Using PINNs as solvers for the Cahn-Hilliard equation

John Sipahioglu

Kent State University Stark Campus

North Canton, Ohio

jsipahio@kent.edu

Abstract-- The Cahn-Hilliard equation is a fourth-order, nonlinear partial differential equation that arose in the study of the phase separation of binary alloys. Since then, it has served as the basis for models pertaining to many other problems, including chemical diblock copolymers, topology optimization, image inpainting, segmentation and denoising, modeling tumor growth, modeling systems with elastic inhomogeneities, and multi-phase fluid flows. With its widespread use, many numerical methods have been proposed to solve it, with many of them being very difficult to implement. Here, we introduce the basic theory behind physics informed neural networks. We then present a comparison of some of the existing libraries. The primary conclusion is that most of the libraries right now are not production quality, and they are not extensible to a problem as complex as the Cahn-Hilliard equation. As it currently stands, if one were to use a physics inspired neural network to solve the Cahn-Hilliard equation, a novel implementation would be required.

I. INTRODUCTION

The original Cahn-Hilliard equation (CH equation) was proposed by Cahn and Hilliard in 1958 to find the free energy of a nonuniform system, primarily to model the phase separation in an alloy [1]. Since then, there have been many additional applications discovered for this equation. Due to its many applications, there are many existing solutions to the CH equations [2]. An overview of the existing numerical solutions can be found in section III.

The CH equation is a fourth order, nonlinear partial differential equation (PDE). It is well known that analytic solutions for nonlinear PDEs are often impossible to find, thus numerical approximations must be used. Many of the existing methods are highly complex and very difficult to implement. Here, we will explore a new method for solving PDEs, called physics informed neural networks (PINN).

In short, a PINN is a deep neural network (DNN) that uses the residuals of the PDE and the initial and boundary conditions as loss functions to arrive at a solution. It is "physics informed" in the sense that it is governed by the equation and its solutions must satisfy the initial and boundary conditions through the loss function. In this paper, solutions for the CH equation using some of the existing PINN frameworks will be attempted, and the results of each framework will be compared.

First, the various applications of the CH equation will be explored to demonstrate the importance of a numerical model for solving the equation accurately and quickly. Next, a survey of methods for solving the CH equation will be presented. Next, an outline of the theory behind PINNs will be introduced. Finally, a comparison between many of the existing PINN libraries for Python, as well as one for Julia, will be undertaken.

II. APPLICATIONS

The various applications of the CH equation outlined in the survey by Kim, Lee, and Choi [2] will be discussed in minor detail to demonstrate the importance of the equation in many fields, and to provide motivation for numerical methods to solve it. Additionally, image denoising and segmentation is discussed, as this topic provided the original motivation for the author's research. The application areas discussed are spinodal decomposition, diblock copolymers, image inpainting, multiphase fluid flows, microstructure elasticity, tumor growth, topology optimization, and image denoising and segmentation.

A. Spinodal Decomposition

Cahn in [3] used the CH equation to model spinodal decomposition of alloys. Zang in [4] defines spinodal decomposition as "a mechanism by which a solution of two or more components can separate into distinct phases with distinctly different chemical compositions and physical

properties." Zang also notes that spinodal decomposition is of interest due to its simplicity, allowing for there to be plausible quantitative theory. Rozman [5] presents the derivation of the dimensionless CH equation that will be studied in this paper.

$$\frac{\partial u}{\partial t} = \nabla^2(-\nabla^2 u + u^3 - u) \qquad (1)$$

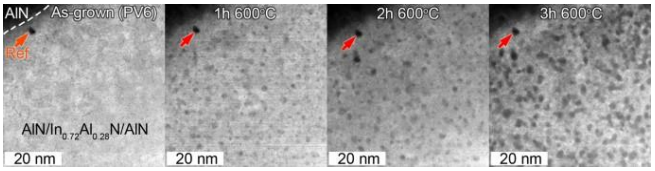

Fig. 1: The spinodal decomposition of InAlN

Source: [6]

B. Diblock Copolymers

The CH equation has also been used in describing the phase-field models diblock copolymers. A diblock copolymer is a monomer A chemically joined to another monomer B to create one polymer. There exists a critical temperature where the monomers can no longer be joined and will undergo a phase separation. Jeong et. al have proposed a model where the energy-minimizing wavelengths of the equilibrium state for a diblock copolymer can be calculated using the CH equation as a model for the copolymer [7].

