International Journal of Numerical Methods and Applications

https://pphmjopenaccess.com
https://doi.org/10.17654/0975045226003
Volume 26, Number 1, June 2026, Pages 55-79 **P-ISSN: 0975-0452**

CONSTRAINED NEURAL NETWORKS APPROACH OF THE CAHN-HILLIARD EQUATION WITH *u*-DEPENDENT MOBILITY

Abdou W. Bello[1], Jamal Adetola[2], Said A. Abdillah[3] and Charbel Z. J. Mamlankou[2]

[1]Université d'Abomey-Calavi (UAC)
Faculté des Sciences et Techniques (FAST)
Benin
e-mail: wahidi.bello@fast.uac.bj

[2]Université Nationale des Sciences, Technologies, Ingénierie et Mathématiques (UNSTIM)
École Nationale Supérieure de Génie Mathématique et Modélisation (ENSGMM)
Benin
e-mail: adetolajamal@unstim.bj
charbelzeusmamlankou@gmail.com

Received: August 3, 2025; Revised: August 9, 2025; Accepted: September 19, 2025
2020 Mathematics Subject Classification: 65J15, 65N15, 65K05, 68U20.
Keywords and phrases Cahn-Hilliard, phase separation, concentration-dependent mobility, c-PINNs, finite element method, neural networks, physical constraints.
Communicated by D. A. M. Abo Kahla

How to cite this article: Abdou W. Bello, Jamal Adetola, Said A. Abdillah and Charbel Z. J. Mamlankou, Constrained neural networks approach of the Cahn-Hilliard equation with *u*-dependent mobility, International Journal of Numerical Methods and Applications 26(1) (2026), 55-79. https://doi.org/10.17654/0975045226003

Published Online: October 29, 2025

[3]Université des Comores
Laboratoire de Mathématiques, Statistique, Informatique
et Application (LMSIA)
Benin
e-mail: intissoir2002@hotmail.fr

Abstract

This paper presents a comparative study of numerical methods for the Cahn-Hilliard equation with concentration dependent mobility, contrasting a classical finite element approach with a modern deep learning framework. We introduce a constrained Physics-Informed Neural Network (c-PINNs), which directly incorporates the physical bounds of the concentration field into the learning process. Our results demonstrate that this approach not only captures the complex dynamics of phase separation with high visual fidelity but also achieves strong quantitative agreement with a high fidelity finite element reference solution. The c-PINNs method adeptly handles the high order spatial derivatives inherent to the problem via automatic differentiation, bypassing the complexities of traditional discretization. While the initial training phase is computationally intensive, the resulting model acts as a fast and accurate surrogate, capable of instantaneous predictions. This work highlights the potential of constrained neural networks as a robust and flexible alternative for simulating complex physical phenomena, paving the way for more efficient exploration of phase separation dynamics in materials science and beyond.

1. Introduction

The Cahn-Hilliard equation is a fourth-order parabolic partial differential equation that models phase separation in binary mixtures, as discussed by Grant [19]. Initially developed for materials science, it has since been extended to diverse fields, including spinodal decomposition, image processing, and tumor growth simulation, as highlighted by Kim et al. [22]. The equation can be derived from physical principles or formulated as a gradient flow of a free energy functional, as noted by Cowan [12]. Recent

research has explored its well-posedness and long-term behavior, particularly in the presence of dynamic boundary conditions, as studied by Wu [32]. The versatility of the Cahn-Hilliard equation lies in its ability to describe complex phase separation processes, making it a valuable tool in both theoretical and applied contexts. Its mathematical analysis involves functional analysis and partial differential equation (PDE) theory [12], while computational simulations provide insights into the fundamental mechanisms underlying different modeling approaches [22].

One of the most prominent applications of the Cahn-Hilliard model is in image processing, particularly in medical imaging, inpainting, and denoising. Lupu et al. [23] demonstrated that a modified Cahn-Hilliard filter enhances the classification accuracy of damaged images processed by neural networks, improving prediction reliability in medical scans. The equation's ability to model interface dynamics makes it highly effective for image inpainting, where it reconstructs missing or corrupted regions while preserving structural coherence. Zhang et al. [34] applied Cahn-Hilliard dynamics to binary image restoration, achieving superior visual quality, while Chen et al. [11] proposed a generalized approach to adapt the model to various types of image degradation. Moreover, in denoising applications, the Cahn-Hilliard model treats noise as perturbations in the phase field, allowing it to smooth out unwanted artifacts while maintaining essential image details an approach particularly beneficial in medical imaging, as emphasized by Miranville et al. [24].

Beyond image processing, the Cahn-Hilliard equation has also been employed in biological modeling, particularly in tumor growth simulation. Coupled with reaction-diffusion equations, it has been used to study tumor progression and nutrient dynamics, providing valuable insights into cancer development and treatment strategies [25]. The study of complex physical systems, often involving quantum phenomena or interactions with electromagnetic fields, has seen significant theoretical and applied interest, as evidenced by works exploring quantum coherence and inversion in semiconductor quantum dots and three-level atoms [1-5, 27]. These diverse

applications underscore the broad relevance of sophisticated modeling and numerical techniques across scientific disciplines.

From a mathematical perspective, the study of concentration-dependent mobility in the Cahn-Hilliard equation has been a subject of considerable interest. While early studies often assumed a constant mobility function, Cahn and Hilliard demonstrated that mobility depends on concentration, revealing enhanced diffusion in interfacial regions compared to pure phases a phenomenon later confirmed experimentally [10, 21]. A commonly used mobility function is $1 - u^2$, as proposed in [6]. The existence of solutions has been rigorously established in various settings: Yin proved existence in one dimension for degenerate mobility [33], Elliott extended these results to higher dimensions [14], and Grün provided additional theoretical insights [20]. However, uniqueness for the degenerate mobility case remains an open problem.

