force field, set as 0.98. And the other parameters, the relative mass flux J, the viscous stress τ_v, the capillary stress ζ, and the mixture energy density Ψ, are given as:

$$\begin{aligned} J &:= m\frac{\rho_1 - \rho_2}{2}\nabla\mu, \\ \tau_v &:= \eta(\nabla u + (\nabla u)^T), \\ \zeta &:= -\sigma\epsilon\nabla\phi \otimes \nabla\phi + I\left(\frac{\sigma\epsilon}{2}|\phi|^2 + \frac{\sigma}{\epsilon}\Psi\right), \\ \Psi(\phi) &:= \frac{1}{4}(\phi^2 - 1)^2. \end{aligned} \tag{2}$$

The remaining model parameters are the mobility parameter $m > 0$ and the interface-thickness parameter $\epsilon > 0$. The m affects the time and the ϵ affects length scales. Also the material parameters are

σ where $2\sqrt{2}\sigma = 3\sigma_{DA}$ and a rescaling of the droplet-ambient surface tension $\sigma_{DA} = 1$, the mixture density ρ and the mixture viscosity η. The ρ and η depend on ϕ in general cases. The viscosity can be balanced with the viscosity interpolation, which is applied in the same way of the Arrheninus mixture-viscosity model (See [2])

$$log\eta(\phi) = \frac{(1+\phi)\Lambda log\eta_2 + (1-\phi)log\eta_1}{(1+\phi)\Lambda + (1-\phi)} \tag{3}$$

when the intrinsic volume ratio is defined as $\Lambda = \rho_2 M_1/\rho_1 M_2$ between the two fluids where M_1 and M_2 are respective molar masses for each.

1.1.2 Sharp-interface Governing Equations

Taking the interface tickness $\epsilon \to 0$, we can see that the width of the diffuse interface in the NSCH model diminishes to zero. This new model that results from this limit, as discussed in the introduction, depends on how the mobility m is scaled. In this way, the classical sharp-interface model is obtained where the mobility m appropriately approaches zero. Therefore, the NSCH equations with the conservation of momentum and mass is expressed as

$$\begin{cases} \rho(x)\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) = -\nabla p + \eta(\phi)\Delta u + \rho(x)g \\ \nabla \cdot u = 0, \\ [u]|_\Gamma = 0, \\ [pI + \eta(\nabla u + (\nabla u)^T)]|_\Gamma \cdot n = \sigma_{DA}\kappa n \end{cases} \tag{4}$$

where a space-time domain $\Omega \times [0,T]$ is fixed with $\Omega \in \mathbf{R}^2$, n is the unit normal at the interface pointing into Ω_1, $[A]|_\Gamma = A|_{\Omega_1 \cap \Gamma} - A|_{\Omega_2 \cap \Gamma}$ defines the jump of a quantity A across the interface Γ and κ is the curvature of the interface Γ. Fluid 1 is assumed to occupy domain Ω_1 and entirely surround fluid 2, which occupies Ω_2. We also assume that the surface interface as $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$ and whole domain as $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$ where $\partial\Omega_2 \cap \partial\Omega = \emptyset$.
The effects of surface tension are taken into account on the last two equations in (4), and thus the surface tension force is in the force balance at the interface Γ. The third equation implies that the velocity is continuous across the interface Γ. In contrast, the forth equation describes the force balance on the interface Γ.

1.2 Initial Configuration

The initial configuration, depicted in the Figure 1, is applied to actual training. The features shows a circular bubble with a radius of $r_0 = 0.25$, centered at $[0.5, 0.5]$ within a rectangular domain of $[1 \times 2]$. The bubble's density is less than that of the surrounding fluid, i.e.$\rho_2 < \rho_1$. The top and bottom boundaries use a no-slip condition $u = 0$, while the vertical walls use a free-slip condition $u \cdot n = 0, \tau \cdot (\nabla u + (\nabla u)^T) \cdot n = 0$, with τ being the tangential vector.