C. Image Inpainting

Additionally, the CH equation can be used for image inpainting. Image inpainting is the process of recovering an undamaged image from a damaged image. Bertozzi, Esedoglu, and Gillette proposed a model using the CH equation to fill in large regions that are obscured [8]. The resulting images are noticeably blurry, see Figure 2. Vijayakrishna later modifies the work in [8] by converting the double well potential in the CH equation to a multi-well potential [9]. The resulting images are still blurry, however. Kumar, Halim, and Vijayakrishna propose that this is due to the biharmonic term in the modified CH equation and rectify it by introducing a greyscale intensity variation sensitive anisotropic non-linear version of the biharmonic term in their model for image denoising and segmentation [10].

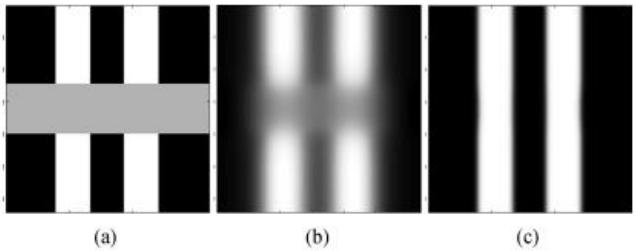

Fig. 2: Image processed by algorithm in [8]. (a) is the initial image (grey bar is inpainting region), (b) is an intermediate time step, and (c) is the resulting image after all time steps have been completed. The resultant image has blurring on the edges and noticeable distortion where the grey bar crossed.

Source: adapted from [8]

D. Multiphase Fluid Flows

Phase-field models replace sharp boundaries with thin but nonzero transitions where interfacial forces are smoothly distributed. The CH equation is often used to model this [11]. The CH equation is combined with the Navier-Stokes equation (CH-NS equation) to model immiscible fluids [12]. The CHNS equation can be derived for both the concentration and density settings of the fluids. The constitutive equations for multi-fluid mixtures in these models are based on the maximization of the rate of entropy production [13].

E. Microstructures with Elastic Inhomogeneity

The CH equation was used in conjunction with elasticity equations by Zhu, Chen, and Shen [14] to study the effects of elastic inhomogeneity in the coarsening of elastically isotropic and elastically anisotropic systems. They found that the rate of coarsening decreases as the elastic inhomogeneity increases. An important conclusion of [14] is that the coarsening does not freeze, which is a contradiction to predictions made by older models.

The use of precipitation hardening is widespread in the manufacturing of high-strength alloys. Most studies of precipitation hardening only look at the problem in the elastic domain. Zaeem et. al [15] combined the CH equation with elasticity equations to draw many conclusions about the behaviors of systems with mismatched elastic strains.

F. Simulations of Tumor Growth

The growth of multispecies tumors can be modeled by replacing sharp edges with thin transition layers. As seen in some of the previous applications, the CH equation works well for this task. CH-type diffusion equations for the cell species combined with reaction-diffusion equations for the substrate components works well for modelling tumor growth [16]. Garcke, Lam, and Signori [17] propose a model using the CH equation and taking elastic effects into account to model the effect of the stresses caused by tumor growth on tumor growth. Experimental studies show that stresses slow the tumor growth, which is not able to be modeled using a gradient flow setting used in previous works.

G. Topology Optimization

Optimizing the objective function for a region Ω in $\mathbb{R}^2$ or $\mathbb{R}^3$ is the goal of structural topology optimization. Original attempts to solve these problems assumed that Ω is

made of a homogenous material, which is not practical in real-world applications [18]. Initial solutions were to homogenize the region, that is, discretize the region into small sections that can be treated as homogenous, and solve for each of the sections [19]. Zhou and Wang [18] proposed using a CH equation system to transform the topology optimization problem into a partial differential equation model. They accomplish this by "[treating] the problem of minimizing the mean compliance of a multimaterial structure as a thermodynamic system with mass concentration as phase field variable."

H. Image Denoising and Segmentation

Recently, the CH equation has been applied in the field of image denoising and segmentation. After the successful use of the CH equation for image inpainting, Kumar, Halim, and Vijayakrishna in [10] apply the CH equation for image denoising and segmentation. They note that these processes go hand in hand as the segmentation process requires the smoothness of edges. Carillo et. al in [20] use the CH equation's application for inpainting to prepare images to be processed by neural networks.

III. Numerical Methods

Here, various methods for the solution of the CH equation will be presented. Due to the many uses presented for the CH equation, there are many existing numerical schemes to solve the equation. Only a high-level overview of these models will be given as this paper is not meant as a survey of existing methods. The aim is to show that there are existing numerical schemes, but they are complex and difficult to implement.

