

武汉大学

本科毕业论文（设计）

基于网格的偏微分方程深度学习算法

姓 名：祝镪
学 号：2020305231078
专 业：信息与计算科学
学 院：数学与统计学院
指导教师：张晓平

二〇二四年四月

原创性声明

本人郑重声明：所呈交的论文（设计），是本人在指导教师的指导下，严格按照学校和学院有关规定完成的。除文中已经标明引用的内容外，本论文（设计）不包含任何其他个人或集体已发表及撰写的研究成果。对本论文（设计）做出贡献的个人和集体，均已在文中以明确方式标明。本人承诺在论文（设计）工作过程中没有伪造数据等行为。若在本论文（设计）中有侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

作者签名：祝强 指导教师签名：
日期：2024 年 5 月 12 日

张晓平

版权使用授权书

本人完全了解武汉大学有权保留并向有关部门或机构送交本论文（设计）的复印件和电子版，允许本论文（设计）被查阅和借阅。本人授权武汉大学将本论文的全部或部分内容编入有关数据进行检索和传播，可以采用影印、缩印或扫描等复制手段保存和汇编本论文（设计）。

作者签名：祝强 指导教师签名：
日期：2024 年 5 月 12 日

张晓平

摘 要

近年来, 基于网格的 PDE 深度学习算法引起了国内外学者广泛关注。国外学者 Raissi 等人于 2017 年提出物理信息神经网络 (PINNs), 将深度学习与 PDE 求解相结合, 并取得了一系列成果。国内学者也积极开展相关研究, 在 PDE 求解的精度、效率和鲁棒性方面取得了进展。目前, 基于网格的 PDE 深度学习算法尚处于发展初期, 主要存在以下挑战: 如何提高 PDE 求解的精度; 如何提高计算效率; 如何增强模型的鲁棒性。本课题将研究基于网格的 PDE 深度学习算法, 重点研究双线性插值和内嵌物理神经网络两种方法的结合、PDE 求解的精度、效率和鲁棒性提升以及复杂 PDE 问题的求解。

关键词: 内嵌物理神经网络; 双线性插值; 基于网格

ABSTRACT

In recent years, grid-based PDE deep learning algorithms have attracted wide attention from domestic and foreign scholars. Foreign scholars, such as Raissi et al., proposed the Physics-Informed Neural Networks (PINNs) in 2017, which combine deep learning with PDE solving and have achieved a series of results. Domestic scholars have also actively carried out related research, making progress in the accuracy, efficiency, and robustness of PDE solving.

Currently, grid-based PDE deep learning algorithms are still in the early stages of development, facing the following challenges: 1. How to improve the accuracy of PDE solving; 2. How to improve computational efficiency; 3. How to enhance the robustness of the model. This project will study grid-based PDE deep learning algorithms, focusing on the combination of bilinear interpolation and embedded physics-informed neural networks, improving the accuracy, efficiency, and robustness of PDE solving, and solving complex PDE problems.

Keywords: PINN; bilinear-interpolation; Grid-based

目 录

摘要	I
ABSTRACT	II
1 绪论	1
1.1 概述	1
1.1.1 问题描述	2
1.1.2 损失函数的构建	3
1.1.3 逆问题	4
1.1.4 神经网络的优化器选择	6
1.2 基于网格的 PINN	8
1.2.1 导论	8
1.2.2 双线性插值	9
1.2.3 多尺度网格	10
2 实例与结果	12
2.1 Burgers 方程	12
2.2 Allen-Cahn 方程	13
2.3 一维对流方程	14
2.4 二维椭圆偏微分方程	15
3 结论	17
参考文献	18

1 绪论

1.1 概述

内嵌物理神经网络（PINN）是一种用于求解偏微分方程（PDE）的机器学习方法[1]。它结合了神经网络的灵活性和偏微分方程的物理约束条件，从而能够有效地学习系统的行为并且对其进行预测，并用于解决涉及复杂物理过程的偏微分方程（PDE）建模和求解问题。

传统的数值方法通常需要离散化空间和时间，然后使用有限元、有限差分或有限体积等技术求解偏微分方程。然而，这些方法在处理高维、多尺度、非线性和不规则问题时往往面临着巨大的挑战。PINN的出现旨在克服这些挑战，提供一种更加灵活和高效的方法来处理这些复杂的物理问题。相比之下，PINN不需要离散化空间和时间，而是直接从数据中学习系统的行为和物理规律。PINN的核心思想是将物理方程嵌入到神经网络的训练过程中，优化神经网络的参数来最小化PDE的残差，同时满足物理边界条件，以此来获得物理问题的近似解。具体来说，PINN通过在神经网络的损失函数中包含物理方程的残差项来实现这一目标。这种残差项可以看作是物理方程在神经网络参数下的预测值与真实值之间的差异，通过最小化这个残差项，可以使神经网络逐渐逼近物理方程的解[2]。

这种方法可以通过监督学习框架来实现，其中损失函数由PDE的残差项和边界条件的残差项组成。通过反向传播算法，可以有效地更新神经网络的参数，使得残差逐渐减小，从而逼近真实的物理解。与传统的数值方法相比，PINN不需要对问题域进行离散化，因此可以更好地处理复杂的几何形状和边界条件，并节省大量计算资源和时间。

PINN在多个领域都取得了显著的成就，例如流体力学、固体力学、声学 and 电磁学等[3]。在流体力学中，PINN被广泛应用于模拟流体流动和热传导等问题，其高精度和高效率使其成为研究热传导、湍流和多相流等复杂现象的有力工具。在固体力学中，PINN可以用于预测结构件的变形和应力分布，为工程设计提供重要参考。在声学 and 电磁学中，PINN可以用于模拟声波和电磁波的传播和散射过程，为声学成像和无损检测等应用提供支持。PINN作为一种融合了深度学习和物理建模的新方法，为解决复杂物理问题提供了一种全新的思路 and 工具。随

着研究的不断深入，相信PINN将会在更多领域展现出其强大的潜力，并为解决实际工程和科学问题提供更加高效和可靠的解决方案。

然而，尽管PINN在处理物理问题上取得了一定的成功，但在实际应用中仍然面临一些困难和不足之处。首先，PINN的训练过程通常需要大量的数据和计算资源。由于物理方程的复杂性和高维性，需要大规模的数据来训练神经网络，这在某些情况下可能会成为一个挑战。此外，PINN的训练过程通常需要进行大量的迭代和调试，这对于一些非专业领域的研究者可能会造成一定的困扰。