1.3 The Training Set Up Cases

As we mentioned before in introduction and problem setting chapter, we will use the "picture norm' as simulation result representation. Typically, the "picture norm" is used to validate interfacial flow codes. This involves qualitatively comparing interface shapes to other numerical experiments or ensuring that the bubble shape falls within the correct region according to the Clift et al. [6]. However, this method is inadequate for verifying if the numerical solution is the correct solution to the Navier-Stokes equations. For example, bubble shapes shown in Figure 1, calculated by five different codes with identical problem formulations, should ideally be identical. Unfortunately, they are not, and while the shapes are similar, it is unclear which, if any, are correct. Therefore, we must move beyond the "picture norm" and establish precise metrics to directly measure convergence.
The difficulty arise from this point. Even though there are NSCH equation system to express the movement of bubble, especially in the sharp interface model with free boundary, we have four separate part to analyze the physical status, i.e. interior of each two fluids, free boundary for the moving

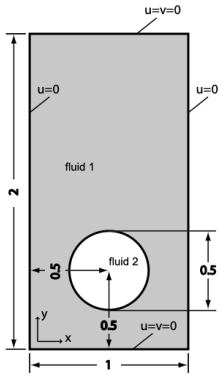

Figure 1: Initial Configuration Figure

Test Case	ρ_1	ρ_2	μ_1	μ_2	g	σ_{DA}	ρ_1/ρ_2	μ_1/μ_2
Case on Sharp Interface Model	1000	100	10	1	0.98	24.5	10	10

Table 1: Parameters on each training

interface of bubble and boundary part. Though most of simulations are working well on continuous variable to deal with nature movement, in this separate regions, the simulation must go through the decreasing loss procedure on each separate part of these with discontinuous variables on free boundary. This means that the explicit coordinate of the moving boundary according to change of time t is not clearly elucidated yet. Therefore the movement of bubble interface is vague for physicians and mathematicians.

Of course, this makes the experimental simulation for the sharp interface model makes harder than the diffuse interface model. Especially the discontinuous variables are emerged from the difference of density. So the density, pressure and viscosity variables in the sharp interface model are discontinuous on the free boundary, making the experiment more difficult(See [5]).

On training test for a case based on Diffuse interface or Sharp interface model, each test case is classified from the density ratio ρ_1/ρ_2 and viscosity ratio μ_1/μ_2. As we mentioned before, the initial radius of the bubble is r_0 and the gravitational constant is g.

On the below Table 1, it details the fluid and physical parameters that define the test cases, which are tracked for the time range from 0 to 1. The test case involves a rising bubble with density and viscosity ratios of 10. Experimental simulation in [6] by Clift et al. indicate that such a bubble will end up in the ellipsoidal organization. With assumption of this two-dimensional bubble, surface tension effects should be considered as strong enough to hold the bubble together, preventing any breakup in this scenario.

1.4 Level Set Method for Interface Tracking

The fundamental concept of level set methods involves representing an interface Γ in R^3, which bounds an open region $\Omega \subset R^3$, as the zero level set of a higher-dimensional function $\varphi(\mathbf{x}, t)$. The Γ is represented as a curve in 2D or a surface in 3D, within a higher-dimensional function φ. This can be formulated as:

$$\Gamma(t) = \{x \in R^d \mid \varphi(x, t) = v_{ls}\}$$

This scalar function φ in R^3 simplifies the description of the interface, particularly during topological changes like pinching and merging (refer to 'A hybrid particle level set method for improved interface capturing').

Here, v_{ls} is the contour level or isosurface value that clearly defines the interface. Although v_{ls} can be chosen arbitrarily, it is usually set to zero, enabling the identification of different phases by the sign of φ. It is advisable to prescribe (or at least initialize) φ as a signed distance function.

