

基于物理信息神经网络的Allen-Cahn方程应用研究

刘昫阳

北方工业大学理学院, 北京

收稿日期: 2026年7月20日; 录用日期: 2026年8月13日; 发布日期: 2026年8月26日

摘 要

近年来, 随着深度学习这一领域的不断崛起与发展, 其中的物理信息神经网络(PINN)可以通过其无网格化的求解框架实现对于偏微分方程的精确求解功能。基于上述观点, 本文提出以Allen-Cahn方程为研究模型, 结合PINN中的激活函数和训练次数等要素对方程进行分析, 研究在不同参数设置下解的误差情况, 并得出相对最优的配置方案。此外, 本文还针对其中的扩散系数 D 进行反演, 发现随着训练次数的增加, 反演的参数在不断逼近真实值。此外, PINN在反演时不依赖初始值的设定, 无论距离真实值距离远或近都可以实现高精度的反演。

关键词

物理信息神经网络, Allen-Cahn方程, 网络参数设置, 反问题, 参数反演

Study on the Application of the Allen-Cahn Equation Based on Physics-Informed Neural Networks

Yunyang Liu

College of Science, North China University of Technology, Beijing

Received: July 20, 2026; accepted: August 13, 2026; published: August 26, 2026

Abstract

In recent years, with the rapid rise and development of deep learning, physics-informed neural networks (PINNs) have emerged as a powerful tool for solving partial differential equations through their

mesh-free framework. Based on this perspective, this paper takes the Allen-Cahn equation as a model problem and analyzes the solution accuracy by incorporating factors such as activation functions and training iterations within PINNs. The study investigates the error behavior under various parameter settings and identifies relatively optimal configurations. Furthermore, the paper conducts inversion of the diffusion coefficient D and finds that as the number of training iterations increases, the inverted parameters progressively approach the true values. Notably, PINNs do not rely on initial guesses during inversion; they achieve high-precision results regardless of how far or close the initial values are to the actual ones.

Keywords

Physics-Informed Neural Network, Allen-Cahn Equation, Network Parameter Setting, Inverse Problem, Parameter Inversion

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

深度学习(Deep Learning), 是基于深层神经网络模型和方法的机器学习。它是机器学习的重要分支, 核心是通过多层神经网络或深度神经网络来对人类大脑的层级化信息处理方式进行模拟, 是当前人工智能领域的坚实基础。

随着人工智能与物理模型的结合越发紧密, 物理信息神经网络(Physics-Informed Neural Networks, 简称 PINN)作为一种结合了深度学习与物理学知识的机器学习模型应运而生, 其原理就是通过数据误差项和物理信息误差项对损失函数进行设计并通过把偏微分方程的残差等参数融入损失函数, 并由此指导模型的训练, 提高其泛化能力。从技术演化的角度来看, PINN 的发展主要可以分为三个关键阶段。首先是聚焦基础框架与理论结构建立的第一阶段。在本阶段中, Raissi 等人通过对 Allen-Cahn 方程等几类非线性偏微分方程进行求解, 得出了 PINN 相较于传统的神经网络在非线性偏微分方程求解的领域具有效率高、准确性好的特点[1]。其次是 PINN 的理论体系不断完善的第二阶段。在本阶段中, Siddhartha Mishra 等人为 PINN 在逼近 PDE 正问题解时的泛化误差提供了严格的上界, 将 PINN 与经典数值方法在误差分析层面建立了联系[2]; Sifan Wang 等人首次建立了 PINN 与神经正切核(NTK)的对应关系, 同时得出了 PINN 训练困难的核心机理是方程残差损失、边界条件损失、初始条件损失的收敛速率存在差异这一重要结论, 进一步完善了 PINN 的基础理论体系[3]。最后是呈现出应用场景多元化趋势的第三阶段。在本阶段中, A. Zhao 等人研究了 PINN 在生理信号建模这一领域的优点, 并覆盖多个临床生理场景, 是 PINN 在生物医学工程应用方向的重要参考[4]; 林旻等人围绕 PINN 在空天领域的应用现状进行了讨论与梳理, 随后提出了一种基于 PINN 的轨道外推研究方案[5]。

在物理信息神经网络这一领域中, 损失函数也起到了很重要的作用。因为 PINN 是通过训练神经网络来使损失函数最小化, 从而达到近似所求偏微分方程解的目的。一般的损失函数主要由数据损失、偏微分方程损失、初始条件损失和边界条件损失几部分组成, 以确保模型不仅能够对数据进行拟合, 还能够满足相关的物理定律。

在非线性偏微分方程的求解中, 与传统数值方法相比, PINN 在研究中已被证实有以下优点:

(1) PINN 求解具有良好的泛化性。

(2) PINN 求解时不依赖方程的类型且能有效处理偏微分方程中的非线性项。而传统数值方法则需要根据方程的类型和条件构造不同的算法。

(3) PINN 求解时具有无网格化、自动微分的工具, 大幅提高求解效率。而传统数值方法则依赖于对问题域的网格划分, 但网格生成过程复杂, 因此导致效率低下。

此外, 若偏微分方程的定解问题中某些已知条件变成未知条件, 需要通过方程中的已知数据及初边值条件来反推这些未知量, 则这类问题称为偏微分方程的反问题。常见的反问题有五种形式, 分别是: 确定偏微分方程的某些结构参数; 确定过程的过去状态; 确定外界过程的作用; 确定状态变量在物体边界上的变化规律及确定物体的边界形状。并且, 在 PINN 求解偏微分方程反问题这一领域也有专家学者对其进行了研究。如 Xi Chen 等人通过 PINN 方法对时空非局部梅尔尼科夫方程的孤子解开展了数据驱动参数辨识研究[6]; Siddhartha Mishra 等人为 PINN 求解许多偏微分方程的反问题提供了严格理论证明并研究了数据同化反问题[7]。

通过上述分析可知, PINN 在求解偏微分方程的正反问题上都有很广泛的应用, 但是在实际生产生活中的应用中往往需要通过神经网络的参数优化得出最优参数配置以实现最高精度的求解, 而且在求解偏微分方程反问题时, 未知参数的初值选取也至关重要。因此, 本文采用物理信息神经网络对具有强非线性 Allen-Cahn 方程进行研究, 观察在不同激活函数的影响下求解的准确性以及在不同网络参数设置下解的误差情况, 并通过对比得出相对最优的参数配置。其次, 本文针对 Allen-Cahn 方程中的扩散系数 D 进行反演研究, 分析了训练次数对反演准确性的影响, 并首次针对 Allen-Cahn 方程的反演问题, 系统验证了 PINN 方法对参数初始值的高度鲁棒性。