其次，PINN在处理非线性和多尺度问题时可能会遇到困难[4]。虽然PINN可以很好地处理一些线性问题和简单的非线性问题，但对于涉及复杂非线性和多尺度现象的问题，PINN的效果可能会受到限制[5]。这主要是因为神经网络在表示复杂函数时存在一定的局限性，需要更加深入和有效的网络结构和训练方法来克服这些问题。

此外，PINN的泛化能力和鲁棒性也是一个需要关注的问题。由于神经网络的特性，PINN在训练数据范围之外的区域可能会表现不佳，这对于一些需要在整个问题域上进行预测的应用来说可能是一个不足之处。为了提高PINN的泛化能力和鲁棒性，需要进一步研究和改进网络结构、训练方法和正则化技术[6]。总的来说，尽管PINN在物理建模和求解方面具有很大的潜力，但在实际应用中仍然面临着一些挑战 and 不足之处。

因此，本文将讨论和探索一种改进的PINN方法，通过结合传统数值方法和神经网络的优势解决PINN目前面临的挑战，目的集中在提高PINN的训练效率、处理非线性问题、提高泛化能力和鲁棒性等方面，以进一步发展和完善这一方法，使其能够更好地应用于实际工程和科学问题的求解中。

1.1.1 问题描述

PINN主要用于解决偏微分方程中的初边值问题，一般形式如下：

$$\begin{aligned} F\left(x, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial t^2}, \dots\right) &= 0, x \in \Omega, t \in [0, T] \\ G\left(x, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial t^2}, \dots\right) &= g(x, t), x \in \partial\Omega \\ u(x, 0) &= u_0(x), x \in \Omega \end{aligned} \quad (1)$$

其中， $u(x, 0) = u_0(x), x \in \Omega$ 为初值条件，而 $G\left(x, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial t^2}, \dots\right) = g(x, t), x \in \partial\Omega$ 代表偏微分方程的边界条件。偏微分方程通常描述了自然界中的物理现象，方程式 $F\left(x, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial t^2}, \dots\right) = 0$ 蕴含了物理定律。例如，波动方程描述了波动的传播，热传导方程描述了热量的传导，流体力学方程描述了流体运动等。为了在物理神经网络（PINN）中实现对物理定律的嵌入，我们需要构建神经网络的损失函数，并在中包含物理方程的残差项。

1.1.2 损失函数的构建

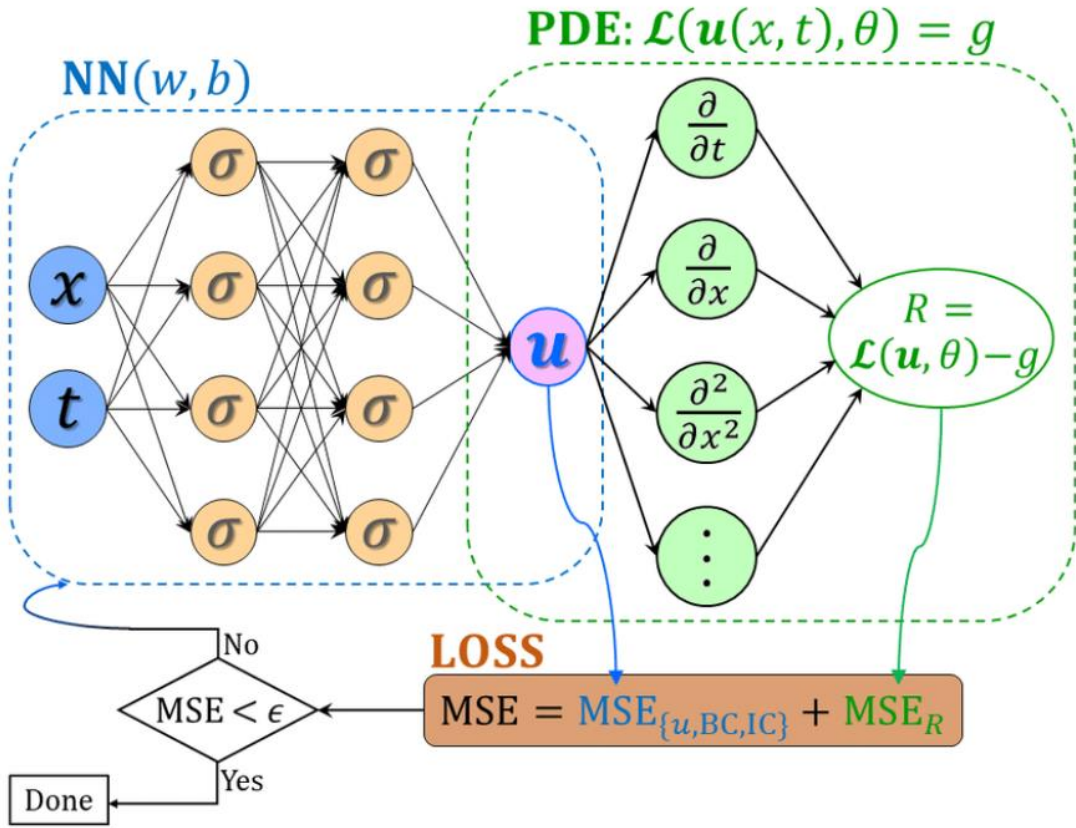


图1 PINN

实际上，我们可以通过神经网络模拟任何函数 $NN(x, t) \approx y(x, t)$ ，并且把其看作一个特殊的函数

$$NN(X) = W_n \sigma_{n-1}(W_{n-1} \sigma_{n-2}(\dots(W_2(W_1 X + b_1) + b_2) + \dots) + b_{n-1}) + b_n$$

我们以这样一个扩散方程为例 $\frac{\partial y}{\partial t} = \frac{\partial^2 y}{\partial x^2} - f$

由于NN是一个函数，我们可以通过自动求导得到它的各阶偏导 $\frac{\partial NN}{\partial t}, \frac{\partial^2 NN}{\partial x^2}$ 。

我们假定 $NN(t, x) \approx y(t, x)$

那么

$$\left(\frac{\partial NN}{\partial t} - \frac{\partial^2 NN}{\partial x^2}\right) + f \approx \left(\frac{\partial y}{\partial t} - \frac{\partial^2 y}{\partial x^2}\right) + f = 0 \quad (2)$$

即

$$\left(\frac{\partial NN}{\partial t} - \frac{\partial^2 NN}{\partial x^2}\right) + f \approx 0 \quad (3)$$

我们把这个函数定义为 F ，也即

$$F(t, x) = \left(\frac{\partial NN}{\partial t} - \frac{\partial^2 NN}{\partial x^2}\right) + f \quad (4)$$

