INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 20, Number 5, Pages 693–708

doi: 10.4208/ijnam2023-1030

NUMERICAL APPROXIMATIONS OF THE ALLEN-CAHN-OHTA-KAWASAKI EQUATION WITH MODIFIED PHYSICS-INFORMED NEURAL NETWORKS (PINNS)

JINGJING XU, JIA ZHAO, AND YANXIANG ZHAO*

Abstract. The physics-informed neural networks (PINNs) has been widely utilized to numerically approximate PDE problems. While PINNs has achieved good results in producing solutions for many partial differential equations, studies have shown that it does not perform well on phase field models. In this paper, we partially address this issue by introducing a modified physics-informed neural networks. In particular, they are used to numerically approximate Allen-Cahn-Ohta-Kawasaki (ACOK) equation with a volume constraint. Technically, the inverse of Laplacian in the ACOK model presents many challenges to the baseline PINNs. To take the zero-mean condition of the inverse of Laplacian, as well as the volume constraint, into consideration, we also introduce a separate neural network, which takes the second set of sampling points in the approximation process. Numerical results are shown to demonstrate the effectiveness of the modified PINNs. An additional benefit of this research is that the modified PINNs can also be applied to learn more general nonlocal phase-field models, even with an unknown nonlocal kernel.

Key words. Physics-informed neural networks, Allen-Cahn-Ohta-Kawasaki equation, phase field models.

1. Introduction

Ohta-Kawasaki(OK) free energy [22], usually associated with a volume constraint, has been used to simulate the phase separation of diblock copolymers, i.e., polymers consisting of two types of monomers, A and B, that are chemically incompatible and connected by a covalent chemical bond. It is formulated as

$$E[u] = \int_{\mathbb{T}^d} \left[\frac{\epsilon}{2}|\nabla u|^2 + \frac{1}{\epsilon}W(u)\right] dx + \frac{\gamma}{2}\int_{\mathbb{T}^d} \left|(-\Delta)^{-\frac{1}{2}}\Big(f(u)-\omega\Big)\right|^2 dx + \frac{M}{2}\left[\int_{\mathbb{T}^d} (f(u)-\omega)\ dx\right]^2. \tag{1}$$

Here $\mathbb{T}^d = \prod_i [-X_i, X_i) \subset \mathbb{R}^d,\ d = 1,2,3$ is a periodic domain, and $u = u(x)$ is a phase field labeling function representing the density of A species with interfacial width ϵ. Function $W(u) = 18(u^2-u)^2$ is a double well potential. The nonlinear function $f(u) = 6u^5 - 15u^4 + 10u^3$ keeps the 0-1 structure of u as $f(0) = 0$ and $f(1) = 1$. More importantly it also has the property that $f'(0) = f'(1) = 0$, which localize the force and avoid possible non-zeros and non-ones near the boundary of the interface [29] when considering L^2 gradient flow dynamics of (1). The first integral is a local surface energy, the second term represents the long-range interaction between the molecules with γ controlling its strength and $\omega \in (0,1)$ being the fraction of species A, and the third term is a penalty term to fulfill the volume constraint

$$\int_{\mathbb{T}^d} (f(u)-\omega)\ dx = 0. \tag{2}$$

Received by the editors on July 14, 2022 and, accepted on May 30, 2023.

2000 *Mathematics Subject Classification.* 65L15, 34L16.

*Corresponding author.

Allen-Cahn-Ohta-Kawasaki(ACOK) model [29], which is the focus of this paper, takes the L^2 gradient dynamics of the OK free energy in (1) as

$$\begin{aligned} u_t =& -\frac{\delta E[u]}{\delta u} \\ =&\varepsilon\Delta u - \frac{1}{\varepsilon}W'(u) - \gamma(-\Delta)^{-1}(f(u)-\omega)f'(u) - M\left[\int_{\mathbb{T}^d}(f(u)-\omega)\ dx\right]f'(u). \end{aligned} \tag{3}$$

Here the long-range interaction term satisfies the zero-mean condition:

$$\int_{\mathbb{T}^d}(-\Delta)^{-1}(f(u)-\omega)dx = 0. \tag{4}$$

Various successful and efficient partial differential equation (PDE) solvers can be applied to the model, such as Finite Difference, Finite Element [20], Spectral method [10, 15], etc. In recent years, some novel numerical methods have also been studied for Cahn-Hilliard dynamics, i.e., the H^{-1} gradient flow dynamics of OK energy, such as the midpoint spectral method [1], and the Invariant Energy Quadratization (IEQ) method [5]. In [29], a first-order energy stable linear semi-implicit method is introduced and analyzed for the ACOK equation. Our goal in this paper is to study the approximation of the solution of the ACOK equation with neural networks.

Recently, there has been an increasing interest in solving PDEs with machine learning methods, among which the physics-informed neural networks(PINNs) [24] have gained tremendous popularity. PINNs are based on a fully-connected feed-forward deep neural network (DNN) [14, 26], which is often interpreted utilizing universal approximation theorem [6, 11, 12, 18]. The structure of a fully-connected DNN, consisting of an input layer, hidden layers, and an output layer, is shown in Figure 1. Every node in one layer is connected with every node in another by a series of computations. The nodes in the input layer are the specific data studied, and the nodes in the output are the expected outcome. For example, in the case of facial recognition, the input could be the RBG values (features) of each pixel of the sample pictures, while the output could be the assigned numerical values representing the individuals (classes). The nodes in the hidden layers are called neurons, and they are responsible for the performance of the neural network. There can be many hidden layers in a neural network; in the case of Figure 1, the number of hidden layers is three.

A column vector is fed to each node of the input layer. Then in the first hidden layer, we multiply it by another row vector, i.e., weight, to get the product. This is the case that there is only one neuron in the relevant hidden layer. If there are more neurons, the row vector will be a weight matrix instead, where the number of rows of the matrix represents the number of neurons. A bias vector is then added to the product, and an activation function is applied to the new vector, which contributes to the output of the current hidden layer, which is also the input of the next layer. In a nutshell, let M be the number of hidden layers, p be the input of the neural network, then the output a of the neural network will be a^M, with

$$a^0 = p, \tag{5}$$

$$a^i = f^i(W^i a^{i-1} + b^i),\ i = 1, ..., M-1, \tag{6}$$

where a^{i-1} is the input, W^i represents the weights, b^i represents the bias, and f^i is the activation function in the ith layer.