2. Allen-Cahn 方程简介

Allen-Cahn 方程(Allen-Cahn equation), 是一种描述相变过程中的经典数学模型, 被广泛应用于材料科学、流体力学、图像处理和计算机视觉等多个领域。最早是由物理学家 John W. Cahn 和 Sam Allen 于 1970 年代提出[8], 主要被应用于描述系统中的界面动态以及由两相物质分隔的界面的运动和形态演化。

针对 Allen-Cahn 方程, 国内外许多专家学者们都对其进行了研究。比如王常健等人以带狄利克雷边界条件的 Allen-Cahn 方程为研究对象, 进一步完善了 Dirichlet 边值条件下 Allen-Cahn 方程的奇异极限理论[9]。乔寒月等人主要研究相场模拟中的 Allen-Cahn 模型, 并对一维 Allen-Cahn 方程紧差分方法的数值逼近进行了研究[10]。韦昌等人提出了一种基于龙格库塔法的自回归物理信息神经网络模型(SR-PINN-RK), 并对 Allen-Cahn 方程进行了求解[11]。Pengyu Chen 等人使用物理信息神经网络对一维和二维的 Allen-Cahn 方程进行数据驱动学习, 并提出了一种基于硬约束 PINN 来表示初始条件和边界条件。结果表明, 所提出的方法比使用软约束表示的 PINN 具有更优越的性能[12]。Indrajit Das 等人提出仅通过一个神经网络即可在多个连续区间内依次求解 PDE, 并对 Allen-Cahn 方程进行求解来验证观点[13]。

因此基于上述分析, 本文研究在 Dirichlet 条件下的 Allen-Cahn 方程, 其方程的数学表达式为:

$$\begin{cases} \frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} - f(u) \\ u(x, 0) = g(x) \\ u(a, t) = h_1(t), u(b, t) = h_2(t) \\ x \in [a, b], t \in [0, T] \end{cases} \quad (1)$$

其中, 方程第一行为核心部分, 反映了 u 随时间不断演化的过程, 等式右边为扩散项及非线性项, 其中 D 为扩散项系数; 第二行为初始条件, 反映了初始时刻空间上的分布情况; 第三行为边界条件, 反映了

空间区域两端的参数随时间的变化情况。

3. 物理信息神经网络

3.1. 物理信息神经网络原理

物理信息神经网络(Physics-Informed Neural Networks, 后简称 PINN)是一种将微分方程所描述的物理规律融入到损失函数中并以此引导模型的求解过程, 从而使得求解结果更贴合物理规律约束的神经网络模型。其求解上述 Allen-Cahn 方程的基本结构如图 1 所示。

其中, 左边虚线框内是用于求解 Allen-Cahn 方程的物理信息神经网络的基本结构。方程中的空间坐标 x 和时间坐标 t 作为输入数据被传入输入层, 随后进入隐藏层中通过激活函数等进行进一步的运算处理。最后经由输出层输出方程的近似解 $u(x, t)$ 。而右侧虚线框内是面向 Allen-Cahn 方程的自动微分与损失计算的流程示意图, 其主要作用就是求解方程中包含的一阶与二阶的偏导数, 并构建对应的四类约束, 为后续研究提供所需的残差等数据。

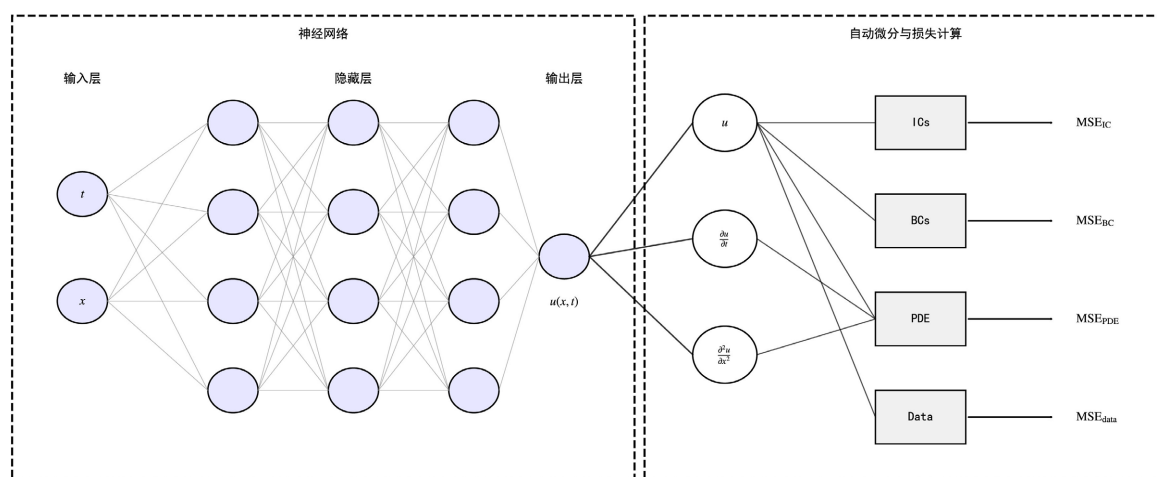


Figure 1. The structure diagram of the physics-informed neural network used for solving the Allen-Cahn equation
图 1. 用于 Allen-Cahn 方程求解的物理信息神经网络结构图

3.2. 激活函数

3.2.1. 双曲正切(Tanh)激活函数

双曲正切(tanh)激活函数是一种在神经网络和递归神经网络等领域应用极其广泛的一种激活函数, 其数学表达式为:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

根据(2)式可以发现, $\tanh$ 函数是一个定义在全体实数集上面的奇函数, 其值域是 $(-1, 1)$ 。该激活函数的优点是其以 0 为对称中心, 且具备中心对称性与 0 均值特性。这就使得在训练时能有助于缓解参数更新所导致的结果不稳定, 进而有效加速模型的收敛。此外, $\tanh$ 函数的导数为:

$$\tanh'(x) = 1 - \tanh^2(x) \quad (3)$$

其在原点附近的梯度无限接近于 1, 可见其有更高的梯度信号, 也就说明其可以有效进行信息的反向传播, 使得 $\tanh$ 函数可以在浅层网络中能够保持高精度的输出。