Due to the strong nonlinearity and high order of the CH equation, any explicit scheme must have a time-step $\Delta t = O(h^4)$ [18]. One way to increase the time step is to use an adaptive mesh, as presented in [21]. The equation on a narrow band time-adaptive domain using an explicit Saul'yev scheme with a mass correction algorithm applied after each time step. See Figure 3 for a comparison between a uniform mesh and an adaptive mesh. A Saul'yev scheme is a scheme that can make an implicit finite difference method explicit by solving left-to-right or right-to-left with sufficient boundary conditions. However, solving in this way creates a high degree of error. Alternating between the left-to-right and right-to-left solutions at each time step eliminates most of this error [22]. Using the work of [23] on the Allen-Cahn equation, the authors of [21] apply the mass correction step to maintain the conservatism of the algorithm.

Traditional explicit schemes require extremely small time-steps and often do not obey energy conservation, leading to the use of semi-implicit or even fully implicit schemes [25]. Shen and Chen in [26] present a semi-implicit Fourier spectral scheme for solving the CH equation. In their

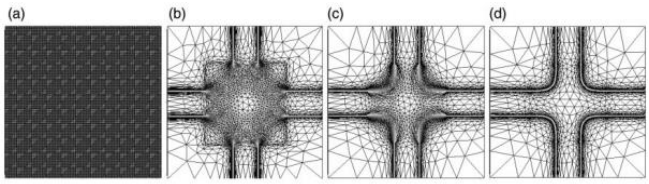

Fig. 3: Comparison between uniform mesh and adaptive mesh. (a) is a uniform mesh, (b) – (d) represents an adaptive mesh for iterations 0, 3, and 20, respectively.

Source: adapted from [24]

treatment, they first take the Fourier transform of the CH equation, then solve the fourth order terms implicitly and the nonlinear terms explicitly. This is done because, while the Fourier transform is accurate spatially, it is only first-order accurate in time and would suffer the same time-step restraints as a traditional explicit scheme. Equation (2) presents their numerical scheme.

$$(3 + 2\Delta t k^4)\hat{u}^{n+1}(\boldsymbol{k}) = 4\hat{u}^{n}(\boldsymbol{k}) - \hat{u}^{n-1}(\boldsymbol{k}) + 2\Delta t k^2\left[\widehat{2\{f(u^n)\}}_k - \widehat{2\{f(u^{n-1})\}}_k\right] \quad (2)$$

It is known that in practice, however, that the time-steps for this type of scheme must be small as truncation errors can cause the system to lose energy stability [25]. He, Liu, and Tang [27] propose that adding an additional term to the numerical scheme can increase the stability of the system, thus allowing for a larger time-step to be used.

The stabilization term in [27] depends on a priori assumptions on the solution to the equation [25]. In [28], Shen and Yang use Lipschitz nonlinearities to "modify" the nonlinear f(u) function. However, this method still requires assumptions to be made on the Lipschitz nonlinearity. In [25], Li and Qiao propose a modified energy functional that does not depend on Lipschitz nonlinearities and a stabilization term to create an unconditional stable semi-implicit scheme.

Fully implicit schemes are unconditionally energy stable, however they require solving a nonlinear system at each time step [28]. The authors of [29] present an implicit scheme, displayed in equations (3) and (4), for solving the CH equation. They use a traditional finite-difference scheme to discretize the space variables, and a Crank-Nicolson method for temporal discretization.

$$\frac{u_{ij}^{n+1} - u_{ij}^{n}}{\Delta t} = \nabla_d[M(u)_{ij}^{n+\frac{1}{2}} \nabla_d \mu_{ij}^{n+\frac{1}{2}}] \quad (3)$$

$$\mu_{ij}^{n+\frac{1}{2}} = \frac{1}{2}\left(F'\left(u_{ij}^{n+1}\right) + F'\left(u_{ij}^{n}\right)\right) - \frac{\epsilon^2}{2}\nabla_d(u_{ij}^{n+1} + u_{ij}^{n}) \quad (4)$$

IV. PHYSICS INFORMED NEURAL NETWORKS

Recently, developments have been made in the use of physics informed neural networks (PINNs) as a method to solve differential equations. Unlike traditional methods for solving PDEs, PINNs are meshless, making them easier to scale to higher dimensions. PINNs are supervised machine learning models that obey the initial conditions (ICs), boundary conditions (BCs), and the governing PDE. We treat the independent variables of the PDE as the input and the solution of the PDE as the output. PNNs use a multitask loss function, where the ICs and BCs act as the mean squared error (MSE) and the PDE loss is represented by the residual of the governing PDE at a random collection of points [30].