Employing a distance function has two primary benefits: it is generally smooth and facilitates the construction of regularized Heaviside and delta functions. Techniques involving local grid adaptation

and grid deformation can utilize the distance function to pinpoint where grid refinement is necessary. So, in our case, the density ρ can play this role with linear translation, since its values are 1000 on fluid 1 and 100 on fluid2. Moreover, normal and curvature fields are globally defined as:

$$\hat{n} = \frac{\nabla \varphi}{|\nabla \varphi|}, \quad \kappa = \nabla \cdot \hat{n}$$

Geometric properties such as the unit normal and curvature can be derived from the level set function as above. Spatial derivatives in these calculations are typically determined using standard second-order accurate central differencing when denominators are nonzero; otherwise, one-sided differencing is applied.

An initial configuration of a level set function initialized as a signed distance function, with Γ depicted as the initial circle with radius r_0, can be seen in figure of the chapter 1.2.

To derive an evolution equation for the level set field φ, it is crucial to recognize that the following condition must be met for the moving interface, which is considered here as the zero level set:

$$\varphi(x(t), t) = 0$$

While the particle level set method has not been extended to handle more complex flows involving geometry, such as motion normal to the interface or motion by mean curvature, it can straightforwardly address the re-initialization equation. This is because the exact solution requires that both the interface (zero level set) and marker particles remain stationary.

2 Physics Informed Neural Networks and Improvement

Check : Refer the past 5.10 report !!!Merit of D: Easy to simulate -¿ Simulate S by applying the skills in D (For initial condition, use tanh instead of strict -1(0), 1)
We will consider the physics informed neural network as reiview. And, several strategies are introduced to improve the accuracy and approximation level of the PINN.

2.1 Physics Informed Neural Networks

We clearly set the specific system to test 2-dimensional rising bubble governing equations based on the previous chapter.

The diffuse interface model is

$$\begin{cases} \partial_t(\rho u) + \nabla \cdot (\rho u \otimes u) + \nabla \cdot (u \otimes J) + \nabla p - \nabla \cdot \tau_v + \nabla \cdot \zeta = \rho g, \\ \nabla \cdot u = 0, \\ \partial_t \phi + \nabla \cdot (\phi u) - \nabla \cdot (m \nabla \mu) = 0, \\ \mu + \sigma \epsilon \Delta \phi - \dfrac{\sigma}{\epsilon} \Psi' = 0, \end{cases} \tag{5}$$

where $(x, y) \in [0, 1] \times [0, 2]$ and $t \in [0, 1]$.

The sharp interface model is set as follows. First, for the NS equation with divergence-free equation is

$$\begin{cases} \rho(x) \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla p + \nabla \cdot (\eta(\nabla u + (\nabla u)^T)) + \rho(x) g \\ \nabla \cdot u = 0, \end{cases} \tag{6}$$

where $(x, y) \in [0, 1] \times [0, 2]$ and $t \in [0, 1]$. Secondly, the moving(free) boundary equations are

$$\begin{cases} [u]|_\Gamma = 0, \\ [pI + \eta(\nabla u + (\nabla u)^T)]|_\Gamma \cdot n = \sigma_{DA} \kappa n \end{cases} \tag{7}$$

where $(x, y) \in \Gamma$ and $t \in [0, 1]$. Third, the no-slip boundary condition on top and bottom boundaries and free-slip condition on the vertical walls are

$$\begin{cases} u(x, 0, t) = u(x, 2, t) = 0, \\ u(0, y, t) \cdot n = 0, \\ u(1, y, t) \cdot n = 0, \\ \tau\dot{(}\nabla u(0, y, t) + (\nabla u(0, y, t))^T) \cdot n = 0, \\ \tau\dot{(}\nabla u(0, y, t) + (\nabla u(0, y, t))^T) \cdot n = 0 \end{cases} \tag{8}$$

where $(x, y) \in [0, 1] \times [0, 2]$ and $t \in [0, 1]$.