3.2.2. Sigmoid 激活函数

sigmoid 激活函数, 又称逻辑函数, 是人工神经网络历史中的一个非常重要的激活函数, 其数学表达式为:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

根据(4)式可以看出, sigmoid 函数是一个定义在全体实数集上面的函数, 其值域为(0, 1)。该激活函数的优点是将输出范围限定在了 0 和 1 之间, 相当于做了归一化操作, 适合作为特定模型的输出函数, 比如用于输出一个 0 到 1 的概率值。此外, sigmoid 函数的导数为:

$$\text{sigmoid}'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} \quad (5)$$

通过求导可以看出 sigmoid 函数的梯度是比较平滑的, 这就使得在输出的时候避免了结果的跳跃性。

3.2.3. Leaky RELU 激活函数

Leaky RELU 激活函数是一种修正线性单元, 它在自变量为非负数时保持线性, 而在自变量为负数时引入一个较小的正斜率, 以防止梯度消失等问题的出现。其数学表达式为:

$$\text{Leaky RELU}(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0 \end{cases} \quad (6)$$

从(6)式可以看出, Leaky RELU 激活函数的相对优势就是当 x 小于 0 也就是进入负输入部分时, 也会赋予一个小的梯度, 从而避免了神经元死亡的问题, 使得神经元可以正常参与训练。

3.3. 损失函数

在物理信息神经网络求解 Allen-Cahn 方程的过程中, 损失函数也是一个很重要的因素, 主要用于度量预测值与真实值的差距并基于此让模型通过反向传播对参数进行更新, 来降低真实值和预测值的差距, 从而达到高精度学习的目的。

在本文的研究中, 求解 Allen-Cahn 方程的 PINN 损失函数 L_{total} 采用均方误差 MSE , 并对初始点 D_{ic} 、边界条件点 D_{bc} 、数据点 D_{data} 及方程内部配置点 D_{PDE} 分别求损失函数 L_{ic} 、 L_{bc} 、 L_{data} 、 L_{PDE} 。

其中:

(1) L_{PDE} 为 PDE 残差损失, 即在对应计算域内采样 N_{PDE} 个点, 通过自动微分计算 Allen-Cahn 方程中的一阶与二阶导数, 构造 PDE 残差:

$$R(x, t) = \frac{\partial \hat{u}}{\partial t} - D \frac{\partial^2 \hat{u}}{\partial x^2} - k(\hat{u} - \hat{u}^3) \quad (7)$$

其中, $\hat{u}(x, t, \theta)$ 为神经网络的预测解; θ 为网络可训练参数, PDE 损失定义为所有残差的均方误差。则 L_{PDE} 的表达式为:

$$L_{PDE} = MSE_{PDE} = \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} |R(x_i, t_i)|^2 \quad (8)$$

(2) L_{ic} 为初值损失, 即在初始时刻上采样 N_{ic} 个点, 则初值损失定义为预测值与给定边界条件的均方误差, 其表达式为:

$$L_{ic} = MSE_{ic} = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} |\hat{u}(x_i, 0) - u(x, 0)|^2 \quad (9)$$

(3) L_{bc} 为边界损失, 即在边界上采样 N_{bc} 个点, 则边界损失定义为预测值与给定边界条件的均方误差, 其表达式为:

$$L_{bc} = MSE_{bc} = \frac{1}{N_{bc}} \left(\sum_{i=1}^{N_{bc}} |\hat{u}(-1, t_i) - u_b|^2 + \sum_{i=1}^{N_{bc}} |\hat{u}(1, t_i) - u_b|^2 \right) \quad (10)$$

(4) L_{data} 为观测数据损失, 即引入参考解作为判断求解准确性的依据。在计算域内采样 N_{data} 个点, 数据损失则定义为预测值与真实值的均方误差, 其表达式为:

$$L_{data} = MSE_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} |\hat{u}(x_i, t_i) - u_{true}(x_i, t_i)|^2 \quad (11)$$

则最终神经网络求解 Allen-Cahn 方程的目标就是所有损失项之和:

$$\min_{\theta} L_{total} = L_{PDE} + L_{ic} + L_{bc} + L_{data} \quad (12)$$

4. PINN 求解 Allen-Cahn 方程的反问题

基于上述分析, 已知偏微分方程的反问题包括五种不同的类型, 本文选择其中的第一类进行研究, 即对偏微分方程中的某些结构参数及条件进行反演。也就是在已知方程的某些观测数据及部分参数的情况下, 反推会生成这些数据的特定参数。

4.1. Allen-Cahn 方程的参数反演

考虑如(1)式所示的 Allen-Cahn 方程, 其中扩散项系数 D 为待识别的参数。研究的目标就是在其他条件均为已知的情况下, 给定待观测和学习的数据 $\{x_i, t_i, u_i\}$, 进而实现对扩散项系数的反演, 也就是使得它能够随着训练不断逼近真实值。

4.2. PINN 反演流程

本文将基于物理神经网络对方程中的扩散系数 D 进行反演, 其大致流程为:

- (1) 通过有限差分法生成参考解, 并获得对应的参数 (x, t) 和 $u(x, t)$ 。并选取一定数量的数据点作为观测数据。
- (2) 将扩散系数 D 设置为可学习的参数, 赋予其初始值, 在训练过程中通过梯度下降法实现参数的更新。
- (3) 定义损失函数用于更新神经网络权重和待反演参数 D 。再配置学习率调度器, 保证参数更新稳定。
- (4) 每一轮迭代中通过前向传播计算总损失, 再通过自动微分机制求解损失对神经网络权重和参数 D 的梯度, 同时利用优化器更新权重和 D 的数值。
- (5) 重复(4), 直至训练完成后输出最终反演得到的扩散系数 D 。

5. 数值实验

本节数值算例以带 Dirichlet 条件的 Allen-Cahn 方程为研究模型, 其表达式如下:

$$\begin{cases} \frac{\partial u}{\partial t} = 0.001 \frac{\partial^2 u}{\partial x^2} - 5(u - u^3) \\ u(x, 0) = x^2 \cos(\pi x) \\ u(1, t) = u(-1, t) = -1 \\ x \in [-1, 1], t \in [0, 1] \end{cases} \quad (13)$$