In recent years, the advent of scientific machine learning has offered a new paradigm for solving such complex PDEs. Physics-Informed Neural Networks (PINNs) have emerged as a powerful, mesh-free alternative, leveraging automatic differentiation to embed physical laws directly into the training process [30]. This methodology has shown promise across various domains, including phase field modeling [18], and has been extended to efficient frameworks for both forward and inverse problems [26]. Other innovative approaches combine physics-informed principles with geometric adaptability, such as PhyGeoNet for solving parametric PDEs on irregular domains [17]. This synergy between classical numerical analysis and modern deep learning techniques offers robust tools for addressing challenging problems in various fields, from materials science to quantum optics, where accurate modeling of dynamics and coherence is crucial [29]. However, the application of PINNs to fourth order equations with degenerate coefficients and hard physical constraints, such as the Cahn-Hilliard equation studied here, remains an active and challenging area of research.

The aim of this work is to develop and compare two efficient numerical methods for solving the Cahn-Hilliard equation with concentration

dependent mobility: a finite element approximation using a Crouzeix-Raviart formulation and our proposed constrained Physics-Informed Neural Network. Foundational contributions in this area include the works of Elliott and Cahn [14] and Novick-Cohen [28], whose studies provide essential theoretical and computational frameworks. This paper is organized as follows. In Section 2, we present the mathematical formulation of the problem. Section 3 is devoted to the finite element discretization. Section 4 details our neural network approach. In Section 5, we perform numerical experiments with both methods. In Section 6, we discuss the results of our study, and conclude in Section 7.

2. Mathematical Formulation

In this section, we present the mathematical model used to describe the phase separation process, namely, the Cahn-Hilliard equation. We first introduce the continuous problem and its main components, which will serve as the basis for the subsequent numerical discretization.

2.1. Continuous problem

Let Ω be a bounded domain in $\mathbb{R}^d$ with $d = 2$, and let $\partial\Omega$ denote its Lipschitz boundary. We consider the initial boundary value problem for the Cahn-Hilliard equation, which seeks to find the order parameter $u(\mathrm{x}, t)$:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot (M(u)\nabla\mu), & \text{if } \Omega_T \coloneqq \Omega \times (0, T), \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \forall \mathbf{x} \in \Omega, \\ \dfrac{\partial u}{\partial \mathbf{n}} = \dfrac{\partial \mu}{\partial \mathbf{n}} = 0, & \text{on } \partial\Omega \times (0, T), \end{cases} \tag{1}$$

where $\mathbf{n}$ is the unit outward normal vector to $\partial\Omega$ and $\gamma > 0$ is a positive constant related to the interface energy. In this system, μ represents the chemical potential, defined as:

$$\mu \coloneqq -\gamma\Delta u + f'(u). \tag{2}$$

The term $f(u)$ is the homogeneous free energy density, a double-well potential that energetically favors phase separation. A standard choice, which we adopt, is the quartic potential [16]:

$$f(u) = \frac{1}{4}(u^2 - 1)^2. \tag{3}$$

Its derivative is thus $f'(u) = u^3 - u$. The function u is an order parameter, physically constrained to the interval $[-1, 1]$, where $u = \pm 1$ denote the pure material phases. The homogeneous Neumann boundary conditions in (1) ensure mass conservation and represent contact-angle-free conditions at the boundary. This physical constraint on u is mathematically enforced by using a degenerate diffusional mobility $M(u)$. In this paper, we use:

$$M(u) = 1 - u^2. \tag{4}$$

This mobility satisfies $M(-1) = M(1) = 0$ and $M(k) > 0$ for all $k \in (-1, 1)$. The degeneracy at $u = \pm 1$ ensures that if $|u_0(\mathbf{x})| \leq 1$, then the solution remains bounded within $[-1, 1]$ for all time, as the diffusion process halts at the pure phase values.

We do not prove existence and uniqueness for system (1), as these have been established by Barrett et al. [7] for the Cahn-Hilliard equation with degenerate mobility.

2.2. Weak formulation

To apply the finite element method, we first derive the weak (variational) formulation. The standard approach is to split the fourth-order PDE for u into a system of two second-order PDEs by treating μ as an independent variable [15]:

$$\frac{\partial u}{\partial t} - \nabla \cdot (M(u)\nabla\mu) = 0 \text{ in } \Omega, \tag{5}$$

$$\mu + \gamma\Delta u - f'(u) = 0 \text{ in } \Omega. \tag{6}$$

The weak formulation is obtained by multiplying each equation by a suitable test function from the Sobolev space $H^1(\Omega)$ and integrating over Ω. Using Green's first identity and applying the zero-flux boundary conditions $\frac{\partial \mu}{\partial \mathbf{n}} = 0$ and $\frac{\partial u}{\partial \mathbf{n}} = 0$ to eliminate the boundary integrals, we arrive at the following mixed variational problem:

For each time $t \in (0, T]$, find $(u(t), \mu(t)) \in H^1(\Omega) \times H^1(\Omega)$ such that

$$\begin{cases} \int_\Omega \frac{\partial u}{\partial t} v \, d\mathbf{x} + \int_\Omega M(u) \nabla \mu \cdot \nabla v d\mathbf{x} = 0, & \forall v \in H^1(\Omega), \\ \int_\Omega \mu w \, d\mathbf{x} + \gamma \int_\Omega \nabla u \cdot \nabla w d\mathbf{x} - \int_\Omega f'(u) w d\mathbf{x} = 0, & \forall w \in H^1(\Omega). \end{cases} \tag{7}$$

3. Finite Element Discretization

In this section, we describe the numerical scheme used to approximate the solution of the Cahn-Hilliard problem defined by the weak formulation (7). We first introduce the spatial discretization based on the finite element method and then the temporal discretization.

Let $\{\mathcal{T}_h\}_{h>0}$ be a family of quasi-uniform partitions of the square domain Ω into nonoverlapping open simplices (triangles) κ, where $h := \max_{\kappa \in \mathcal{T}_h} \text{diam}(\kappa)$. The nonlinearity of the system, particularly in $M(u)$ and $f'(u)$, requires an iterative solver at each time step. Our numerical scheme is designed to solve the nonlinear variational problem for the pair (u_h, μ_h).