Now, from the above systems, we can set the input variables and output variables of the neural network as the beginning step of the PINN simulation.
Input variables are same as x, y and t. In the sharp interface system, output variables are 2-dimensional velocity u, pressure P, density ρ and viscosity η. They are defined as u-network, P-network, ρ-network and η-network respectively:

$$\begin{aligned} &u : (x, y, t) \to u(x, y, t) = 0, && P : (x, y, t) \to P(x, y, t), \\ &\rho : (x, y, t) \to \rho(x, y, t) = 0, && \eta : (x, y, t) \to \eta(x, y, t). \end{aligned} \tag{9}$$

In the diffuse interface system, output variables are also 2-dimensional velocity u, pressure P, order parameter ϕ and chemical potential μ. They are defined as u-network, P-network, ϕ-network and μ-network respectively:

$$\begin{aligned} &u : (x, y, t) \to u(x, y, t) = 0, && P : (x, y, t) \to P(x, y, t), \\ &\phi : (x, y, t) \to \phi(x, y, t) = 0, && \mu : (x, y, t) \to \mu(x, y, t). \end{aligned} \tag{10}$$

With the output variables above, equations in diffuse interface system (5) and in sharp interface system (6)-(8) have error values after computation. These errors from all the equations can be measured by various measurement methods. We employed Binary Classification Entropy loss function for the initial configuration condition equations and Mean Squared Error loss function for the remaining equations. So, the each loss function on the both models for the PINN are defined as the adding of all the loss from each equation in the model.

$$\begin{aligned} &loss_{\text{Diffuse}} = \\ &\quad W_{\text{Initial Condition}} BCE_{\text{Initial Condition}} + W_{\text{NS PDE}} MSE_{\text{NS PDE}} + W_{\text{Boundary}} MSE_{\text{Boundary}} \\ &loss_{\text{Sharp}} = \\ &\quad W_{\text{Initial Condition}} BCE_{\text{Initial Condition}} + W_{\text{NSCH PDE}} MSE_{\text{NSCH PDE}} + \\ &\quad W_{\text{Boundary}} MSE_{\text{Boundary}} + W_{\text{Moving Boundary}} MSE_{\text{Moving Boundary}} \end{aligned} \tag{11}$$

where W is the constant weight for each loss term.

2.2 Enhancing the Accuracy of Physics Informed Neural Networks

Enhancing the accuracy of Physics- Informed Neural Networks While the Physics - Informed Neural Network (PINN) demonstrates effectiveness in solving the equation, its direct application to phase field equations often fails to provide accurate solutions in various scenarios. To address this limitation, we propose several strategies to enhance the approximation capability of the PINN.

2.3 Choosing the Loss Function and Coefficients on Each Equations of System

A straightforward technique to enhance the learning capability of the PINN for phase field models involves adding weights to the loss function. This approach is motivated by the dissipate nature

of phase field equations, implying irreversibility. For instance, the Allen-Cahn equation, like other reactive diffusion equations, can only be solved in the forward time direction. Consequently, if the PINN struggles to learn the solution at time E=t, adequately, it is unlikely to achieve accurate solutions at later times ($t_2 > t_1$). To prioritize learning the solution near $t = 0$, we assign greater weight to BCE of initial condition equation and MSE of NS PDE eqaution to emphasize that the u-neural network and p-neural network satisfy the initial condition. To be specific, we can define the loss function in (11) as $W_{\text{Initial Condition}}$ is constant which is big enough. We will set this constant as 1000.

2.4 Adaptive Strategies in Time for Enhanced Accuracy of Physics Informed Neural Networks

While adaptive sampling of collocation points addresses the challenge posed by the moving interface in phase field solutions, certain phase field problems with sharp transitions may still present convergence issues even with this adaptive approach. Therefore, additional strategies ave required to ensure convergence.

In this section, we introduce two adaptive strategies in time to enhance the convergence of the Physics - Informed Neural Network (PINN). The first approach, termed time-adaptive approach I, resembles the adaptive method introduced earlier, where collocation points in time are strategically selected to enhance learning. The second approach, time - adaptive approach II, adopts a different tactic by dividing the domain of interest into smaller simulation parts.

2.4.1 Adaptive Sampling of Collocation Points

In this approach, at each time step, data points including initial, boundary, and general collocation points, both original and resampled) are required to fall within a specified time interval. For example, when approximating the solution in the time domain $[0, 1]$, we initiate with small time intervals such as $[0, t_1]$, where t_1 is close to zero. Subsequently, we progressively increase the time span $[0, t_i]$, where i=1,2,..., N, ensuring that each time span is learned effectively. Once the solution is adequately learned across the entire domain, the training process advances to the next time step. Adaptive space sampling is employed within each restricted time interval to enhance learning, with collocation points sampled adaptively.