如果 NN 是方程解的逼近，那么就有 $F \rightarrow 0$

通过在定义域内部采样，我们可以得到残差

$$MSE_F = \frac{1}{N_F} \sum_{i=1}^{N_F} |F(t_f^i, x_f^i)|^2 \quad (5)$$

通常情况下，测试集是指我们已经知道答案的一组数据。

本例子中，由于边界条件是狄利克雷条件，我们可以在定义域边界采样进行监督学习得到边界条件的残差项。

$$MSE_u = \frac{1}{N_u} \sum_{i=1}^{N_u} |y(t_u^i, x_u^i) - NN(t_u^i, x_u^i)|^2 \quad (6)$$

MSE_u 是有关数据的误差，通过训练减小与物理定律的偏离，并防止过拟合。另一方面 MSE_F 是与偏微分方程蕴含的物理定律相关的误差，通过训练减小以确保这样的神经网络在给定未知输入时更好地泛化。

由此我们导出了损失函数

$$MSE = MSE_u + MSE_f \quad (7)$$

通过最小化损失函数，我们可以得到训练完成的神经网络，也即偏微分方程真实解的一个逼近。

1.1.3 逆问题

偏微分方程（PDE）逆问题是指从已知的观测数据反推未知的模型参数或初始条件的问题。通常情况下，我们可以通过解正向问题（已知模型参数或初始

条件，求解PDE得到的解）来获得观测数据，然后利用逆问题的方法来确定模型参数或初始条件。逆问题在科学与工程领域中具有广泛的应用，例如地球物理学中的地震波反演、医学成像中的逆问题重建等。

对于科学与工程领域而言，PDE逆问题的解决对于理解和预测自然现象具有重要意义。通过逆问题求解，我们可以更好地理解模型参数对于系统行为的影响，从而指导实际问题的决策和优化。例如，在医学成像中，逆问题的解决可以帮助医生更准确地诊断疾病；在地球物理学中，逆问题的解决可以帮助科学家更好地理解地球内部结构和地壳运动规律。因此，通过PINN求解PDE逆问题，不仅可以提高问题求解的效率和精度，还可以推动科学研究和工程应用的进步。

在逆问题中，我们通常希望通过已知的边界条件或者观测数据来推断物理系统的未知参数或者边界条件。PINN的引入为这一类问题提供了一种全新的解决思路。使用内嵌物理神经网络（PINN）来求解偏微分方程（PDE）的逆问题是一种新颖而具有挑战性的方法。

内嵌物理神经网络（PINN）在解决PDE逆问题时具有独特的优势。传统的逆问题求解方法往往需要假设模型的参数形式或者采用迭代优化的方法，如最小二乘法、梯度下降法等，来寻找最优的参数或者边界条件。然而，这些方法通常需要手工设计损失函数，并且可能会受到局部最优解的影响。而PINN则可以通过神经网络灵活地拟合模型参数的形式，将物理方程嵌入到神经网络的损失函数中，可以直接在网络的训练过程中获得物理系统的解，从而避免了对参数形式的假设以及手工设计损失函数的复杂性。此外，PINN还可以结合物理约束条件，比如保证逆问题求解的解满足PDE，从而提高了逆问题求解的准确性和鲁棒性。具体来说，对于PDE的逆问题，我们可以将观测数据与物理方程的解联系起来，构建一个包含观测数据残差和物理方程残差的综合损失函数。通过最小化这个综合损失函数，可以使神经网络逐渐逼近物理系统的真实解，从而实现逆问题的求解。

对于参数化和非线性偏微分方程，具有一般形式

$$u_t + \mathcal{N}[u; \lambda] = 0 \quad (8)$$

类似于原问题，假设 $N(t, x) \approx u(t, x)$ ，令 $f(t, x) = N_t + \mathcal{N}[N, \lambda]$ ，我们可以导出损失函数。

$$MSE_f = \frac{1}{N_u} \sum_{i=1}^{N_u} |f(t_u^i, x_u^i)|^2 \quad (9)$$

$$MSE_u = \frac{1}{N_u} \sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - N(t_u^i, x_u^i)|^2 \quad (10)$$

$$MSE = MSE_u + MSE_f \quad (11)$$

对其进行最小化即可得到参数 λ 的近似。与传统方法相比，PINN具有更好的灵活性和适应性，可以更好地处理复杂的逆问题，并且可以在数据不完整或者噪声较大的情况下取得更好的效果。

1.1.4 神经网络的优化器选择

基于梯度的优化算法用于优化损失函数，L-BFGS 和 Adam 广泛应用于 PINN。自动微分用于计算两个微分算子关于输入坐标的梯度，以及关于可训练神经网络的损失函数参数。

其中，L-BFGS优化器是一种经典的基于梯度的优化算法，被广泛应用于解决无约束优化问题。L-BFGS优化器是对BFGS算法的改进，通过限制存储的历史梯度信息来降低存储开销，适用于大规模问题和高维参数空间。L-BFGS算法以其高效性和稳定性在机器学习、数值优化和科学计算等领域得到了广泛应用。

L-BFGS优化器的基本原理是利用近似表示目标函数的Hessian矩阵的逆来实现对参数的更新。与BFGS算法不同的是，L-BFGS算法只存储和更新最近的一些历史梯度信息，而不是完整的Hessian矩阵。这样可以降低存储开销，并且能够在迭代过程中动态地调整近似Hessian矩阵的逆，从而适应不同的问题和参数空间。

L-BFGS算法的主要步骤包括：首先初始化参数和初始近似Hessian矩阵的逆；然后计算梯度，并利用前几步的历史信息构建一个近似Hessian矩阵的逆；接着更新参数，并重复这个过程直到满足停止准则。由于L-BFGS算法只需要存储和更新少量历史信息，因此具有较低的存储需求和计算复杂度，特别适用于大规模问题和高维参数空间的优化。

在实际应用中，L-BFGS优化器通常与其他优化技术结合使用，如线搜索和正则化技术，以提高优化的效果和稳定性。L-BFGS算法在解决一些复杂的优化

问题中表现出色，如神经网络训练、图像处理和信号处理等领域，为这些领域的研究和应用提供了重要的优化工具。

总的来说，L-BFGS优化器作为一种高效和稳定的优化算法，在解决无约束优化问题中具有重要的应用价值。随着研究的不断深入，相信L-BFGS算法将继续发挥重要作用，并为优化领域的发展和应用带来更多的创新和突破。