Advancements in automatic differentiation and the availability of open-source packages for neural networks led to increased research into neural networks as solvers for PDEs [31]. PINNs in their current form are attributed to Raissi, Perdikaris, and Karniadakis in their two-part paper on data-driven solutions to PDEs [32][33]. In their papers, they deal with the sparse information for training by applying the decades of physical knowledge already gained from previous studies of these problems.

Mattey and Ghosh [34] report that the traditional PINN models do not handle the strong nonlinearity and high order of the CH equation well. To rectify this, they propose a variation called the backwards-compatible PINN (bcPINN) that retrains itself over time-steps, looking to still satisfy the solutions at the previous time-steps. Phase field models often have sharp, abrupt changes in space as they progress over time. Traditional PINNs have difficulties with sudden changes, which is why their performance suffers when used for the CH equation [35]. Wight and Zhao [35] present an adaptive sampling method for PINNs to focus higher sampling rates in areas where sharp changes occur.

V. METHOD

Two of the most popular PINNs right now are the DeepXDE [36] and Nvidia Modulus libraries for Python. Additionally, searching the internet yielded several other existing frameworks. These included IDRLnet [37] and SciANN [38]. Additionally, the capabilities of the Julia library NeuralPDE.jl [39] will be explored.

For all cases, the CH equation will be based on the form shown in equation (1). If required for the implementation, the expanded form with the Laplacian del operators applied will be used:

$$u_t = -\left(u_{xxxx} + 2u_{xxyy} + u_{yyyy}\right) - \left(u_{xx} + u_{yy}\right) + 3u^2\left(u_{xx} + u_{yy}\right) + 6u\left(u_x^2 + u_y^2\right) \quad (5)$$

Due to the limitations on how functionals are defined in some of the libraries the above form may be required, although the presence of the mixed derivative term also presents issues.

VI. RESULTS

In testing it was found that none of these libraries were robust enough to effectively solve the CH equation. In fact, many of them were not even capable of solving their own example problems. The findings for each library tested are presented below.

A. DeepXDE

The DeepXDE initially seemed the most promising. DeepXDE ended up being the second-best library in terms of documentation and functionality, though. All the examples provided in their documentation worked, however none considered a two-dimensional time-dependent system. Initially, a solution to the simpler linear parabolic heat equation (equation (6)) were explored, however partway through training, between 1,000 and 2,000 epochs, some part of the code in the backend broke down and the program died. As the CH equation is more complex variation off of the heat equation, its solution was not further explored using this library.

$$u_t = c(u_{xx} + u_{yy}) \quad (6)$$

B. Nvidia Modulus

Based on literature and the Nvidia name, initial expectations for this library were high. However, those were short lived. The pip package for modulus does not actually allow access to the Modulus library. Following the instructions on the Nvidia Modulus website led to installing Ubuntu 20.04 on a computer equipped with an Nvidia graphics card (GPU), and then going through many hoops to get the correct version of Docker and the Modulus docker image installed. Once installed, it was not apparent how to use the image in any meaningful way, and the documentation was sparse. After two full days of attempting to make Modulus work, it was given up on.

C. IDRL.net

The most nonfunctional of the libraries that were able to be tested. Was not capable of running any examples due to multiple issues within the source code, including deprecated syntax and missing class definitions.

E. SciANN

The second-best Python library and third-best overall, SciANN was able to solve the one-dimensional examples provided with its documentation. Some two-dimensional examples were also available, but due to time restraints they were not tested.

F. NeuralPDE.jl

This was the best PINN solver tested. Its documentation was very strong, and its examples were far

from trivial. It was able to solve a two-dimensional wave equation in a reasonably short amount of time. Issues with this network are modeling the free energy functional, as it requires a mixed derivative that was unable to be implemented.

G. Discussion

Despite the novelty of PINNs, many of the existing libraries for PINNs appear to be using antiquated syntax, packages, dependencies, and operating systems. Additionally, much of the GPU support depends on Cuda, which is specific to Nvidia, forcing the use of an Nvidia GPU and docker images if GPU support is desired. The most disappointing library was Modulus, as it was expected that a library created by a large corporation like Nvidia would be easier to use and have better documentation. The NeuralPDE.jl library was the most impressive, as it was unexpected that a library for a newer language would offer better documentation, easier use, and better performance than those for Python, which has been around for some time and is incredibly popular.