基于上述表达式, 首先研究迭代次数、激活函数等网络参数设置对于训练效果的影响, 再利用 PINN

求解其反问题，即利用 PINN 对 Allen-Cahn 方程中的扩散项系数 D 进行反演，并研究训练次数及初始值设定情况对反演准确性的影响。

本节算例均基于 Python 环境进行研究。所用到的软件环境为 Python 3.7.16、Pytorch 1.13.1 + cpu、Numpy 1.21.6、matplotlib 3.5.3。测试采用的平台为 Intel(R) Core(TM) i7-13650HX (14 核，2.60 GHZ)和 NVIDIA GeForce RTX 4060。

5.1. 激活函数选取

激活函数是神经网络中非常重要的组成部分。它通过引入非线性因素，使得神经网络能够学习复杂的模式和特征。在各方面的研究中，常用的激活函数主要包括 tanh、sigmoid、LeakyReLU 等。在不同的情况下会选取不同的激活函数，它们在复杂度和缓解梯度消失等方面各具优势。

本节算例中隐藏层数固定为 3 层，每层神经元的个数均为 64，学习率为 0.001，训练次数为 10,000 epoch。

激活函数的训练时间和误差对比情况如表 1 所示。由此可知，在其余网络参数均固定的情况下，sigmoid 函数训练时长较长且误差较大。这是因为 sigmoid 函数的值域为(0, 1)，不具备中心对称特性，导致在训练过程中神经元的输入发生偏移，权重更新方向的一致性降低，所以最终训练误差较大。而且 sigmoid 函数在多次反向传播后会出现网络参数更新幅度降低的现象，故相同参数的情况下训练所需时长也会增加；LeakyReLU 函数为分段线性函数，其二阶导数几乎处处为 0，进而导致 Allen-Cahn 方程中依赖高阶算子的 PDE 残差约束失效，同时神经网络易退化为线性变化，丧失训练非线性方程的能力，进而导致模型的误差总在高值却不下降；而 tanh 激活函数值域为(-1, 1)，具备中心对称性，使得权重更新的梯度方向一致性更强。而且 tanh 函数的导数连续且平滑，使得神经网络能够准确地拟合微分算子，所以其误差也是最小的。

综上所述，本文的研究均采用 tanh 作为激活函数。

Table 1. Training duration and error under different activation functions

表 1. 不同激活函数下的训练时长及误差

激活函数	训练时长(秒)	最终训练 L_2 误差
tanh	314.25	0.025715
sigmoid	346.63	0.266478
Leaky Relu	297.07	1.451517

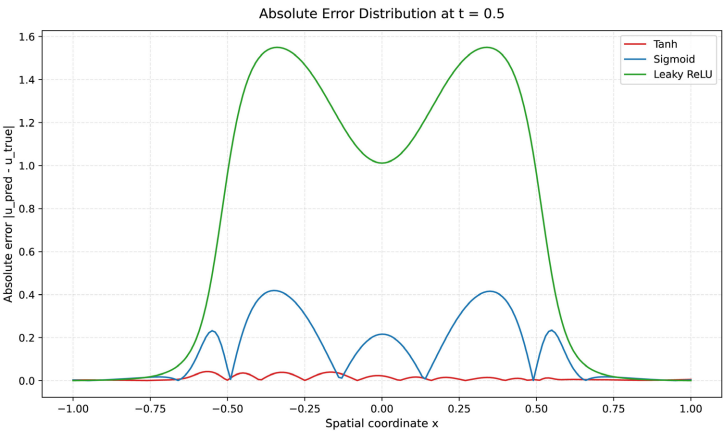


Figure 2. The error line graph of three different activation functions at $t = 0.5$

图 2. 三种不同激活函数在 $t = 0.5$ 时的误差折线图

图 2 进一步说明了三种不同激活函数的误差对比情况，可以看出： $\tanh$ 函数由于全域光滑且可导，而且输出范围也与 Allen-Cahn 方程的定义域匹配，故只在相变界面处有小幅误差波动，是三者中精度最高的。 sigmoid 函数本身就容易出现梯度消失的问题，且输出范围是 $[0, 1]$ ，故需要缩放才可以匹配定义域，所以误差总体略高。而 Leaky RELU 函数是分段线性的，二阶导数处处为 0，无法匹配含有二阶导数项的 Allen-Cahn 方程，使得方程的物理残差很难被有效约束，所以在三者中误差最高。

5.2. 模型参数设置

在已有的研究与分析中，PINN 的网络深度及每层所含的神经元个数要根据具体的研究场景和实际问题而采用不同的参数设置。此外，学习率的大小也影响着模型训练时的性能。本节仍以(13)式为研究对象，探究不同参数对于模型训练的影响。

神经网络隐藏层数和每层神经元个数的设置按下表所示，优化器为 Adam，学习率为默认值 0.001，训练次数为 10,000 epoch。训练完成后的误差如表 2 所示。

Table 2. The impact of hidden layer count and neuron number on equation solving error

表 2. 隐藏层数和神经元个数对方程求解误差的影响

神经元个数 隐藏层数	16	32	48	64	128
1	0.596867	0.530145	0.486776	0.518686	0.523734
2	0.169762	0.215896	0.163866	0.104693	0.069532
3	0.090586	0.047848	0.029711	0.025715	0.026519

由表 2 可知，从总体趋势上看，随着神经元个数和隐藏层数的增加，最终训练的误差在逐渐减小，说明这两个因素确实影响了 PINN 的求解精度和准确性。但是相对应地，增加隐藏层数和神经元个数反而会导致训练的时间成本和所需算力大幅增加且可能会出现数据过拟合现象导致精度不及预期。所以综合上述考量，本文的研究选择隐藏层数为 3，每层神经元的个数为 64。

在上文分析的基础上，优化器仍然选择 Adam，学习率按表 3 所示设置了 4 种，训练次数为 10,000 epoch，训练结果如表 3 所示。

Table 3. Training errors under different learning rates

表 3. 不同学习率下的训练误差

学习率	最终训练 L_2 误差
0.1	0.9683917823
0.01	0.6896298427
0.001	0.0257146248
0.0001	0.3559540520

通过表 3 中的训练结果可以看出：当学习率设置为 0.1 时，过大的更新步长超出了 PINN 的稳定收敛半径，参数迭代时会越过最优解的区域，进入没有物理意义的退化解，所以误差也一直在 0.968 左右稳定而不下降。当学习率为 0.01 时，此时步长仍高于收敛阈值，参数更新时会在最优解附近反复震荡，导致