3.1. Space and time discretization

3.1.1. Spatial discretization with Crouzeix-Raviart elements

We approximate the solution (u, μ) in a discrete space $V_h \subset L^2(\Omega)$. For this study, we employ the non-conforming Crouzeix-Raviart (CR) finite element space [13]. The space V_h is defined as:

$$V_h := \{v_h \in L^2(\Omega) | \, v_h|_\kappa \in \mathcal{P}_1(\kappa), \, \forall \kappa \in \mathcal{T}_h, \, v_h \text{ is continuous at edge midpoints}\}.$$

The choice of non-conforming CR elements is motivated by several factors. They offer a simpler structure with fewer degrees of freedom compared to conforming linear elements, which can reduce computational cost. Moreover, they have proven to be stable and effective for fourth-order equations like Cahn-Hilliard when split into a system, as they relax the strict continuity requirements of conforming elements [9].

The semi-discrete problem is to find $(u_h(t), \mu_h(t)) \in V_h \times V_h$ such that for all $(\phi_h, \psi_h) \in V_h \times V_h$:

$$\begin{cases} \left(\dfrac{\partial u_h}{\partial t}, \phi_h \right) + (\nabla_h (M(u_h) \nabla_h \mu_h), \nabla_h \phi_h) = 0, \\ (\mu_h, \psi_h) + \gamma (\nabla_h \mu_h, \nabla_h \psi_h) - (f'(u_h), \psi_h) = 0. \end{cases} \tag{8}$$

Here, $(\cdot, \cdot)$ denotes the $L^2(\Omega)$ inner product, defined as $(a, b) := \int_\Omega ab \, d\mathbf{x}$ and ∇_h is the element-wise gradient.

3.1.2. Fully-discrete scheme

To obtain a fully-discrete scheme, we apply a finite difference approximation in time to the semi-discrete weak formulation (8). A common and effective strategy is to use a semi-implicit scheme, which treats stiff linear terms implicitly and nonlinear terms explicitly [31]. This method yields a linear system of equations to be solved at each time step.

Let $\Delta t > 0$ be the time step size. Applying a first-order, semi-implicit Euler scheme to (8) leads to the following fully-discrete problem: Given $u_h^n \in V_h$, find $(u_h^{n+1}, \mu_h^{n+1}) \in V_h \times V_h$ such that for all $(\phi_h, \psi_h) \in V_h \times V_h$:

$$\begin{cases} \left(\dfrac{u_h^{n+1} - u_h^n}{\Delta t}, \phi_h \right) + (M(u_h^n) \nabla_h \mu_h^{n+1}, \nabla_h \phi_h) = 0, \\ (\mu_h^{n+1}, \psi_h) + \gamma (\nabla_h u_h^{n+1}, \nabla_h \psi_h) = (f'(u_h^n), \psi_h). \end{cases} \tag{9}$$

This semi-implicit treatment is computationally efficient but is only conditionally stable, typically requiring the time step Δt to be sufficiently small with respect to the mesh size h. The analysis of such schemes is well-documented in the literature [7].

3.2. Solution of the nonlinear system

For improved stability, a fully implicit scheme is often preferred over the semi-implicit version. This requires solving a nonlinear system at each time step:

$$\begin{cases} \left(\dfrac{u_h^{n+1} - u_h^n}{\Delta t}, \phi_h \right) + \left(M(u_h^{n+1}) \nabla_h \mu_h^{n+1}, \nabla_h \phi_h \right) = 0, \\ (\mu_h^{n+1}, \psi_h) + \gamma (\nabla_h u_h^{n+1}, \nabla_h \psi_h) - (f'(u_h^{n+1}), \psi_h) = 0. \end{cases} \tag{10}$$

We apply a fixed-point (Picard) iteration to solve the nonlinear system (10). This method consists of linearizing the problem by evaluating the nonlinear terms at the previous iteration's solution. At each time step n, we seek the solution u_h^{n+1} by generating a sequence $\{u_h^{(k)}\}_{k \geq 0}$ that converges to it. We start with an initial guess, $u_h^{(0)} = u_h^n$. For each iteration $k \geq 0$, we solve the following linear variational problem for the next iterate $(u_h^{(k+1)}, \mu_h^{(k+1)})$:

$$\begin{cases} \left(\dfrac{u_h^{(k+1)} - u_h^n}{\Delta t}, \phi_h \right) + \left(M(u_h^{(k)}) \nabla_h \mu_h^{(k+1)}, \nabla_h \phi_h \right) = 0, \\ (\mu_h^{(k+1)}, \psi_h) + \gamma (\nabla_h u_h^{(k+1)}, \nabla_h \psi_h) = (f'(u_h^{(k)}), \psi_h), \end{cases} \tag{11}$$

for all test functions $(\phi_h, \psi_h) \in V_h \times V_h$. At each step k, we solve a linear system of algebraic equations. The iteration proceeds until a stopping criterion, such as $\dfrac{\| u_h^{(k+1)} - u_h^{(k)} \|_{L^2}}{\| u_h^{(k+1)} \|_{L^2}} < \text{tol}$, is met. The procedure is detailed in Algorithm 1.

Algorithm 1. Fixed-point iterative scheme for the Cahn-Hilliard FEM discretization

1: **Input**: Initial condition u_h^0, parameters Δt, γ, tolerance tol, max iterations $K_{\max}$.

2: **Initialization**: Set $u_h^n \leftarrow u_h^0$.

3: **for** each time step $n = 0, 1, 2, \ldots$ **do**

4: Set initial guess for the fixed-point iteration: $u_h^{(k)} \leftarrow u_h^n$ (for $k = 0$).

5: **for** $k = 0, 1, \ldots, K_{\max} - 1$ **do**

6: Assemble and solve the linear system from (11) for $(u_h^{(k+1)}, u_h^{(k+1)})$,

7: **if** $\| u_h^{(k+1)} - u_h^{(k)} \|_{L^2} < \text{tol} \cdot \| u_h^{(k+1)} \|_{L^2}$ **then**

8: Set $u_h^{n+1} \leftarrow u_h^{(k+1)}$ and break inner loop.