VII. DIFFICULTIES

Many difficulties were faced during this research. First, the initial paper by Zheng [40] was poorly cited and lacking in substance. Eventually, this led to the adoption of [10] as the primary basis for continued research. Initially, this source seemed promising as it included pseudocode which the author naively believed could be implemented easily. When it became clear that this would not be the case, a shift was made to attempt to understand numerical methods from the ground up. The videos created by Dr. Steve Brunton have proved invaluable in the author's basic understanding of various numerical schemes and the fast Fourier transform [41].

Several attempts were made to solve the problem, with various attempts to implement equations and pseudocode found in multiple papers. However, if the author were to be truthful, he would report that he was significantly out of his depth. Having no experience in numerical analysis led to a great degree of trying various techniques to see if any usable result could be demonstrated. Everything attempted resulted in failure. Explicit finite difference methods, even with extremely small time-steps, led to unbounded energy growth or the loss of mass conservation. Suitable stabilization schemes were not found for this, and these attempts were shelved.

When it became clear that any functional implementation would likely take the whole semester, if not longer, the original investigation into image denoising and segmentation was abandoned in search of an implementable numerical scheme. The goal of the author became to find a way scheme that would favor ease of implementation over unconditional stability and speed. This was decided due to the high level of computational power that is now available. As discovered in this research, if numerical analysis is not the researcher's area of expertise, it is easy to spend all the time that could be spent researching an application of an equation searching for a suitable way to solve it.

If provided with additional time, more networking could be done with experts in the field of numerical analysis to refine the presented scheme to be more stable and functional. Additionally, the conditions for convergence, energy stability, and mass conservation could be derived and proved. Overall, with more time and resources, a more thorough investigation could be completed, possibly yielding the desired result of a more readily accessible scheme for someone with novice numerical analysis skills to apply.

VIII. CONCLUSION

In this paper, the various applications of the CH equation were presented to motivate the need for a simple numerical scheme to find the solution. These applications included its original application for modeling spinodal decomposition, as well as the modeling of diblock copolymers, image inpainting, image denoising and segmentation, tumor growth, topology optimization, elastic inhomogeneity of microstructures, and multi-phase fluid flows. Following, the complexity and challenges of various explicit, semi-implicit, and fully implicit schemes were discussed. Additionally, the relatively new field of PNNs was introduced as a possible method of solving complex equations. In exploring the existing PINN libraries, it was found that most of them lacked the needed functionality to implement a solution to the CH equation. Future work would be implementing a PINN from scratch that can solve more complex equations.

REFERENCES

[1] J.W. Cahn, J.E. Hilliard. "Free Energy of a Nonuniform System. I. Interfacial Free Energy." *J. Chem. Phys.* vol. 28, no. 2, pp. 258-267. Feb. 1958. doi: https://doi.org/10.1063/1.1744102 [online] Available: https://aip.scitation.org/doi/10.1063/1.1744102

[2] J. Kim, S. Lee, Y. Choi, S.-M. L, D. Jeong. "Basic Principles and Practical Applications of the Cahn–Hilliard Equation." *Math. Problems Eng.* vol 2016, Oct. 2016. Art. no. 9532608. doi: https://doi.org/10.1155/2016/9532608. [online] Available: https://www.hindawi.com/journals/mpe/2016/9532608/

[3] J.W. Cahn. "On spinodal decomposition." *Acta Metallurgica* vol. 9, no. 9, pp. 795-801, Sep. 1961. doi: https://doi.org/10.1016/0001-6160(61)90182-1 [online] Available: https://www.sciencedirect.com/science/article/pii/0001616061901821

[4] L. Zang. (2004). "Lecture 22: Spinodal Decomposition: Part 1: general description and practical implications." [PDF]. Available: https://my.eng.utah.edu/~lzang/images/lecture-22.pdf

[5] M. Rozman. (2016). "Cahn-Hilliard equation and kinetics of spinodal decomposition" [PDF]. Available: https://www.phys.uconn.edu/~rozman/Courses/P2200_16F/downloads/cahn-intro.pdf

[6] J. Palisaitis, C.-L. Hsiao, L. Hultman, J. Birch, P.O.Å. Persson. "Direct observation of spinodal decomposition phenomena in InAlN alloys during in-situ STEM heating." *Sci. Rep.* vol. 7, Mar. 2017, Art. no. 44390. doi: https://doi.org/10.1038/srep44390. [online] Available: https://www.nature.com/articles/srep44390