训练过程中出现反复震荡的趋势。所以不能实现充分收敛，使得误差仍然较大。当学习率为 0.0001 时，此时步长过小，压制了 PINN 每轮迭代的参数更新幅度，使得梯度下降的幅度也不断减小，所以误差下降的速度极为缓慢，无法达到充分收敛的目的。而当学习率为 0.001 时，步长与 PINN 的训练机制匹配度极高，所以能够在保证参数更新效率的情况下不引发误差震荡，所以此时误差也能够快速且平稳地下降到较低水平。综上所述，本文实验选择学习率为 0.001。

5.3. 训练次数与训练结果的关系

在实际的训练过程中，训练次数(epoch)也是很重要的一个考量因素。它指的是整个数据集完成一次前向传播和一次向后传播的过程，即一个 epoch 就是所有训练样本都已经用于训练模型一次。由于神经网络的应用场景复杂，在处理不同的问题时也需要对训练次数进行调整。本文以(14)式为研究模型，探讨训练次数与训练结果的关系。

基于上述分析，本节中神经网络的优化器为 Adam，学习率为默认值 0.001，激活函数为 tanh，隐藏层数为 3，神经元个数为 64，训练总次数设置为 30,000 epoch。

表 4 展示了随着训练次数的增加，方程 L_2 误差的变化情况及数值解与预测解的关系。

Table 4. The relationship between the number of iterations and the L_2 error

表 4. 训练次数与 L_2 误差的关系

训练次数	对应训练次数的 L_2 误差
5000	0.214933
10,000	0.025715
15,000	0.010065
20,000	0.006224
25,000	0.005020
30,000	0.004097

由表 4 可知：从总体趋势上看，随着训练次数的增加，训练的误差也在不断下降。从细节上观察，训练次数在 0~10,000 次这个区间内时误差下降速度较快，一方面是因为此时训练刚刚开始，初始网络参数为随机种子的赋值，导致与最优解之间的距离较大，损失函数处于高梯度的陡峭区域，所以每次训练时参数更新的幅度也就较大，误差呈现快速下降的趋势；另一方面，PINN 的损失函数中包括初边值的约束，而初边值是点对点的简单拟合任务，所以训练初期会对这方面进行优先拟合，带动整体误差快速下降。而到 10,000 次训练以后，此时的预测解已经到最优解附近，损失函数的梯度会逐渐变小，所以每次参数更新的幅度也会逐渐变小，收敛速度就会变慢；同时初边值的拟合已经基本完成，重心逐渐转向偏微分方程的残差约束，在反向传播的过程中需要经过多层激活函数的二次求导过程，拟合的难度升高，所以误差下降速度也会降低。但是总体仍然保持稳定下降的趋势，直至逼近参考解。

从图 3 中可以看出，在黑色实线作为参考解的情况下，随着训练次数的增加，神经网络的预测解即各条虚线所示的曲线在不断与黑色实线重合。而从图 4 中可以进一步看出，随着训练次数的增加，最大误差的值在逐渐减小，全局误差也在变小，说明神经网络对于偏微分方程解的预测在逐渐趋于准确。

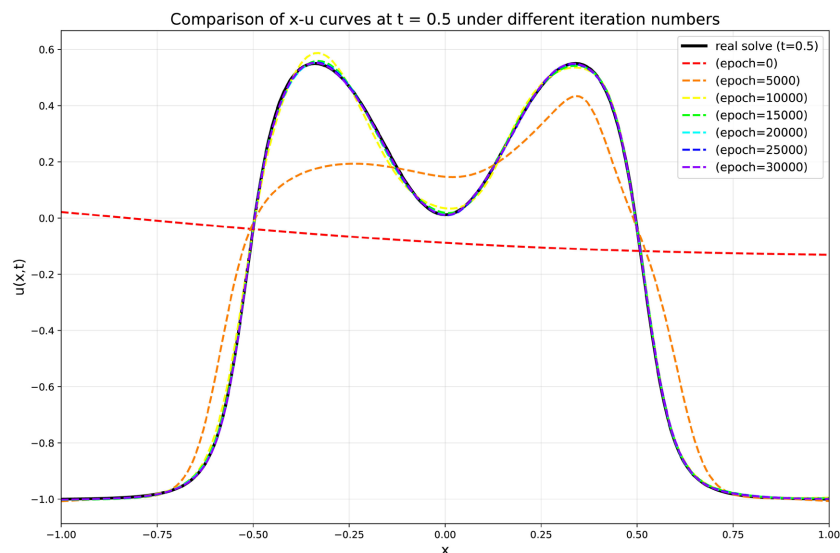


Figure 3. The x - u curve graph of the predicted and reference solutions as a function of training iterations

图 3. 随训练次数变化的预测解与固定参考解的 x - u 曲线图

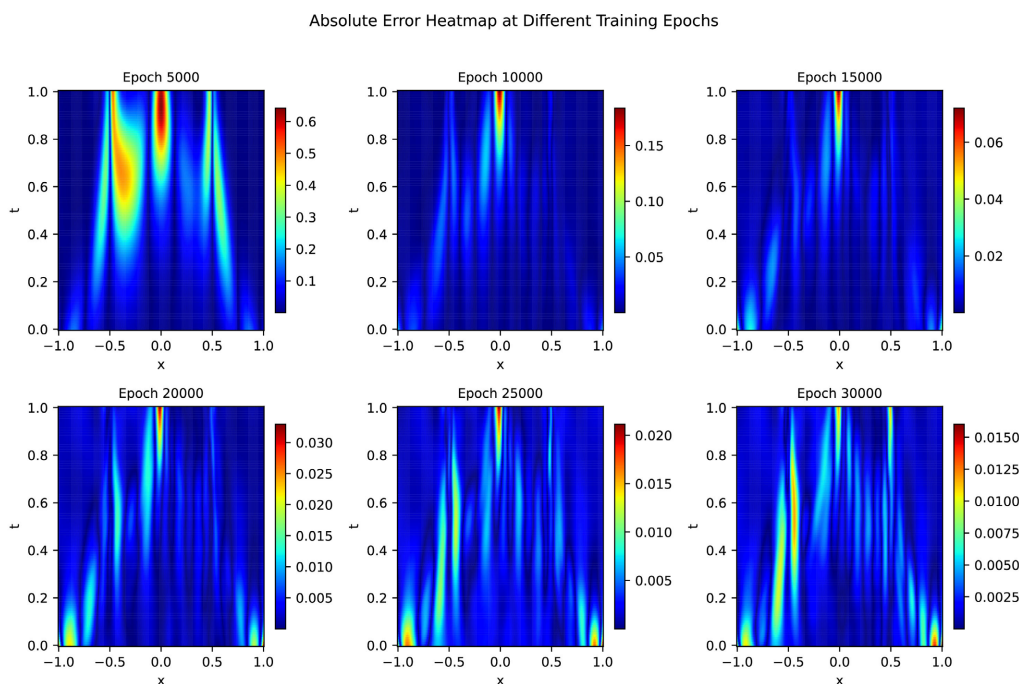


Figure 4. Heatmap of the error between predicted and reference solutions as a function of training iterations