2.4.2 Adaptive Strategies for combining new network on each separated domain

In this alternative approach, we partition the domain of interest into smaller segments. Unlike the first approach, where a single network adapts collocation points in time, here we create separate networks for each time step. For instance, if the domain of interest is [0, 1] with a time step $\Delta t = 0.1$, one network is trained to learn the solution within the interval $[0, 0.1]$. Once proficient in this time interval, another network is trained for the interval $[0.1, 0.2)$, and so forth until the entire time domain is covered.

A notable consideration is that the initial condition cannot be used for later networks, as it is only valid for $t = 0$. Instead, the prediction from the previous. time steps network serves as the initial condition for the current network. This process continues until the entire time domain of the original problem is addressed. The individual networks can be integrated to obtain a solution at any point within the domain of the original problem.
These adaptive strategies, encompassing both time and space, enhance the accuracy of the improved PINNS by avoiding certain local minimum or saddle points. encountered in the baseline PINN.

3 Challenging Problem; Gradient Blowing-up on Density

The simulation of two-phase fluid systems using the Navier-Stokes-Cahn-Hilliard (NSCH) equations presents significant challenges, particularly when dealing with moving boundaries. One prominent example is the 2-dimensional rising-bubble system in two fluids. In this PDE system simulation, the density discontinuity at the moving boundary can cause **blow-up gradients problem** in the neural network forward pass, complicating the training process of Physics-Informed Neural Networks (PINNs). This writing explores the difficulties encountered in such simulations and proposes a method

to overcome these challenges using ‘Normalize initial input variable‘, ‘Normalization on a layer‘, ‘Adjust domain range for the loss computation‘,‘gradient clipping‘ and ‘adaptive loss weighting‘.

The primary difficulty in simulating NSCH systems with moving boundaries arises from the density discontinuity at the interface of two fluids. On the 2-dimensional domain, density is our goal to expect as simulation result as a variable in neural network forward pass. The discontinuity of this variable leads to gradients that can blow up during the neural network forward pass since two different values in small region brings enormous increasing gradient theoretically. Specifically, the density variable, ρ, exhibits sharp changes at the moving boundary of bubble border line becuase the rising-bubble is located between two different fluids with different density for each.
This rapid change of density causes the gradient of the density variable to become excessively large. When the loss function includes terms dependent on these gradients, the resulting gradients during backpropagation can become unmanageable, leading to errors in loss computation and training instability.

During the backward pass, the gradients of the neural network parameters are computed using automatic differentiation. This process involves constructing a computational graph during the forward pass, which records the operations performed on the inputs. When the ‘.backward()‘ command is called, the chain rule is applied to this graph to compute the gradients of the loss with respect to each variable.
When gradients blow up due to the density discontinuity, the loss term can become infinite or undefined, causing errors in the backward pass. This issue makes it impossible to compute meaningful gradients, halting the training process. Consequently, the PINN model fails to complete on learning an accurate solution to the NSCH equations.

To address the blow-up gradient problem, we implement gradient clipping between the loss computation and the backward operation. Gradient clipping involves setting a threshold beyond which gradients are scaled down to a manageable size. This technique ensures that no gradient exceeds a predefined maximum value, preventing the loss term from becoming infinite.

```
# Example code for gradient clipping in PyTorch
import torch

# Assuming loss is computed
loss = compute_loss()

# Perform gradient clipping
torch.nn.utils.clip_grad_norm_(model.parameters(), max_norm=1.0)

# Backward pass
loss.backward()

```

By bounding the gradient of the density variable in the forward pass, we can adaptively adjust the coefficients for loss terms on different regions in the domain. Specifically, we assign different weights to the loss terms for the interior of each fluid and the moving boundary interface.