[7] D. Jeong, J. Shin, Y. Li, Y. Choi, J.-H. Jung, S. Lee, J. Kim. "Numerical analysis of energy-minimizing wavelengths of equilibrium states for diblock copolymers." *Cur. Appl. Phys.* vol. 14, no. 9, pp. 1263-1272, Sep. 2014. doi: https://doi.org/10.1016/j.cap.2014.06.016 [online] Available: https://www.sciencedirect.com/science/article/pii/S1567173914001849

[8] A. L. Bertozzi, S. Esedoglu, A. Gillette. "Inpainting of Binary Images Using the Cahn–Hilliard Equation." *IEEE Trans. Image Process.* vol. 16, no. 1, pp. 285-291, Dec. 2006. doi: https://doi.org/10.1109/TIP.2006.887728 [online] Available: https://ieeexplore.ieee.org/abstract/document/4032803

[9] Vijayakrishna, R.: "A unified model of Cahn-Hilliard grayscale inpainting and multiphase classification." PhD thesis, Indian Institute of Technology Kanpur, India, 2015

[10] B.V.R. Kumar, A. Halim, R. Vijayakrishna. "Higher order PDE based model for segmenting noisy image." *IET Image Process.* vol. 14, no. 11 pp. 2597-2609, Sep. 2020. doi: 10.1049/iet-ipr.2019.0885. [online] Available: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2019.0885

[11] V.E. Badalassi, H.D. Ceniceros, S. Banerjee. "Computation of multiphase systems with phase field models." *J. Comp. Phys.* vol. 190, no. 2, pp. 371-391, Sep. 2003. doi: https://doi.org/10.1016/S0021-9991(03)00280-8. [online] Available: https://www.sciencedirect.com/science/article/pii/S0021999103002808

[12] M. Heida. "On the derivation of thermodynamically consistent boundary conditions for the Cahn–Hilliard–Navier–Stokes system." *Int. J. Eng. Sci.* vol. 62, pp. 126-156, Jan. 2013. doi: https://doi.org/10.1016/j.ijengsci.2012.09.005. [online] Available: https://www.sciencedirect.com/science/article/pii/S002072251200184X#b0080

[13] M. Heida. "On the derivation of thermodynamically consistent boundary conditions for the Cahn–Hilliard–Navier–Stokes system." Int. J. Eng. Sci. vol. 62, pp. 126-156, Jan. 2013. doi: https://doi.org/10.1016/j.ijengsci.2012.09.005. [online] Available: https://www.sciencedirect.com/science/article/pii/S002072251200184X#b0080

[14] J. Zhu, L.-Q. Chen, J. Shen. "Morphological evolution during phase separation and coarsening with strong inhomogeneous elasticity." *Modelling Simul. Mater. Sci. Eng.* vol. 9, no. 6, pp. 499-511, Oct. 2001. doi: 10.1088/0965-0393/9/6/303. [online] Available: https://iopscience.iop.org/article/10.1088/0965-0393/9/6/303/meta

[15] M.A. Zaeem, H.E. Kadiri, M.F. Horstemeyer, M. Khafizov, Z. Utegulov. "Effects of internal stresses and intermediate phases on the coarsening of coherent precipitates: A phase-field study." *Cur. Appl. Phys.* vol. 12, no. 2, pp. 570-580, Mar 2012. doi: https://doi.org/10.1016/j.cap.2011.09.004. [online] Available: https://www.sciencedirect.com/science/article/pii/S1567173911004676

[16] S.M. Wise, J.S. Lowengrub, H.B. Frieboes, V. Cristini. "Three-dimensional multispecies nonlinear tumor growth—I: Model and numerical method." *J. Theor. Biol.* vol. 253, no. 3, pp. 524-543, Aug. 2008. doi: https://doi.org/10.1016/j.jtbi.2008.03.027 [online} Available: https://www.sciencedirect.com/science/article/pii/S0022519308001525

[17] H. Garke, K.F. Lam, A. Signori. "On a phase field model of Cahn–Hilliard type for tumour growth with mechanical effects." *Nonlinear Anal.: Real World Appl.* vol. 57, Feb. 2021, Art. no. 103192. doi: https://doi.org/10.1016/j.nonrwa.2020.103192 [online] Available: https://www.sciencedirect.com/science/article/pii/S1468121820301103