图 4. 随训练次数变化的预测解与参考解的误差热力图

5.4. Allen-Cahn 方程扩散系数 D 的反演

基于上述对 Allen-Cahn 方程反问题的相关分析, 本节算例中以(7)式为研究模型, 采用物理信息神经网络对其中的扩散项系数 D 进行反演, 并研究训练次数和不同初始值对反演效果的影响。

本节中神经网络的优化器为 Adam, 神经网络学习率为默认值 0.001, 激活函数为 tanh, 隐藏层数为 3, 神经元个数为 64, 观测数据点数量设置为 100, 总训练次数设置为 80,000 epoch。

5.4.1. 训练次数对反演结果的影响

由上文的分析可知, 在求解 Allen-Cahn 方程正问题时, 随着训练次数的增加, 求解的效果也就越好。进一步, 为验证在参数反演的情况下是否依然有同样的结论, 本节探究在不同训练次数的影响下, 训练结果的准确性会有何变化。

下面的图表展示了随着训练次数的增加, 扩散系数 D 的反演情况及误差的变化趋势。本节算例中初始 D 的取值为 1.5。

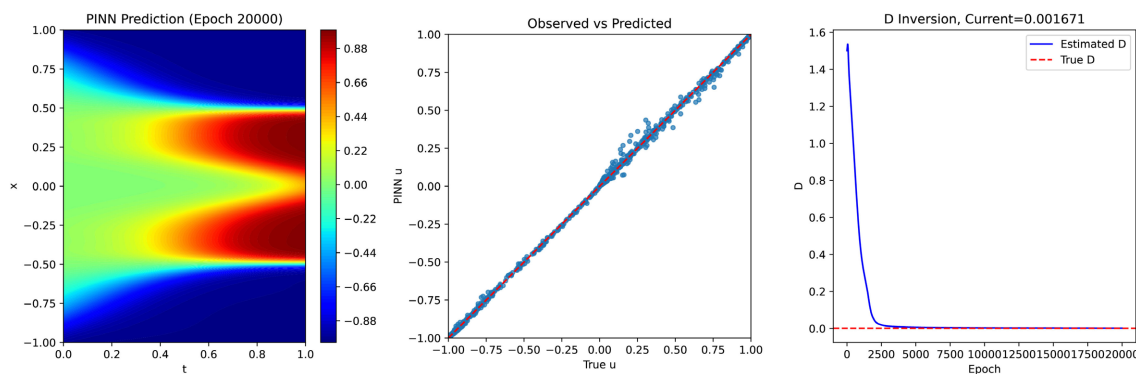


Figure 5., The diagram showing the inversion of the diffusion coefficient D when the epoch = 20,000

图 5. 训练次数为 20,000 时, 扩散系数 D 的反演情况图

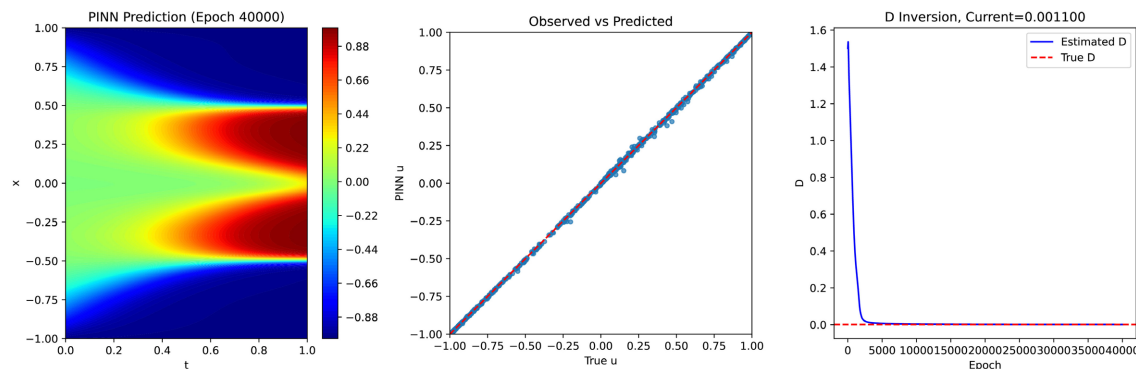


Figure 6. The diagram showing the inversion of the diffusion coefficient D when the epoch = 40,000

图 6. 训练次数为 40,000 时, 扩散系数 D 的反演情况图

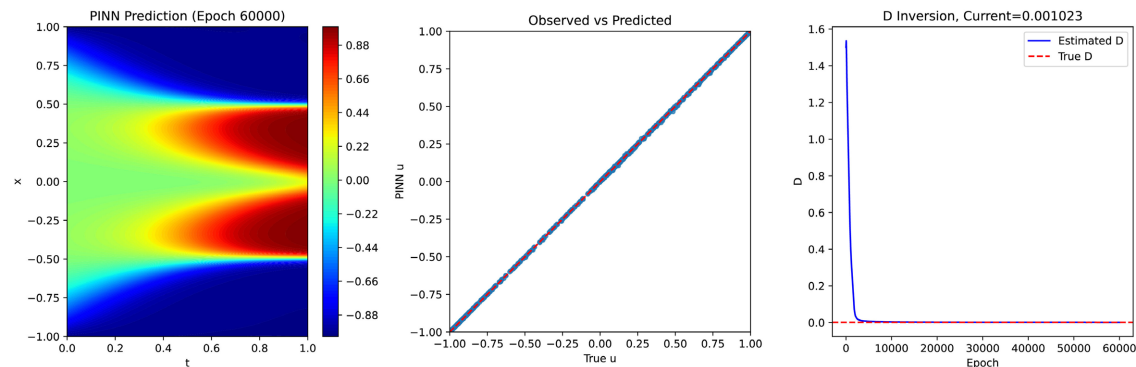


Figure 7. The diagram showing the inversion of the diffusion coefficient D when the epoch = 60,000