另一方面，Adam优化器是一种结合了动量和自适应学习率的优化算法，广泛应用于训练神经网络和解决机器学习问题。Adam优化器结合了AdaGrad和RMSProp算法的优点，具有较快的收敛速度和良好的泛化性能，在深度学习和优化领域取得了显著的成就。

Adam优化器的基本原理是根据每个参数的梯度情况动态地调整学习率和动量，从而加快收敛速度并提高训练效果。具体来说，Adam优化器维护两个移动平均值：一个是梯度的指数加权移动平均，用于估计梯度的均值；另一个是梯度平方的指数加权移动平均，用于估计梯度的方差。通过这两个移动平均值，Adam优化器可以根据每个参数的梯度情况来动态地调整学习率和动量，从而更好地适应不同的参数和问题。

Adam优化器具有自适应学习率，即Adam优化器可以根据每个参数的梯度情况动态地调整学习率，从而在训练过程中保持较好的学习效果。Adam优化器结合了动量的概念，可以在训练过程中积累梯度的历史信息，从而加速收敛速度。Adam优化器对超参数的选择相对不敏感，通常可以在不同的问题和数据集上取得较好的效果。Adam优化器已被广泛应用于深度学习和机器学习领域，取得了许多重要的研究成果和实际应用。Adam优化器作为一种高效和稳健的优化算法，在深度学习和机器学习领域具有重要的应用价值。随着研究的不断深入，相信Adam优化器将继续发挥重要作用，并为优化领域的发展和应用带来更多的创新和突破。

总的来说，Adams优化器适用于大规模的神经网络训练，而L-BFGS更适用于小规模问题或者需要更精确解的问题。在本文中，我们使用 Adam 优化器运行 2000 次迭代，然后切换到 L-BFGS 优化器直到收敛，因为普遍认为Adam 优化器的迭代速度更快，而L-BFGS优化器精度更高。

1.2 基于网格的PINN

1.2.1 导论

传统的PINN具有缺点，一是训练的速度慢，占用计算资源大，二是精度不够，显然采样越多精度越高但同时也会影响模型的训练速度。

为此我们提出了基于网格的PINN。通过架起传统数值方法与神经网络方法的桥梁，可能会拿到传统方法的一些好处，让模型训练更快。

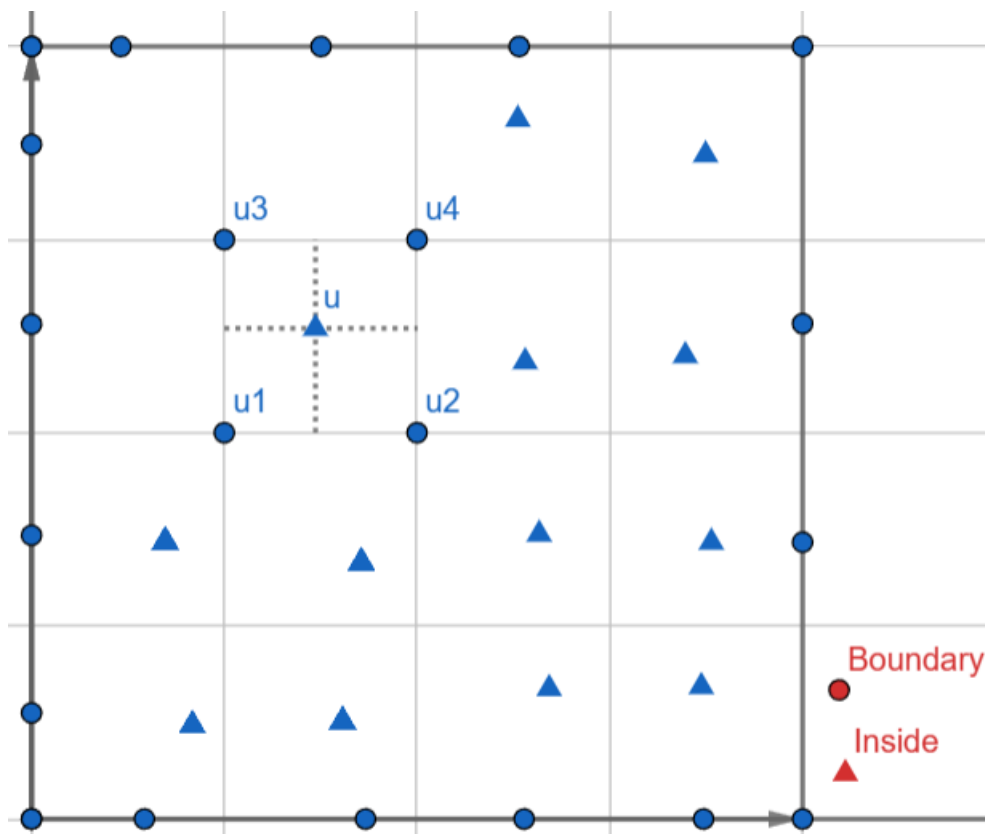


图2 基于网格的PINN

如上图所示，我们依然采取传统物理神经网络的采样方式，不同的是数据点的特征会定义在网格节点。比如上图的点 u ，在计算该点的函数值时，我们先找出 u 处于哪一个小网格中，然后根据四个顶点作插值（例如双线性插值）。这样我们就得到了点 u 的函数信息，导数信息也是同理。

显然网格的稀疏程度会对模型的训练速度和精度产生影响。可以猜测，网格较为稀疏时误差较大，需要选取合适的超参数。

1.2.2 双线性插值

假如我们想得到未知函数 f 在点 $P = (x, y)$ 的值，假设我们已知函数 f 在 $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, $Q_{22} = (x_2, y_2)$ 四个点的值。