[18] S. Zhou, M.Y. Wang. "Multimaterial structural topology optimization with a generalized Cahn–Hilliard model of multiphase transition." *Struct. Multidisciplinary Optim.* vol. 33, pp. 89-111, Feb. 2007. doi: https://doi.org/10.1007/s00158-006-0035-9 [online] Available: https://link.springer.com/article/10.1007/s00158-006-0035-9

[19] M.P. Bendsøe, N. Kikuchi. "Generating optimal topologies in structural design using a homogenization method." *Comp. Methods Appl. Mech. and Eng.* vol. 71, no. 2, pp. 197-224, Nov. 1988. doi: https://doi.org/10.1016/0045-7825(88)90086-2. [online] Available: https://www.sciencedirect.com/science/article/pii/0045782588900862

[20] J.A. Carillo, S. Kalliadasis, F. Liang, S.P. Perez. "Enhancement of damaged-image prediction through Cahn–Hilliard image inpainting." *Roy. Soc. Open Sci.* vol. 8, no. 5, May. 2021. doi: https://doi.org/10.1098/rsos.201294. [online] Available: https://royalsocietypublishing.org/doi/full/10.1098/rsos.201294

[21] S. Ham, Y. Li, C. Lee, S. Kwak, Y. Hwang, J. Kim. "An Explicit Adaptive Finite Difference Method for the Cahn–Hilliard Equation." *J. Nonlinear Sci.* vol. 32, Sep. 2022, Art. no. 80, doi: https://doi.org/10.1007/s00332-022-09844-3. [online] Available: https://link.springer.com/article/10.1007/s00332-022-09844-3.

[22] S.A. Chin. "Understanding Saul'yev-Type Unconditionally Stable Schemes from Exponential Splitting." *Num. Methods Partial Differ. Equ.* vol. 30, no. 6, Nov. 2014. doi: 10.1002/num.21885 [online] Available: https://www.researchgate.net/publication/262574143_Understanding_Saul'yev-Type_Unconditionally_Stable_Schemes_from_Exponential_Splitting

[23] D. Jeong, Y. Li, C. Lee, J. Yang, J. Kim. "A conservative numerical method for the Cahn-Hilliard equation with generalized mobilities on curved surfaces in three-dimensional space." *Commun. Comp. Phys*. vol. 27, no. 2, Feb. 2020. doi: 10.4208/cicp.OA-2018-0202

[24] A. Theljani, H. Houichet, A. Mohamed. "An adaptive Cahn-Hilliard equation for enhanced edges in binary image inpainting." *J. Alg. Comp. Tech.* vol. 14, Aug. 2020. doi:10.1177/1748302620941430 [online] Available: https://journals.sagepub.com/doi/full/10.1177/1748302620941430

[25] D. Li, Z. Qiao. “On Second Order Semi-implicit Fourier Spectral Methods for 2D Cahn–Hilliard Equations.” *J. Sci. Comp.* vol. 70, pp. 301-341, Jan. 2017. doi: https://doi.org/10.1007/s10915-016-0251-4. [online] Available: https://link.springer.com/article/10.1007/s10915-016-0251-4

[26] L.Q. Chen, J. Shen. “Applications of semi-implicit Fourier-spectral method to phase field equations.” *Comp. Phys. Commun.* vol. 108, no. 2-3, pp. 147-158, Feb. 1998. doi: https://doi.org/10.1016/S0010-4655(97)00115-X. [online] Available: https://www.sciencedirect.com/science/article/pii/S001046559700115X

[27] Y. He, Y. Liu, T. Tang. “On large time-stepping methods for the Cahn–Hilliard equation.” *Appl. Num. Methods.* vol. 57, no-5-7, pp. 616-628, May 2007. doi: https://doi.org/10.1016/j.apnum.2006.07.026. [online] Available: https://www.sciencedirect.com/science/article/pii/S0168927406001437

[28] J. Shen, X. Yang. “NUMERICAL APPROXIMATIONS OF ALLEN-CAHN AND CAHN-HILLIARD EQUATIONS.” *Discrete Continuous Dyn. Syst.* vol. 28, no. 4, pp. 1669-1691, Dec. 2010. doi: 10.3934/dcds.2010.28.1669. [online] Available: https://www.math.purdue.edu/~shen7/pub/ac_ch.pdf

[29] D. Lee, Y.-H. Huh, D. Jeong, J. Shin, A. Yun, J. Kim. “Physical, mathematical, and numerical derivations of the Cahn–Hilliard equation.” *Comp. Mater. Sci.* vol. 81, pp. 216-225, Jan. 2014. doi: https://doi.org/10.1016/j.commatsci.2013.08.027. [online] Available: https://www.sciencedirect.com/science/article/pii/S0927025613004801