9: **end if**

10: Update for next iteration: $u_h^{(k)} \leftarrow u_h^{(k+1)}$.

11: **end for**

12: **end for**

13: **Output**: Solution sequence $\{u_h^n\}$.

4. Neural Network Approximation with Constraints

In this section, we transition to the primary focus of our study: the application of neural networks to approximate the solution of the Cahn-Hilliard equation. We employ a variant of Physics-Informed Neural Networks, which we term constrained-PINNs (c-PINNs), to enforce the physical bounds of the solution.

4.1. Constrained-PINNs formulation

PINNs [30] approximate the solution $u(\mathrm{x}, t)$ with a neural network $u_\theta(\mathbf{x}, t)$, where θ represents the trainable parameters. The core of the c-PINNs methodology is to train the network by minimizing a loss function that encodes the problem's physics. The first component of this loss is built

upon the residual of the Cahn-Hilliard equation:

$$\mathcal{R}(\mathbf{x}, t; \theta) := \frac{\partial u_\theta}{\partial t} - \nabla \cdot (M(u_\theta)\nabla \mu_\theta), \tag{12}$$

where the chemical potential is $\mu_\theta := -\gamma \Delta u_\theta + f'(u_\theta)$. A key feature of c-PINNs is the use of automatic differentiation, a cornerstone of modern deep learning frameworks [8]. All partial derivatives required to compute the residual are calculated analytically with respect to the network inputs $(\mathbf{x}, t)$, avoiding discretization errors. The total loss function is a weighted sum of the different physical constraints:

$$\mathcal{L}(\theta) = \lambda_{pde}\, \mathcal{L}_{pde}(\theta) + \lambda_{ic}\, \mathcal{L}_{ic}(\theta) + \lambda_{bc}\, \mathcal{L}_{bc}(\theta). \tag{13}$$

Specifically, these loss terms are defined as mean squared errors over sets of collocation points. The PDE loss, $\mathcal{L}_{pde}$, enforces the governing equation on N_{pde} points in the domain interior:

$$\mathcal{L}_{pde}(\theta) = \frac{1}{N_{pde}} \sum_{i=1}^{N_{pde}} | \mathcal{R}(\mathbf{x}_i^{pde}, t_i^{pde}; \theta) |^2. \tag{14}$$

The initial condition loss, $\mathcal{L}_{ic}$, penalizes deviations from $u_0(\mathbf{x})$ on N_{ic} points at $t = 0$:

$$\mathcal{L}_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} | u_\theta(\mathbf{x}_i^{ic}, 0) - u_0(\mathbf{x}_i^{ic}) |^2. \tag{15}$$

Finally, the boundary condition loss, $\mathcal{L}_{bc}$, enforces the zero-flux conditions on N_{bc} points on $\partial\Omega$:

$$\mathcal{L}_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \left(\left| \frac{\partial u_\theta}{\partial \mathbf{n}} \right|^2 + \left| \frac{\partial \mu_\theta}{\partial \mathbf{n}} \right|^2 \right)_{(\mathbf{x}_i^{bc}, t_i^{bc})}. \tag{16}$$

4.2. Enforcing physical bounds via clipping

A key limitation of standard PINNs is that they cannot ensure their output u_θ to remain within the physically meaningful range $|u| \leq 1$. In our

c-PINNs approach, we enforce this constraint by applying a clipping function to the raw output of the neural network, denoted $\tilde{u}_\theta$, before it is used to compute the losses. We define the *constrained solution* u_θ as:

$$u_\theta(\mathbf{x}, t) = \text{clip}(\tilde{u}_\theta(\mathbf{x}, t), -1, 1) = \begin{cases} -1, & \text{if } \tilde{u}_\theta(\mathbf{x}, t) < -1, \\ \tilde{u}_\theta(\mathbf{x}, t), & \text{if } -1 \le \tilde{u}_\theta(\mathbf{x}, t) \le 1, \\ 1, & \text{if } \tilde{u}_\theta(\mathbf{x}, t) > 1. \end{cases} \tag{17}$$

This hard projection ensures the order parameter remains within $[-1, 1]$. We acknowledge that alternative methods exist, but the clipping approach is conceptually simple and serves as the primary contribution we explore. With u_θ clipped, the mobility $M(u_\theta) = 1 - u_\theta^2$ is automatically within its physical range $[0, 1]$, making an additional clipping on $M(u_\theta)$ unnecessary. The complete procedure is outlined in Algorithm 2.

Algorithm 2. c-PINNs algorithm for the Cahn-Hilliard equation

1: **Define**:
2: A neural network $\tilde{u}_\theta(\mathbf{x}, t)$ with parameters θ.
3: The constrained output $u_\theta = \text{clip}(\tilde{u}_\theta, -1, 1)$ as per (17).
4: The loss function $\mathcal{L}(\theta)$ as defined in (13).
5: **Initialization**:
6: Initialize network parameters θ.
7: Generate collocation points for the PDE, initial, and boundary conditions.
8: **Training loop**:
9: **for** a specified number of optimization epochs **do**
10: Compute the total loss $\mathcal{L}(\theta)$.
11: Compute the gradient: $\nabla_\theta \mathcal{L}(\theta)$.
12: Update the parameters θ using an optimizer (Adam, L-BFGS).
13: **end for**
14: **Output**: The trained parameters θ^*, defining the solution approximant u_{θ^*}.

5. Numerical Experiments

5.1. Experimental setup and training details

We consider a phase separation scenario within the domain $\Omega = [-0.5, 0.5] \times [0, 1]$, with $\gamma = \sqrt{3}$ and a time interval of $T = 5$ seconds. The initial condition, $u_0(\mathbf{x})$, is a spatial random field where values are drawn from $\mathcal{U}[-1, 1]$. To ensure reproducibility, the field is generated once using a pseudo-random number generator initialized with a fixed seed of 1234. A high-fidelity FEM solution computed on a 128×128 grid serves as the reference. A tolerance of 10^{-10} has been set as the stopping criterion for the fixed-point method.