图 7. 训练次数为 60,000 时, 扩散系数 D 的反演情况图

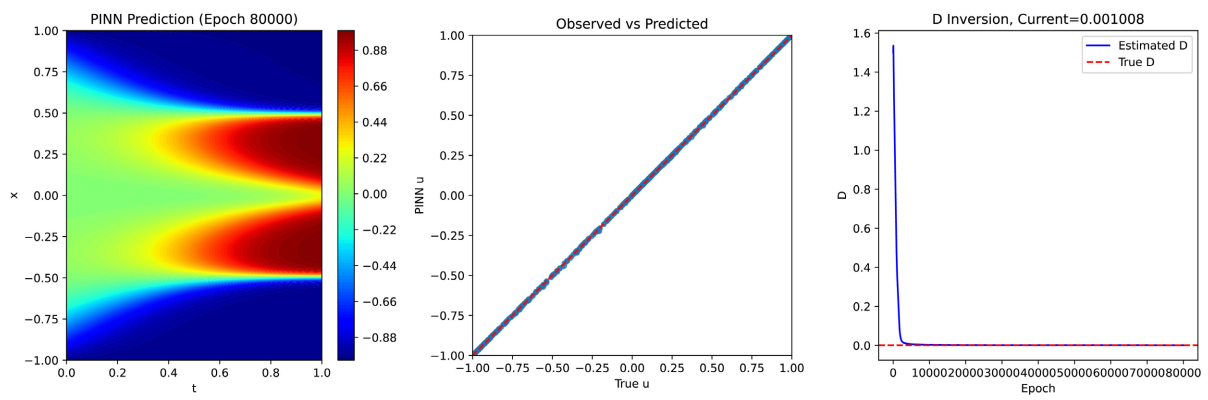


Figure 8. The diagram showing the inversion of the diffusion coefficient D when the epoch = 80,000
图 8. 训练次数为 80,000 时，扩散系数 D 的反演情况图

图 5~8 代表着随着迭代次数的增加，PINN 基于当前次数的 D 绘制的预测解图、观测数据点与预测点对比图及 D 的收敛曲线图。具体来看，这四张图的左图均代表着当前迭代次数下基于此时反演的 D 而绘制的时空解热力图。随着迭代次数的增加， D 逐渐逼近真实值，所以求解出的时空解也在逼近参考解。中间图反映了随训练次数的增加，观测数据点与基于对应时空坐标预测点的关系图。中间的观测 - 预测散点图以 $y=x$ 为标准值刻画了数据点拟合的过程。在训练初期，由于 D 值偏离真值程度较大，而且神经网络还尚未完成训练及拟合的过程，导致大部分的预测数据点均偏离观测点；随着训练次数的增加，神经网络逐渐学习到了方程的物理规律，所以观测数据点和预测值的误差在逐渐减小；进入训练后期，预测数据点基本与基准值重合，说明 PINN 已经基本学习到了方程的特性和规律，参数反演也基本趋于稳定。右图则反映了 D 随训练次数增加的收敛情况，可以看出：在前 10,000 轮次训练时，由于方程的梯度残差较大，所以优化器会快速更新 D 值， D 的值也就呈现快速下降的趋势。10,000 轮训练以后， D 已经逐渐接近真实值，此时方程残差的梯度逐渐变小，优化器进入精细化微调阶段， D 的值就会缓慢下降，直至接近真实值。

Table 5. The variation of diffusion coefficient D and relative error with the number of epoch
表 5. 扩散系数 D 与相对误差随训练次数的变化

训练次数(epoch)	初始 D 值	当前 D 值	相对误差
20,000	1.5	0.00167138	67.14%
40,000	1.5	0.00110013	10.01%
60,000	1.5	0.00102321	2.32%
80,000	1.5	0.00100772	0.77%

通过表 5 可以进一步看出，随着训练次数的增加， D 值在不断逼近真实值 0.001，相对误差也在逐步减小，最后可达到 10^{-2} 的量级。可见 PINN 在反演参数问题上具有收敛速度快、反演精度高的优势。

5.4.2. 验证 PINN 反演不依赖初值设定

进一步，为验证物理信息神经网络在参数反演时具有不依赖初始值选取的优势，即对初始值具有鲁棒性，本节随机选取 5 组在 $(0, 3]$ 范围内距真实值远近不同的初始 D 值进行反演，训练次数设置为 80,000 epoch。

Table 6. The inversion results of 5 groups of random D values within the range of $(0, 3]$ **表 6.** $(0, 3]$ 内 5 组随机 D 值反演情况

初始 D 值	最终反演 D 值	真实 D 值	相对误差
0.25	0.00101683	0.001	1.68%
0.67493	0.00101643	0.001	1.64%
1.42658	0.00100011	0.001	0.11%
1.5	0.00100772	0.001	0.77%
2.56127	0.00102488	0.001	2.48%

通过表 6 可以看出：这 5 组 D 与真实值的差距从百倍到两千倍左右，包含了距离真实值从近到远的不同数据。但无论初始扩散系数与真实值之间的偏离程度如何，最终都能拟合到真实值附近，误差均小于 2.5%，最小的甚至可以达到只有 0.11%。这表明物理信息神经网络对未知扩散系数的初始值表现出极强的鲁棒性，即其不依赖于初始值的选取。

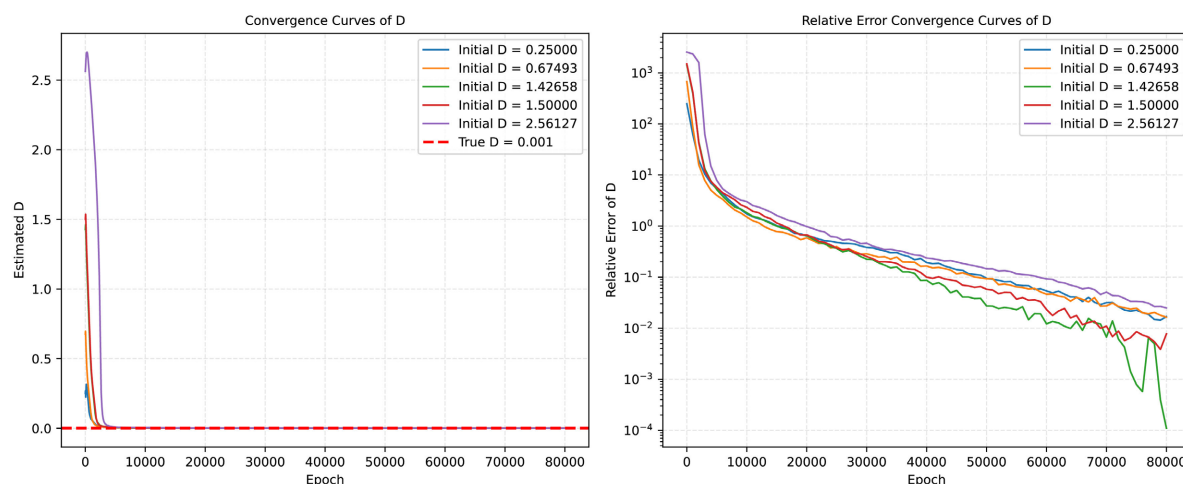
**Figure 9.** Convergence curves of the diffusion coefficient D and relative error under five initial values within the interval $(0, 3]$ **图 9.** $(0, 3]$ 内 5 组初始取值下扩散系数 D 的收敛曲线与相对误差收敛曲线