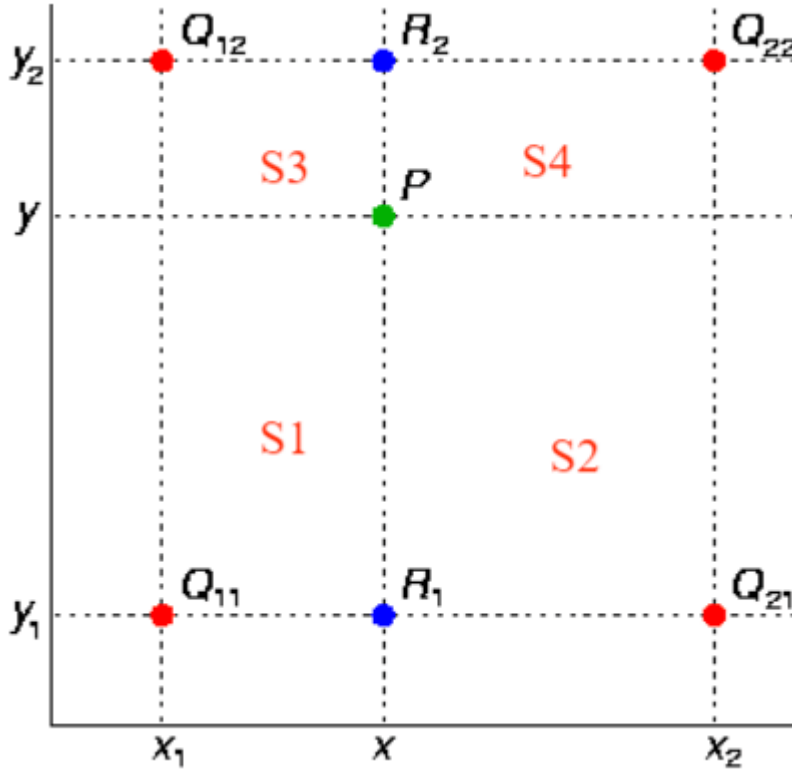


图3 双线性插值

首先在 x 方向进行线性插值，得到

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \text{ Where } R_1 = (x, y_1),$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \text{ Where } R_2 = (x, y_2).$$

然后在 y 方向进行线性插值，得到

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (12)$$

这样就得到所要的结果

$$\begin{aligned} f(x, y) \approx & \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) \\ & + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1) \end{aligned} \quad (13)$$

显然我们可以采用别的插值方法来从网格节点计算得到内部数据点的信息。双线性插值是较为简单方便的插值方法。

实际上，对于双线性插值，我们有

$$f(x, y) \approx f(Q_{11})k\left(\frac{x_2 - x}{x_2 - x_1}\right)k\left(\frac{y_2 - y}{y_2 - y_1}\right) + f(Q_{21})k\left(\frac{x - x_1}{x_2 - x_1}\right)k\left(\frac{y_2 - y}{y_2 - y_1}\right) \\ + f(Q_{12})k\left(\frac{x_2 - x}{x_2 - x_1}\right)k\left(\frac{y - y_1}{y_2 - y_1}\right) + f(Q_{22})k\left(\frac{x - x_1}{x_2 - x_1}\right)k\left(\frac{y - y_1}{y_2 - y_1}\right) \quad (14)$$

其中 $k(x) = x$ 。

我们把 $k(x)$ 称为核函数，为了支持高阶偏导数，需要修改核函数。显然这个函数必须是一个定义域和值域都在 $[0,1]$ 的单调递增光滑函数，并且 $k(x)$ 必须满足 $k(0) = 0, k(1) = 1$ 。

可以推测 $k(x)$ 关于点 $(0.5, 0.5)$ 中心对称时计算复杂度与求解精度最佳。因为 $k(x)$ 实际上映射了数据点到相邻网格点的垂直距离和水平距离，该条件蕴含了对称性。

本文考虑了令 $k(x) = \frac{1}{2}(1 - \cos(\pi x))$ ，因为该函数显然无穷阶可导，几乎在所有偏微分方程当中都具有泛用性。还可以考虑尝试 $k(x) = 3x^2 - 2x^3$ 等其他函数，但由于该函数只是三阶可导，同样和一次函数一样具有局限性。

1.2.3 多尺度网格

可以预见的是，由于我们引入的网格表示法中，每个网格点只覆盖输入域的小块子区域，因此网格分辨率越高，收敛速度越快。网格越密，PINN的收敛速度就越快，解也就越精确。

然而随着网格变密，所需的训练数据点数量可能会以指数级增长。给定随机采样的定位点，网格越细化网格越细，网格单元在训练过程中覆盖这些点的机会就越少。这将导致因为在只有几个数据点的小局部区域内找到满意的解决方案，很容易陷入局部极小值，也就是偏微分方程的残差项部分的损失函数会非常小，但是最后却会产生错误的解。

改变插值方式或许可以解决这一问题，例如用三次插值代替双线性插值，但是可能会大幅提高计算成本，因为在插值时需要查找更多的相邻网格点。

由于这些原因，我们将多个稀疏的网格堆叠起来，通过将所有网格相加来计算每个定位点的值，并称之为多尺度网格。

我们以一个二维单层网格为例。 H 和 W 分别代表网格在空间和时间维度上的尺寸大小。 \hat{x} 和 \hat{t} 代表每个数据点标准化后的坐标，也就是说 $\hat{x} \in [1, H]$ ， $\hat{t} \in [1, W]$ 。我们将网格看作一个特征提取，定义为 $\phi(\hat{x}, \hat{t}, f)$ ，其中 $f \in R^{c \times H \times W}$ 表示定义在网格点的所有参数构成的张量， $f_{ij} \in R^c$ 表示在特定网格点 (i, j) 的可训练参数向量， c 为通道数。举例来说，如果 $H \subset R, W \subset R$ ，也即输入域为二维空间，那么此时 f 为三维张量，超参数 c 取为4是较为合理的。一般来说，若输入域为 n 维空间，那么通道数 c 应取为 2^n ，原因在于此时每个网格点上的参数由周围 2^n 个小区域中的数据确定，若 $c < 2^n$ ，显然极易发生欠拟合。

根据上一节1.2.2的结论，我们有

$$\phi(\hat{x}, \hat{t}, f) = \sum_{i=1}^H k(\max(0, 1 - |\hat{x} - i|)) \sum_{j=1}^W k(\max(0, 1 - |\hat{t} - j|)) f_{ij} \quad (15)$$