- Interior of Each Fluid : In these regions, the collocation points should primarily focus on minimizing the Navier-Stokes PDE (NSpde) loss term. To mitigate the effect of large density gradients on the boundary of each fluid, we use an adaptive coefficient:

$$\text{coefficient}_{\text{NSpde}} = \frac{1}{1 + 0.1 \cdot |\nabla \rho|}$$

This coefficient reduces the influence of high gradients, stabilizing the loss term.

- Moving Boundary Interface : At the interface of moving boundary of bubble, the primary concern is the surface tension force and the phase field's accuracy. We assign a different adaptive coefficient to these terms:

$$\text{coefficient}_{\text{boundary}} = \frac{\text{difference of density in small region centered at each training point}}{0.00001 + |\nabla \rho|}$$

This coefficient emphasizes the importance of accurately recognizing only the interface(Moving boundary of rising bubble) dynamics while preventing blow-up gradients. This helps us to differentiate the boundary training points and non-boundary training points through 2-dimensional domain for training collocation points since difference of density of the non-boundary training collocation points is zero and the one of the boundary training collocation points is non-zero. After this recognizing only the collocation points on boundary collocation points by giving nonzero coefficient as weight of loss term, the denominator $0.00001 + \nabla\rho$ is trainined to be small as a part of the process for decreasing this loss term while neural network training.

To implement this strategy, we modify the loss function to include these adaptive coefficients. The loss function for the PINN model is thus a weighted sum of the NSpde loss term and the surface tension force loss term, with the weights dynamically adjusted based on the gradient of the density variable.

```
# Example code for adaptive loss weighting
def compute_adaptive_loss(u, v, rho, surface_tension_error):
    grad_rho = torch.autograd.grad(rho.sum(), inputs, create_graph=True)[0]
    coeff_NSpde = 1 / (1 + 0.1 * grad_rho)
    coeff_boundary = (rho_diff_small_region) / (0.00001 + grad_rho)

    loss_NSpde = compute_NSpde_loss(u, v, rho) * coeff_NSpde
    loss_boundary = compute_boundary_loss(surface_tension_error) * coeff_boundary

    total_loss = loss_NSpde + loss_boundary
    return total_loss

```

By employing gradient clipping and adaptive loss weighting, we can effectively manage the challenges posed by density discontinuities in NSCH systems with moving boundaries. This approach allows the PINN model to maintain stable training, accurately capture the dynamics of two-phase fluids, and reduce loss errors. These techniques, grounded in the principles of automatic differentiation and gradient management, offer a robust solution for simulating complex fluid systems with moving boundaries.

4 Result Graphs in Progress

In the next page, we show the bubble shape from the density variable of PINN simulation results. We now have the smallest error bound within the range from e^{-3} to e^{-4} for each loss term in (11). Our ultimate goal is to decrease this loss term as scale from e^{-4} to e^{-6}.