图 9 进一步展示了这 5 组远近不同 D 的收敛情况与误差的变化情况。可以看出这 5 组初值均可收敛至真实值。而且初值距离真实值越远，误差下降的速度也就越快， D 也更快速地收敛。这是因为 D 距离真实值越远，PDE 残差的梯度也就越大。而且，扩散系数 D 在 Allen-Cahn 方程中以线性乘子的形式参与训练，使得反问题的训练不存在局部最优的陷阱区域。此外，由于 PINN 在反问题的训练中拥有初边值约束、方程约束和数据约束的多方配合，使得梯度下降算法可以从任意初始值出发并收敛至最优解，进一步证明了 PINN 在参数反演时对初始值具有极强的鲁棒性。

6. 结论

本文聚焦 PINN 在偏微分方程中的应用，通过以 Allen-Cahn 方程为模型的数值算例，首先通过数值实验证明了以 $\tanh$ 作为激活函数的情况下方程的训练时长和训练误差为相对最优；其次通过对不同的网络尺寸进行研究，发现当隐藏层数为 3 且每层神经元的个数为 64 时，在 10,000 次训练后误差就可达到 10^{-2} 的量级；参数优化实验发现，学习率为 0.001 时是具有最小误差与最高效率的。此外，训练次数的设

置对最后的结果也有很大的影响。随着训练次数的增加,与基准解之间的误差也在随之缩小。另一方面,本文还对 Allen-Cahn 方程的反问题中未知参数反演的部分进行了研究,即通过已知的其他观测数据点和初边值条件去共同反演扩散项系数 D 。发现随着训练次数的增加,反演的结果在不断逼近真实值且最终能够实现精准拟合。而且在此场景下, PINN 对 D 的初始值表现出较强的鲁棒性,即对于 $(0, 3]$ 内的不同初始值均可实现高精度反演。综上所述,在求解 Allen-Cahn 方程这类非线性偏微分方程时, PINN 可以作为一种高精度且普适性强的求解方法进行相关正反问题的研究。

参考文献

- [1] Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019) Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational Physics*, **378**, 686-707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [2] Mishra, S. and Molinaro, R. (2020) Estimates on the Generalization Error of Physics Informed Neural Networks (PINNs) for Approximating PDEs. arXiv: 2006.16144.
- [3] Wang, S., Yu, X. and Perdikaris, P. (2022) When and Why PINNs Fail to Train: A Neural Tangent Kernel Perspective. *Journal of Computational Physics*, **449**, Article 110768. <https://doi.org/10.1016/j.jcp.2021.110768>
- [4] Zhao, A., Fattahi, D. and Hu, X. (2025) Physics-Informed Neural Networks for Physiological Signal Processing and Modeling: A Narrative Review. *Physiological Measurement*, **46**, 07TR02. <https://doi.org/10.1088/1361-6579/adf1d3>
- [5] 林灵, 高朋, 洪文兴. 物理信息神经网络在天空领域的应用与展望[J]. 系统工程与电子技术, 2026, 48(4): 1264-1282.
- [6] Chen, X. and Peng, W. (2025) PINN for Solving Forward and Inverse Problems Involving Integrable Two-Dimensional Nonlocal Equations. *Communications in Theoretical Physics*, **77**, Article 025002. <https://doi.org/10.1088/1572-9494/ad75f7>
- [7] Mishra, S. and Molinaro, R. (2021) Estimates on the Generalization Error of Physics-Informed Neural Networks for Approximating a Class of Inverse Problems for PDEs. *IMA Journal of Numerical Analysis*, **42**, 981-1022. <https://doi.org/10.1093/imanum/drab032>
- [8] Allen, S.M. and Cahn, J.W. (1979) A Microscopic Theory for Antiphase Boundary Motion and Its Application to Antiphase Domain Coarsening. *Acta Metallurgica*, **27**, 1085-1095. [https://doi.org/10.1016/0001-6160\(79\)90196-2](https://doi.org/10.1016/0001-6160(79)90196-2)
- [9] 王常健, 郑高峰. 抛物 Allen-Cahn 方程 Dirichlet 问题解的收敛性[J]. 数学物理学报(A 辑), 2025, 45(6): 1768-1790.
- [10] 乔寒月, 张鑫, 刘晓, 等. 一维 Allen-Cahn 方程紧差分格式的离散最大化原则和能量稳定性研究[J]. 应用数学学报, 2021, 44(1): 79-92.
- [11] 韦昌, 樊昱晨, 周永清, 等. 基于 Runge-Kutta 的自回归物理信息神经网络求解偏微分方程[J]. 力学学报, 2024, 56(8): 2482-2493.
- [12] Chen, P. and Jiang, B. (2024) Physics-Informed Neural Networks for Learning Allen-Cahn Equation Using Hard Constraint Representation. 2024 *China Automation Congress (CAC)*, Qingdao, 1-3 November 2024, 6433-6438. <https://doi.org/10.1109/cac63892.2024.10865427>
- [13] Das, I., Das, D., Bhattacharya, A., Debnath, P., Nath, S. and Chanda, M. (2024) Solution of Allen-Cahn Equation Using Physics-Informed Neural Networks. 2024 *IEEE International Conference of Electron Devices Society Kolkata Chapter (EDKCON)*, Kolkata, 30 November 2024-1 December 2024, 236-240. <https://doi.org/10.1109/edkcon62339.2024.10870453>