All simulations were performed on a 2017 MacBook Pro with a 3.1 GHz Intel Core i7 processor, 16 GB of RAM, and an NVIDIA GeForce GT 650M graphics card. The c-PINNs architecture consists of 3 hidden layers with 64 neurons each, using the tanh activation function. For training, we sample $N_{pde} = 10{,}000$ points using Latin Hypercube Sampling, and $N_{ic} = N_{bc} = 2{,}500$ points for the initial and boundary conditions. Training involves 10,000 epochs with the Adam optimizer (learning rate 10^{-3}), followed by L-BFGS for fine-tuning.

5.2. Results and analysis

Figure 1 presents a side-by-side visual comparison of the phase separation dynamics as captured by the reference FEM solution and the c-PINNs prediction. A strong visual correspondence is immediately apparent at every stage of the evolution. In the early stages $(t = 0.15, 0.25)$, the c-PINNs model accurately captures the intricate, fine-grained structures that emerge from the random initial state, a challenging task due to the high spatial frequencies involved. As the system evolves towards equilibrium, the c-PINNs prediction correctly reproduces the coarsening process, where larger, energetically favorable domains grow at the expense of smaller ones. Notably, the morphology, location, and even the subtle curvature of the

interfaces between the yellow $(u \approx 1)$ and black $(u \approx -1)$ phases are remarkably well-matched with the high-fidelity FEM reference. This qualitative agreement, maintained throughout the entire simulation, strongly suggests that the c-PINNs approach has successfully learned the underlying physical laws and complex nonlinearities of the Cahn-Hilliard equation.

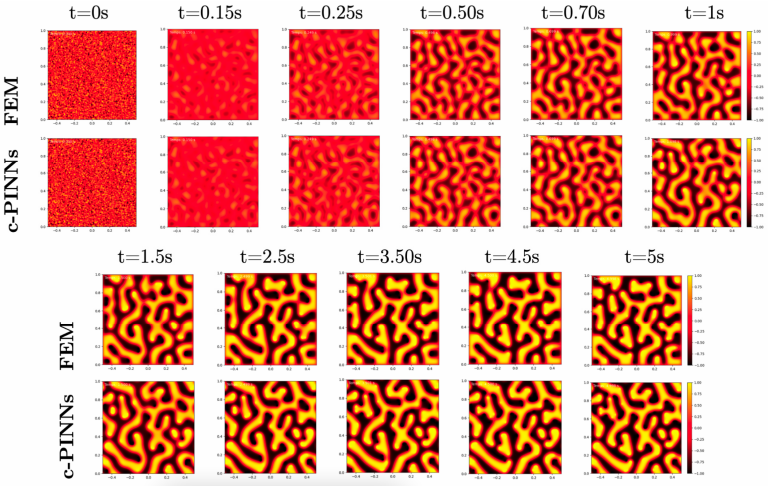

Figure 1. Visual comparison between the reference solution (FEM, top rows) and the c-PINNs prediction (bottom rows) at selected times $t \in \{0, 0.15, 0.25, 0.5, 0.7, 1, 1.5, 2, 3.5, 4.5, 5\}$.

The success of the c-PINNs is contingent upon the effective minimization of the loss function. Figure 2 illustrates the evolution of the total loss during our two-phase training strategy. The initial phase, using the Adam optimizer for a total of 10,000 epochs, shows a rapid and significant decrease in the loss. For visual clarity, the plot on the left does not show every epoch; instead, it records the loss value at intervals of 400 epochs, resulting in 25 data points on the x-axis which represent the progression of the training. This initial phase effectively navigates the complex optimization landscape, reaching a loss value on the order of 10^{-3}. Following this, the second phase employs the L-BFGS optimizer, which achieves further fine-tuning and reduces the loss by an additional order of

magnitude to a final plateau around 10^{-4}. This two-stage process demonstrates a stable and successful training convergence, indicating that the network parameters have settled in a configuration that satisfies the physical constraints of the problem to a high degree of accuracy.

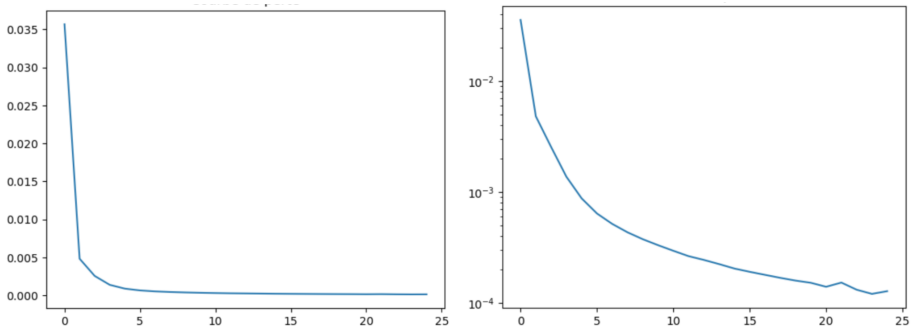

Figure 2. Evolution of the total loss during training. Left: Initial descent with Adam. Right: Fine-tuned convergence with L-BFGS.

For a quantitative validation of our c-PINNs approach, we perform a detailed error analysis against the high-fidelity FEM reference solution. In addition to the absolute error norms, $E_{L^2}(t)$ and $E_{L^\infty}(t)$ defined previously, we also compute the relative L^2 error at each time step to normalize the discrepancy by the magnitude of the solution itself:

$$E_{L^2}^{\text{rel}}(t) = \frac{\| u_h(\cdot, t) - u_\theta(\cdot, t) \|_{L^2(\Omega)}}{\| u_h(\cdot, t) \|_{L^2(\Omega)}}. \tag{18}$$

The temporal evolution of these error metrics is plotted in Figure 3. The plots show that after the initial transient phase, both the absolute and relative errors remain low and do not exhibit any significant growth over time. This temporal stability is a crucial indicator of the robustness of the c-PINNs model, confirming that it has learned a physically consistent dynamic evolution and not just a static fit to the data.