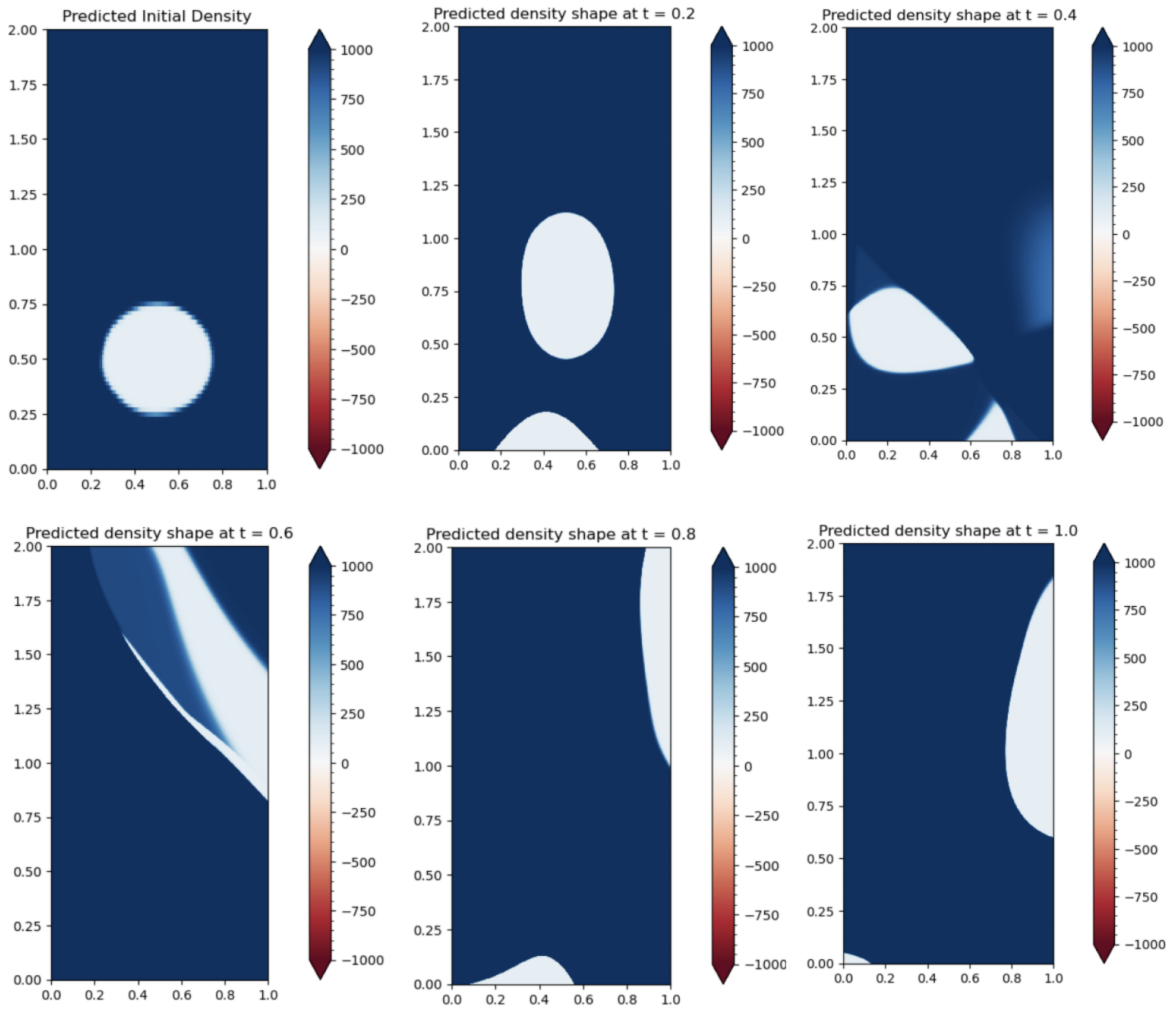

Figure 2: Predicted bubble shape at t = 0, .2, .4, .6, .8, 1.0 in diffuse interface system.