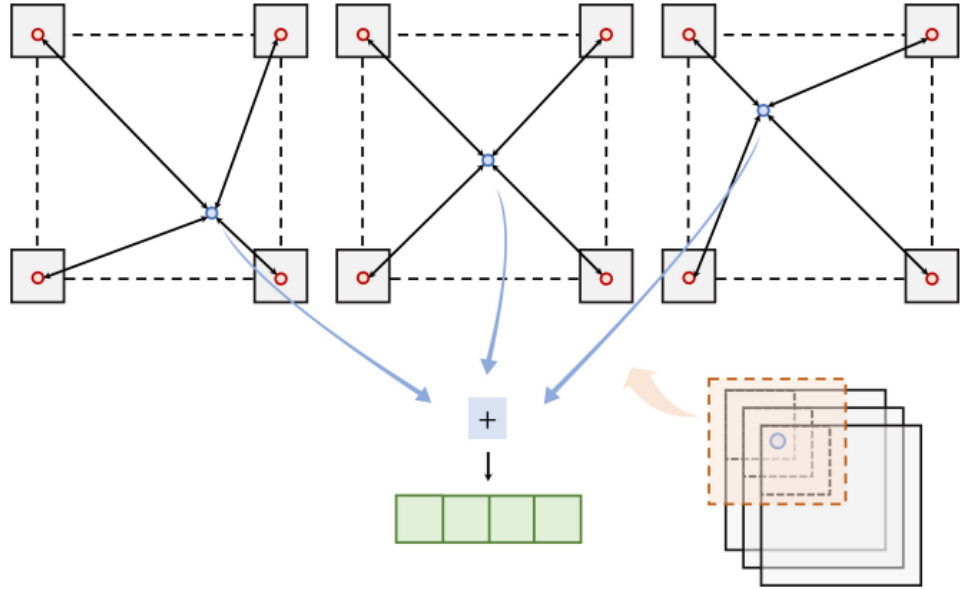


图4 多尺度网格

多尺度网格方法中，我们有 M 个网格，在每个不同的网格当中对给定的数据点进行偏移，使得输入坐标可以位于每个网格中的不同位置。这样，我们可以将有效网格大小增加 M 倍，这意味着神经网络的泛化能力也会增加。如果不进行

偏移操作，那么输入坐标每次都位于相同的位置，网格就会输出相似的值，最后导致过拟合。

我们将多尺度网格定义为 $\phi_M(\hat{x}, \hat{t}, f)$ ，有

$$\phi_M(\hat{x}, \hat{t}, f) = \sum_{i=1}^M \phi\left(\hat{x} + \frac{(i-1)}{M}, \hat{t} + \frac{(i-1)}{M}, f^i\right) \quad (16)$$

2 实例与结果

2.1 Burgers 方程

$$\begin{aligned} u_t + uu_x - \nu u_{xx} &= 0, \nu = \frac{1}{100\pi} \\ u(0, x) &= -\sin(\pi x) \\ u(-1, t) &= u(1, t) = 0 \\ x &\in [-1, 1], t \in [0, 1] \end{aligned} \quad (17)$$

使用内嵌物理神经网络（PINN）求解 Burgers 方程是一种有效的方法。Burgers 方程是一种非线性偏微分方程，描述了具有粘性和非线性特性的流体流动问题[7]。PINN 可以通过将 Burgers 方程嵌入到神经网络的损失函数中来实现对其解的逼近，从而实现对流体流动问题的建模和求解。PINN 的优势在于不需要对问题域进行离散化，可以直接在连续域上求解，从而可以更好地处理复杂的流体流动问题。通过使用 PINN，可以更快地获得流体流动的解，并且可以在不同边界条件和初值条件下进行灵活的求解和模拟，为流体力学研究和工程应用提供了一种新的方法

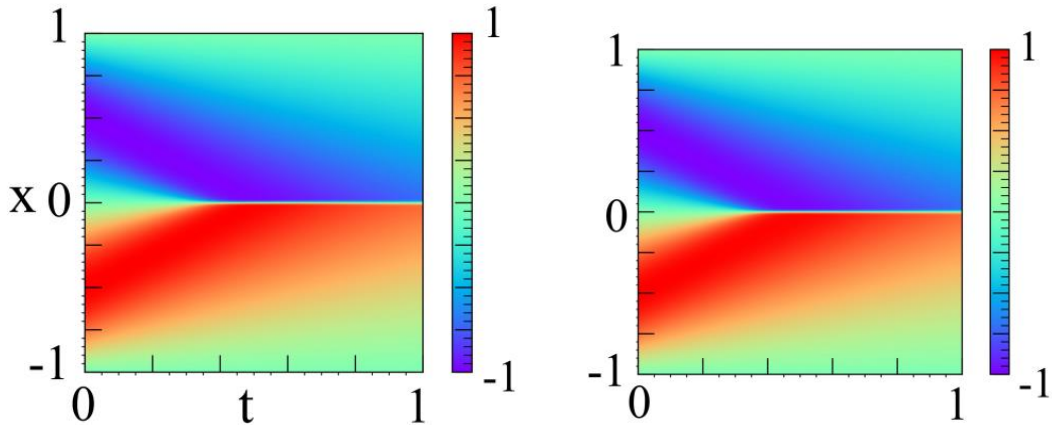


图 5 传统的 PINN 的预测解(左)和基于网格的 PINN 的预测解(右)

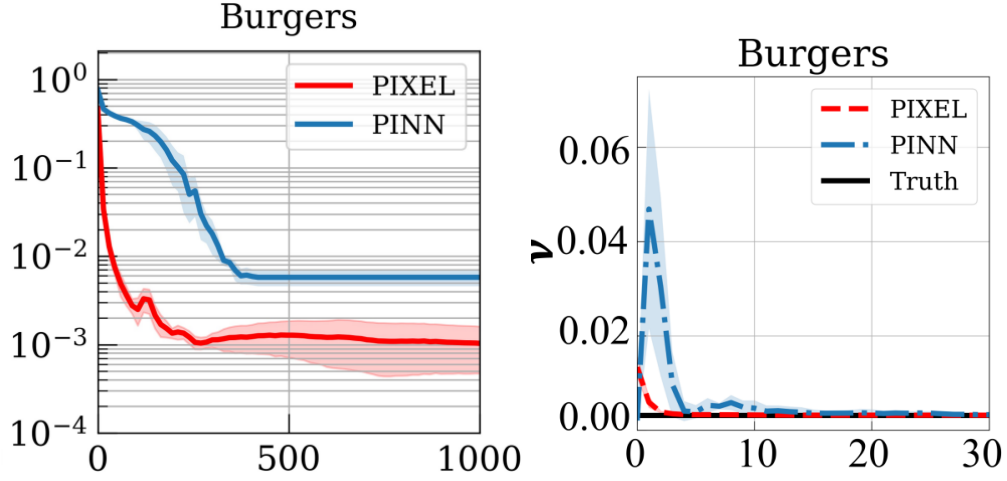


图 6 两种 PINN 在原问题(左)和逆问题(右)收敛速度的比较

Burgers 方程的解在 $x = 0$ 具有奇异行为，实验结果表明，虽然两个预测解并无明显差距，但是基于网格的 PINN 收敛速度明显快于传统的 PINN，并且精度高出一个数量级。在逆问题上，传统的 PINN 鲁棒性较差，测试集不同时波动很大，并且在最初几次迭代中参数值 ν 甚至偏离真实值。