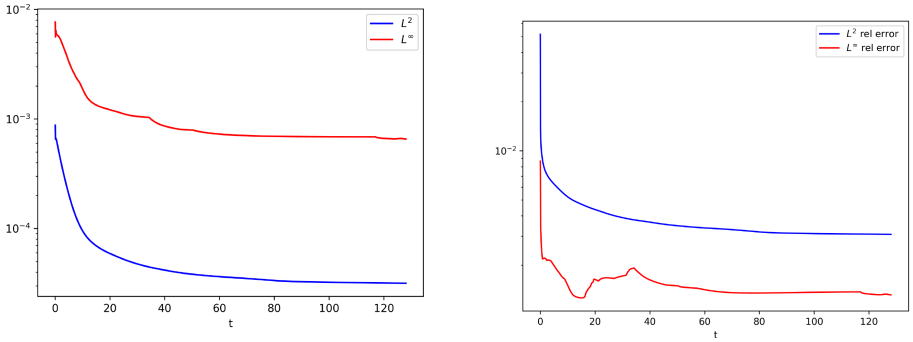

Figure 3. Temporal evolution of absolute (left) and relative (right) errors, quantified using the L^2 and L^∞ norms.

To provide a concise, aggregate measure of the model's performance over the entire simulation, we compute statistics of these time-dependent error metrics. Table 1 reports the mean, minimum, and maximum values of each error type, calculated across all discrete time instances from $t = 0$ to $t = T$. These summary statistics offer a comprehensive view of the model's accuracy, capturing its average behavior as well as its best- and worst-case performance.

The results are highly encouraging. The c-PINNs model achieves a mean absolute L^2 error of 5.78×10^{-4}, signifying a very high degree of global accuracy. The mean L^∞ error, which measures the average of the worst-case local deviations at each time step, is also low at 1.05×10^{-3}. This demonstrates that the model avoids significant localized errors. The relative errors further confirm the fidelity of the prediction, with a mean relative L^2 error of only 3.76%. These quantitative results strongly support the visual evidence, confirming that our c-PINNs approach provides a reliable and accurate approximation to the Cahn-Hilliard equation.

Table 1. Summary of absolute and relative error metrics, aggregated over the entire simulation time. The values demonstrate the high fidelity of the c-PINNs prediction against the FEM reference

Error metric	Mean	Minimum	Maximum
Absolute L^2 error	5.78×10^{-4}	3.16×10^{-4}	8.75×10^{-3}
Absolute L^∞ error	1.05×10^{-3}	6.56×10^{-4}	7.66×10^{-3}
Relative L^2 error (%)	3.76	3.08	51.7
Relative L^∞ error (%)	1.49	1.26	8.65

Finally, a comparison of the computational resources required by each method is presented in Table 2. The table reveals an interesting performance profile. The FEM solver, leveraging a lean fixed-point algorithm within a time-marching loop, completes the entire simulation in approximately 3 minutes. In contrast, the c-PINNs approach requires about 9 minutes for its comprehensive training phase.

However, a direct comparison of these times can be misleading, as the two methods operate on fundamentally different paradigms. The 3-minute FEM time represents a fast, sequential process, where the solution is computed step-by-step. The 9-minute c-PINNs time is a one-off, upfront investment to learn the entire spatio-temporal solution manifold. In terms of memory, the c-PINNs approach has a larger footprint, utilizing approximately 430 MB compared to the 257 MB required by the FEM, which is attributable to storing the neural network parameters.

The true advantage of the c-PINNs becomes apparent post-training. The resulting model acts as a surrogate that can perform inference predicting the solution at any coordinate pair $(\mathbf{x}, t)$ within the training domain almost instantly. This makes the c-PINNs approach particularly attractive for applications requiring numerous model evaluations, such as inverse problems, uncertainty quantification, or design optimization, where the higher initial training cost is quickly amortized over many fast subsequent queries.

Table 2. Comparison of computational performance between the FEM and c-PINNs methods

Metric	FEM	c-PINNs
Total computation time	$\approx$ 3 minutes	$\approx$ 9 minutes (training)
Memory usage	257 MB	430 MB
Time discretization	Time-marching ($\Delta t = 1/128$ s)	Global (t as network input)
Solver / Optimizer	Fixed-point solver	Adam + L-BFGS

6. Discussion

The results presented in this study highlight the viability of constrained Physics-Informed Neural Networks as a powerful tool for simulating the Cahn-Hilliard equation. The strong qualitative agreement shown in Figure 1, combined with the low quantitative errors reported in Table 1, confirms that the c-PINNs method successfully learns the complex, nonlinear dynamics of phase separation.

A key advantage of the c-PINNs approach is its inherent ability to handle the high order derivatives of the Cahn-Hilliard equation. By leveraging automatic differentiation, the method bypasses the intricate formulations required for high order finite elements, offering a more direct implementation path from the governing PDE. Furthermore, the mesh free nature of PINNs provides flexibility, particularly for problems with complex geometries or evolving interfaces where traditional meshing can become a bottleneck. Our proposed clipping mechanism, while simple, proved to be an effective strategy for enforcing the strict physical bounds on the concentration field, a critical requirement for this problem.

The computational performance comparison presented in Table 2 reveals a fundamental trade-off that merits deeper analysis. The FEM solver, relying on a time-marching scheme, demonstrates superior speed for a single, sequential forward simulation, completing the process in approximately 3 minutes. This efficiency stems from its iterative nature, solving localized problems at each discrete time step and spatial node.

In contrast, the c-PINNs approach demands a substantial upfront investment in training time, approximately 9 minutes in our experimental setup. This training duration is influenced by several factors, including the network architecture (number of layers and neurons), the number of collocation points, and the complexity of the PDE itself. The training time for PINNs generally scales with the product of the number of collocation points and the number of network parameters. For a fixed number of samples per epoch, training time scales roughly linearly with the number of network parameters and super-linearly with the dimensionality of the input space (e.g., going from 2D to 3D significantly increases the number of required collocation points and network capacity). The iterative nature of optimization also plays a role, with convergence rates depending on the choice of optimizer and hyper-parameters.