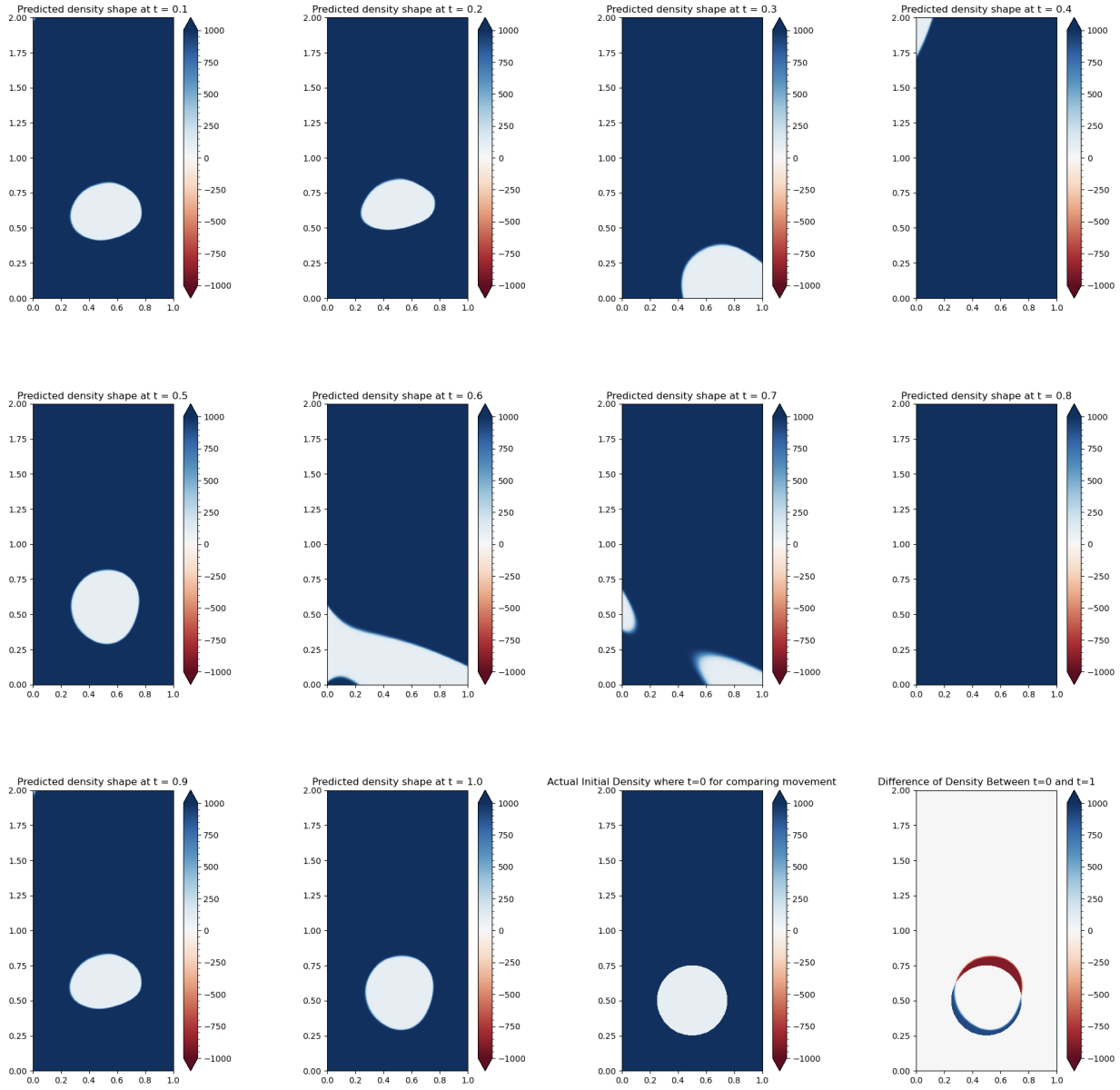

Figure 3: Predicted bubble shape at t = .1, .2, .3, .4, .5, .6, .7, .8, .9, 1.0 and the comparison of the shape t=0 and t=1 in sharp interface system.

References

[1] ABELS, H., GARCKE, H., GRÜN, G. Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities. *Math. Mod. Meth. Appl. Sci. 22*, 1150013, 2012.

[2] ARRHENIUS, S. Über die innere reibung verdünnter wässeriger lösungen. *Z. Phys. Chem.* 1U (1), 285–298, 1887.

[3] T. Demont, S. Stoter, E. van Brummelen, Numerical investigation of the sharp-interface limit of the Navier-Stokes-Cahn-Hilliard equations., *J. Fluid Mech.*, 970 (2023)

[4] D. ENRIGHT, R. FEDKIW, J. FERZIGER and I. MITCHELL, A hybrid particle level set method for improved interface capturing, *Journal of Computational Physics*, 183 (2002), pp. 83–116.

[5] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, *Proposal for Quantitative Benchmark Computations of Bubble Dynamics*, 2007.

[6] Clift R, Grace JR, Weber ME. Bubbles, *Drops and Particles*. Academic Press, 1978.

[7] M. SUSSMAN, P. SMEREKA, and S. OSHER. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114(1):146–159, 1994.