2.2 Allen-Cahn 方程

$$\begin{aligned}
 u_t - 0.0001u_{xx} + \lambda u^3 - 5u &= 0 \\
 x \in [-1, 1], t \in [0, 1], \lambda &= 5 \\
 u(x, 0) &= x^2 \cos(\pi x) \\
 u(-1, t) &= u(1, t) \\
 u_x(-1, t) &= u_x(1, t)
 \end{aligned} \tag{18}$$

Allen-Cahn 方程是描述相变过程中界面演化的偏微分方程，广泛应用于材料科学等领域。PINN 可以通过将 Allen-Cahn 方程嵌入到神经网络的损失函数中来实现对其解的逼近，从而实现对相变过程的建模和求解。与传统的数值方法相比，PINN 不需要对空间离散化，可以直接在连续域上求解，从而可以更好地捕捉相变界面的演化。通过使用 PINN，可以更准确地模拟和预测相变过程中界面的形态和演化规律，为材料设计和相变研究提供了一种新的方法

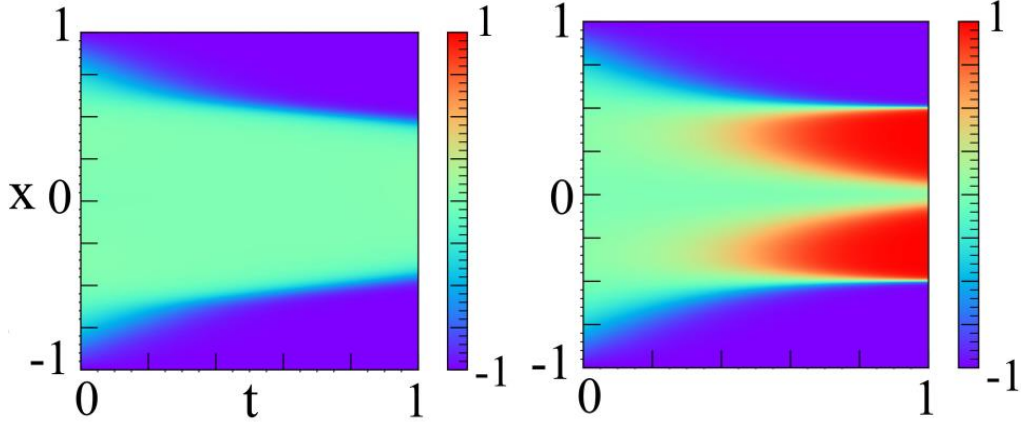


图 7 传统的 PINN 的预测解(左)和基于网格的 PINN 的预测解(右)

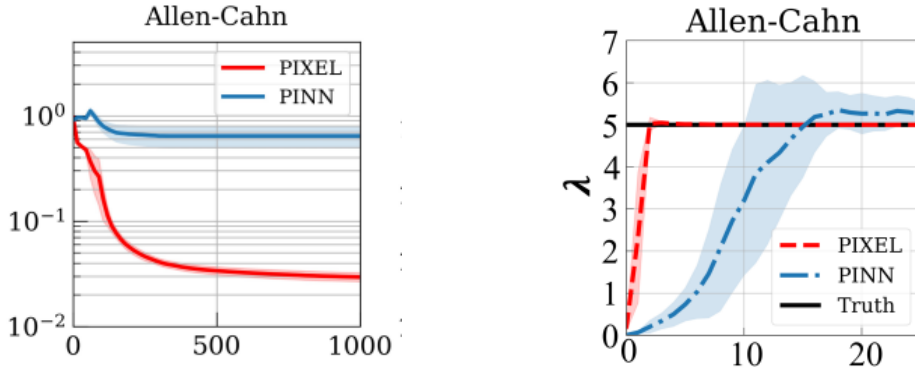


图 8 两种 PINN 在原问题(左)和逆问题(右)收敛速度的比较

实验结果表明，传统的 PINN 的预测解几乎完全没有反映出真实解，甚至在上万次迭代后误差仍然在 10^{-1} 数量级，另一方面，基于网格的 PINN 很好地模拟了方程的精确解。在逆问题上，传统的 PINN 无法计算得到参数的真实值，产生了过拟合现象。Allen-Cahn 方程普遍被认为是一个 PINN 难以解决的问题[8]，也有其他研究者提出了不同的思路和方法，比如自适应重采样法。

2.3 一维对流方程

$$\begin{aligned}
 u_t + \beta u_x &= 0, \quad \beta = 30 \\
 x &\in [0, 2\pi], t \in [0, T] \\
 u(x, 0) &= \sin(x) \\
 u(0, t) &= u(2\pi, t)
 \end{aligned} \tag{19}$$

一维对流方程描述了在一维空间中流体或其他物质的流动情况，是许多实际问题的基础模型。PINN 可以通过将一维对流方程嵌入到神经网络的损失函数中来实现对其解的逼近，从而实现对流动过程的建模和求解。与传统的数值方法相比，PINN 不需要对空间进行离散化，可以直接在连续域上求解，从而可以更好地捕捉流动的演化过程。

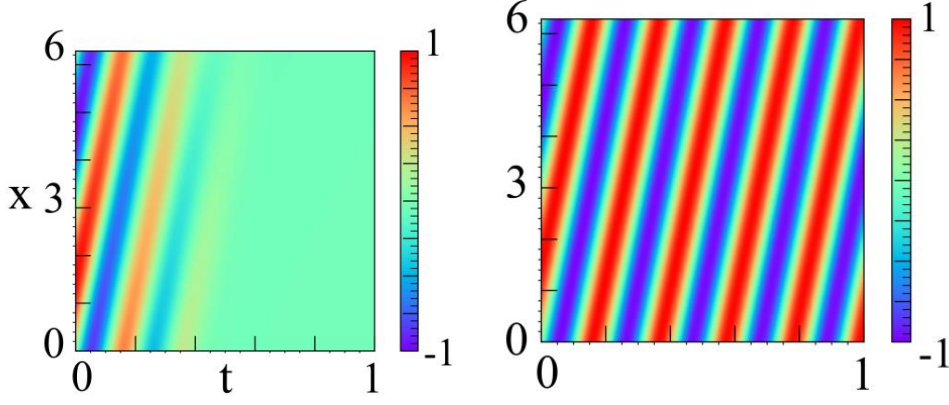


图 9 传统的 PINN 的预测解(左)和基于网格的 PINN 的预测解(右)