Regarding memory efficiency, Table 2 indicates that c-PINNs have a larger footprint (430 MB) compared to FEM (257 MB). This is primarily due to storing the weights and biases of the neural network, as well as the computational graph required for automatic differentiation during training. For very large-scale problems, particularly in 3D, or deeper networks, memory consumption can become a significant constraint, potentially necessitating advanced strategies like gradient check pointing, mixed-precision training, or distributed computing frameworks.

The true and most compelling advantage of the c-PINNs becomes apparent post-training, in its inference speed. Once the network is trained, it acts as a fast, highly efficient surrogate model. Predicting the solution $u(\mathbf{x}, t)$ at any arbitrary coordinate pair $(\mathbf{x}, t)$ within the training domain is virtually instantaneous, typically on the order of milliseconds, requiring only a single forward pass through the already optimized network. This contrasts sharply with FEM, where obtaining a solution at a new spatio-temporal point (e.g., a time not explicitly simulated, or a finer spatial resolution) would necessitate re-running or interpolating from an existing, discretely sampled solution.

This inherent difference in operational paradigms makes c-PINNs particularly attractive for applications requiring numerous model evaluations. For instance, in inverse problems, where material parameters are sought by minimizing the discrepancy between model predictions and experimental data, or in uncertainty quantification studies that involve thousands of Monte Carlo simulations, the initial training cost of c-PINNs is quickly amortized over countless fast subsequent queries. Similarly, in design optimization or real-time control, where rapid predictions are paramount, the c-PINNs surrogate model offers a substantial advantage over traditional solvers. The ability to query the solution at arbitrary points also makes it highly flexible for adaptive sampling or multi-fidelity approaches.

Limitations of the current study suggest avenues for future work. While effective, the hard clipping constraint introduced in (17) provides a clear physical bound but also introduces non-differentiability at the boundaries. Future research could explore smoother architectural constraints to enforce $u \in [-1, 1]$, such as employing a scaled tanh activation function for the final layer output ($u_\theta = \tanh(\tilde{u}_\theta)$ or $u_\theta = 2 \cdot \text{sigmoid}(\tilde{u}_\theta) - 1$), or integrating barrier functions directly into the loss function. These smoother approaches might improve optimization convergence by maintaining differentiability across the entire domain, potentially offering a valuable trade-off. A more significant area for future research is to assess and enhance the scalability of the c-PINNs method to three-dimensional (3D) cases. While 2D simulations provide valuable insights, many real-world applications of phase separation occur in 3D. Extending the method to 3D introduces challenges such as the curse of dimensionality, requiring an exponentially larger number of collocation points and greater network capacity, which in turn demands more powerful computational resources and careful handling of 3D boundary conditions. Nevertheless, the mesh-free nature of PINNs becomes even more advantageous in 3D, as traditional 3D mesh generation and adaptation are notoriously complex. Therefore, investigating techniques like adaptive sampling, domain decomposition, or specialized network architectures will be crucial for making c-PINNs a practical tool for 3D phase separation, an

extension that would significantly enhance the study's impact and applicability.

7. Conclusion

In this work, we have presented a comprehensive comparison between a classical Crouzeix-Raviart finite element method and a novel constrained Physics-Informed Neural Network (c-PINNs) for solving the Cahn-Hilliard equation with degenerate mobility. We demonstrated that the c-PINNs approach, which enforces physical bounds through a direct clipping mechanism, successfully captures the intricate dynamics of phase separation, achieving results that are in excellent qualitative and quantitative agreement with our high fidelity FEM reference.

The primary contribution of this study is the successful validation of this constrained deep learning framework for a challenging, high order PDE. The results underscore the potential of c-PINNs as a flexible, mesh free alternative to traditional numerical methods, particularly due to the natural and exact computation of complex derivatives via automatic differentiation. While the computational paradigms differ, with FEM excelling at single simulations and c-PINNs offering a powerful surrogate model after an initial training cost, our findings confirm that both methods are effective tools for investigating phase separation.

Future research will focus on extending this framework to more complex scenarios. This includes exploring its application to three dimensional problems, coupling the Cahn-Hilliard dynamics with other physical models such as fluid flow, and leveraging the surrogate nature of the trained network to tackle inverse problems, such as determining material parameters from experimental data. Beyond these extensions, the robust and adaptable nature of c-PINNs opens exciting avenues for interdisciplinary applications. In materials science, this approach could accelerate materials discovery by rapidly screening new alloy compositions or microstructural designs. For biological systems, it holds promise in cancer modeling, allowing for more accurate predictions of tumor growth and response to therapies by

incorporating complex, spatio-temporal dynamics. Furthermore, in advanced imaging, c-PINNs could contribute to novel denoising or segmentation algorithms, enhancing the quality and interpretability of medical or scientific images. The synergy between classical numerical analysis and emerging machine learning techniques promises to unlock new insights into the rich behavior of phase field models and drive innovation across a multitude of scientific and engineering domains.

Acknowledgements

The authors thank the TICS MASTER Startup for its partial financial support. The authors also wish to express their gratitude to the anonymous reviewers whose insightful comments significantly contributed to improving the quality of this manuscript.

References

[1] M. Abdel-Aty, D. Abo-Kahla and A. S. F. Obada, Spatial dependence of moving three-level atoms interacting with a three-laser beam, Canadian Journal of Physics 91(12) (2013), 1068-1073.

[2] D. A. M. Abo-Kahla, The Pancharatnam phase of a three-level atom coupled to two systems of *n*-two level atoms, Journal of Quantum Information Science 1(1) (2011), 44-55.

[3] D. A. M. Abo-Kahla, The atomic inversion and the purity of a quantum dot two-level systems, Applied Mathematics and Information Sciences 10(4) (2016), 1-5.