[30] P. Sharma, L. Evans, M. Tindall, P. Nithiarasu. “Stiff-PDEs and Physics-Informed Neural Networks.” *Arch. Comp. Methods Eng.* Feb. 2023. doi: https://doi.org/10.1007/s11831-023-09890-4 [online] Available: https://link.springer.com/article/10.1007/s11831-023-09890-4

[31] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli. “Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What’s Next.” *J. Sci. Comp.* vol. 92, Jul. 2022, Art. no. 88. doi: https://doi.org/10.1007/s10915-022-01939-z. [online] Available: https://link.springer.com/article/10.1007/s10915-022-01939-z

[32] M. Raissi, P. Perdikaris, G.E. Karniadakis. “Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations.” *arXiv:1711.10561 [cs.AI].* vol. 2017c, Nov. 2017. doi: https://doi.org/10.48550/arXiv.1711.10561. [online] Available: https://arxiv.org/abs/1711.10561

[33] M. Raissi, P. Perdikaris, G.E. Karniadakis. “Physics Informed Deep Learning (Part Ii): Data-driven Solutions of Nonlinear Partial Differential Equations.” *arXiv:1711.10566 [cs.AI].* vol. 2017d, Nov. 2017. doi: https://doi.org/10.48550/arXiv.1711.10566. [online] Available: https://arxiv.org/abs/1711.10566

[34] R. Mattey, S. Ghosh. “A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations.” *Comp. Methods Appl. Mech. Eng.* vol. 390, Feb. 2022, Art. no. 114474. doi: https://doi.org/10.1016/j.cma.2021.114474. [online] Available: https://www.sciencedirect.com/science/article/pii/S0045782521006939?via%3Dihub

[35] C.L. Wight, J. Zhao. “Solving Allen-Cahn and Cahn-Hilliard Equations using the Adaptive Physics Informed Neural Networks.” *arXiv:2007.04542 [math.NA].* 2020. doi: https://doi.org/10.48550/arXiv.2007.04542 [online] Available: https://arxiv.org/abs/2007.04542

[36] L. Lu, X. Meng, Z. Mao, G.E. Karniadakis. “DeepXDE: A Deep Learning Library for Solving Differential Equations.” *SIAM Review.* vol. 63, no. 1, pp. 208-228, Feb. 2021. doi: https://doi.org/10.1137/19M1274067 [online] Available: https://epubs.siam.org/doi/abs/10.1137/19M1274067

[37] W. Peng, J. Zhang, W. Zhou, X. Zhoa, W. Yao, X. Chen. “IDRLnet: A Physics-Informed Neural Network Library.” *arXiv.* 2021, Art. no. 2107.04320 doi: https://doi.org/10.48550/arXiv.2107.04320 [online] Available: https://arxiv.org/abs/2107.04320

[38] E. Haghighat, R. Juanes. “SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks.” *Comp. Methods Appl. Mech. Eng.* vol. 373, no. 1, Jan. 2021 Art. no. 113552. doi: https://doi.org/10.1016/j.cma.2020.113552 [online] Available: https://www.sciencedirect.com/science/article/abs/pii/S0045782520307374?casa_token=Knt6Ug2Zh6kAAAAA:L2zzCRwIjvxNgraHB4UmRkkpOXt58RVRtx4uw-dvERaxidFRexSEPTCjEOhpv52QOnxCE4-CJWg

[39] K. Zubov et. al. ” NeuralPDE: Automating Physics-Informed Neural Networks (PINNs) with Error Approximations.” *arXiv.* 2021, Art. no. 2107.09443 doi: https://doi.org/10.48550/arXiv.2107.09443 [online] Available: https://arxiv.org/abs/2107.09443

[40] Q. Zhang. “Construction of Robot Computer Image Segmentation Model Based on Partial Differential Equation.” *J. Sensors.* vol. 2022, Jul. 2022 Art. no. 6216423. doi: https://doi.org/10.1155/2022/6216423 [online] Available: https://www.hindawi.com/journals/js/2022/6216423/

[41] Steve Brunton. “Fourier Analysis [Data-Driven Science and Engineering.” (Mar. 7, 2020 to Aug. 7, 2020.) Accessed: Apr. 30. 2023. [online playlist] Available: https://www.youtube.com/playlist?list=PLMrJAkhIeNNT_Xh3Oy0Y4LTj0Oxo8GqsC