[4] D. A. M. Abo-Kahla, Long-lived quantum coherence and nonlinear properties of a two-dimensional semiconductor quantum well, Journal of the Optical Society of America B 37(11) (2020), A96-A109.

[5] D. A. M. Abo-Kahla, Long-lived quantum coherence in a two-level semiconductor quantum dot, Pramana 94(1) (2020), 1-14.

[6] N. D. Alikakos, P. W. Bates and G. Fusco, On the Cahn-Hilliard equation with concentration dependent mobility, J. Differential Equations 90 (2017), 81-135.

[7] John Barrett, James Blowey and Harald Garcke, Finite element approximation of the Cahn-Hilliard equation with degenerate mobility, SIAM J. Numer. Anal. 37(1) (2000), 286-318.

[8] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul and Jeffrey Mark Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018), 1-43.

[9] N.-E. Bohne, Patrick Ciarlet, Jr. and S. Sauter, Crouzeix-Raviart elements on simplicial meshes in d dimensions, 2024. arXiv:2407.04361 DOI: 10.48550/arXiv.2407.04361.

[10] J. W. Cahn, Phase separation by Spinodal decomposition in alloys, Acta Metallurgica 9(9) (1961), 795-801.

[11] X. Chen, Y. Cheng, J. Xu and C. Wang, Cahn-Hilliard inpainting and a generalization for gray-value images, SIAM J. Appl. Math. 83 (2023), 123-145.

[12] Craig Cowan. The Cahn-Hilliard equation as a gradient flow, PhD thesis, Simon Fraser University, 2005. researchgate.net/publication/241623162.

[13] M. Crouzeix and P. A. Raviart, Conforming and nonconforming finite element methods for solving the stationary stokes equations, Mathematical Modelling and Numerical Analysis 7(R3) (1973), 33-75.

[14] C. M. Elliott and J. W. Cahn, The Cahn-Hilliard equation with a concentration dependent mobility: motion by minus the Laplacian of the mean curvature, European J. Appl. Math. 7(3) (1996), 287-301.

[15] Charles M. Elliott and Stig Larsson, A second order error estimate for a time-dependent Cahn-Hilliard equation, Mathematical Modelling and Numerical Analysis 23(3) (1989), 397-405.

[16] David J. Eyre, Unconditionally gradient stable time-marching the Cahn-Hilliard equation, MRS Online Proceedings Library (OPL) 529 (1998), 39-46.

[17] Hanzhong Gao, Lu Sun and J. X. Wang, Phygeonet: physics-informed geometry adaptive convolutional neural networks for solving parametric pdes on irregular domain, J. Comput. Phys. 428 (2021), 110079.

[18] Souvik Goswami, Minglang Yin, Yue Yu and George Em Karniadakis, Physics-informed neural networks for surrogate modeling of parametric solutions to the Allen-Cahn and Cahn-Hilliard equations, Computational Mechanics 69(4) (2022), 957-981.

[19] Christopher P. Grant, Spinodal decomposition for the Cahn-Hilliard equation, Comm. Partial Differential Equations 18(3-4) (1993), 453-490.

[20] G. Grun, Existence of solutions for the Cahn-Hilliard equation with degenerate mobility, Nonlinear Analysis: Theory, Methods and Applications 25(8) (1995), 1001-1010.

[21] J. E. Hilliard, The theory of phase separation in alloys, Journal of Chemical Physics 53(8) (1970), 2924-2933.

[22] Junseok Kim, Seunggyu Lee, Yongho Choi, Seok-Min Lee and Darae Jeong, Basic principles and practical applications of the Cahn-Hilliard equation, Math. Probl. Eng. 2016(1) (2016), 9532608.

[23] M. Lupu, A. Miranville, A. Signori, C. Walker and H. Wu, Cahn-Hilliard models for medical imaging: a review, Journal of Biomedical Imaging 2021, Article ID 6666354, pp. 1-15.

[24] A. Miranville,A. Signori and S. Zelik, Image prediction through Cahn-Hilliard image inpainting, Royal Society Open Science 8 (2021), 201294

[25] Flore Nabet, Numerical analysis for generalized Cahn-Hilliard type equations with applications to tumor growth, PhD thesis, École Polytechnique, 2023

[26] Morteza A. Nabian, Mohammad Zaki and George E. Karniadakis, Efficient physics informed neural networks for solving forward and inverse problems involving nonlinear partial differential equations, Computer Methods in Applied Mechanics and Engineering 384 (2021), 113938.

[27] A. A. Nahla, D. A. M. Abo-Kahla and M. H. Raddadi, Entanglement and atomic inversion for two two-level atoms interacting with an intensity-dependent quantized field, Internat. J. Modern Phys. B 36(26) (2022), 2250179.

[28] A. Novick-Cohen, Energy methods for the Cahn-Hilliard equation, Nonlinear Diffusion Equations and their Applications, Springer, 1985, pp. 1-20.

[29] M. H. Raddadi, A. A. Nahla and D. A. M. Abo-Kahla, Quantized study for asymmetric two two-level atoms interacting with intensity-dependent coupling regime, Indian Journal of Physics 97(5) (2023), 1345-1358.

[30] Maziar Raissi, Paris Perdikaris and George E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 391 (2019), 686-707.

[31] Jie Shen and Xiaofeng Yang, Numerical methods for the Cahn-Hilliard equation, Discrete and Continuous Dynamical Systems 28(4) (2010), 1649.

[32] Hao Wu, A review on the Cahn-Hilliard equation: Classical results and recent advances in dynamic boundary conditions, Discrete and Continuous Dynamical Systems - Series S 14(1) (2021), 1-54.

[33] W. Yin, Existence of solutions for the Cahn-Hilliard equation with degenerate mobility in one dimension, J. Differential Equations 97(1) (1992), 1-16.

[34] Y. Zhang , B. Liu and Z. Chen, Fast image inpainting strategy based on the space-fractional modified Cahn-Hilliard equations, Comput. Math. Appl. 81 (2020), 1234-1245.