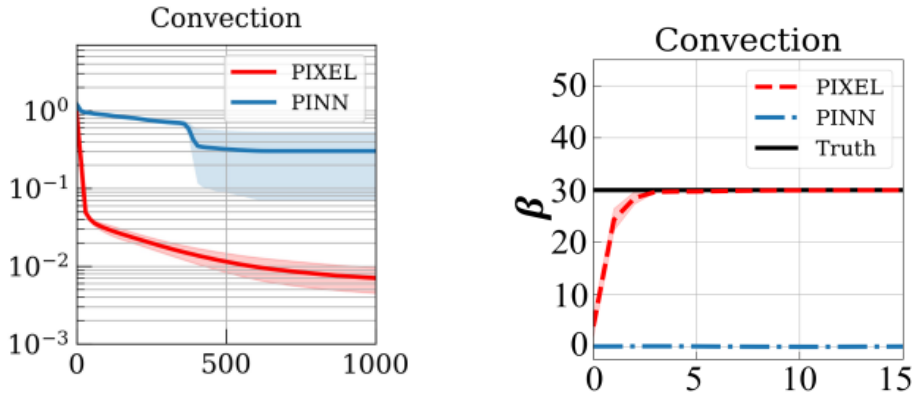


图 10 两种 PINN 在原问题(左)和逆问题(右)收敛速度的比较

实验结果再次表明，传统的 PINN 的预测解与真实解相差甚远，只是在前 500 次迭代后误差略有下降，另一方面，基于网格的 PINN 很好地反应了方程的真实解，尤其是在前 10 次迭代中误差快速下降。在逆问题上，传统的 PINN 算法有重大缺陷，无论何迭代，计算出的参数值 β 一直在初始值 0。我们发现，传统的 PINN 在非狄利克雷边界条件下表现一般。也就是说，如果偏微分方程没有给出方程在边界上的具体函数信息，就会使得训练变得异常困难[9]。传统的 PINN 难以刻画复杂边界条件背后蕴含的物理定律，例如 Allen-Cahn 方程中帶有一阶偏导的边界条件。

2.4 二维椭圆偏微分方程

$$\begin{aligned} \Delta u(x, y) + k^2 u(x, y) &= q(x, y), k = 1 \\ q(x, y) &= k^2 \sin(10\pi x) \sin(10\pi y) - 200\pi^2 \sin(10\pi x) \sin(10\pi y) \\ u(x, y) &= 0, (x, y) \in \partial[-1, 1]^2 \end{aligned} \quad (20)$$

Helmholtz 方程是一种常见的波动方程，描述了波动在空间中的传播情况，广泛应用于声学、光学等领域。PINN 可以通过将 Helmholtz 方程嵌入到神经网络

络的损失函数中来实现对其解的逼近，从而实现对波动传播过程的建模和求解。PINN 可以直接在连续域上求解，从而可以更好地捕捉波动的传播规律。通过使用 PINN，可以更快速、准确地获得波动的解，并且可以在不同边界条件和初始条件下进行灵活的求解和模拟，为声学 and 光学等领域的研究和工程应用提供了一种新的思路。

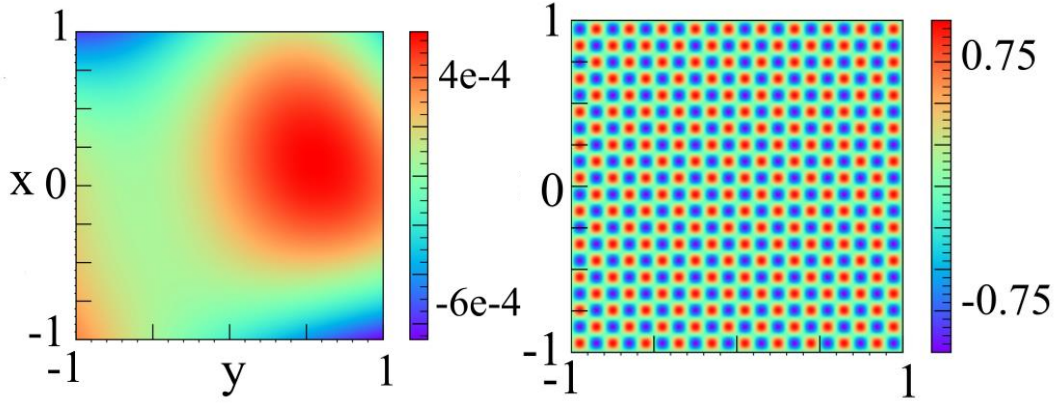


图 11 传统的 PINN 的预测解(左)和基于网格的 PINN 的预测解(右)

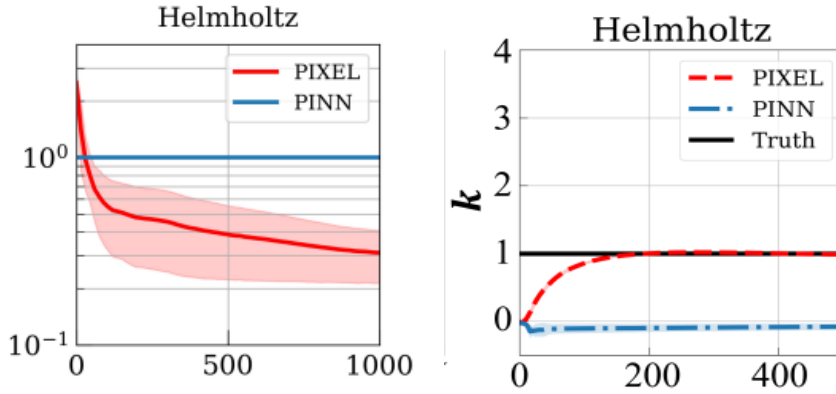


图 12 两种 PINN 在原问题(左)和逆问题(右)收敛速度的比较

方程的右端项 $q(x, y) = k^2 \sin(10\pi x) \sin(10\pi y) - 200\pi^2 \sin(10\pi x) \sin(10\pi y)$ 中 10 是一个可调的参数，反映了波动的频率。一般认为传统的 PINN 难以解决高频率的问题[10]。不出所料，PINN 在这一个问题上具有重大缺陷，不论如何迭代测试误差都没有得到优化。并且在逆问题中也是如此，神经网络一直被困在初始值 0 这一局部最优解中。

3 结论

表 1

	三维椭圆方程	二维椭圆方程	Allen-Cahn方程	Burgers方程	Convection方程
PINN	1.00 ($\pm 7.19\text{e-}04$) (best: 1.00)	1.00 ($\pm 1.49\text{e-}06$) (best : 1.00)	9.08e-01 ($\pm 1.68\text{e-}02$) (best : 5.23e-01)	5.77e-03 ($\pm 1.74\text{e-}03$) (best : 3.35e-03)	3.02e-01 ($\pm 3.40\text{e-}01$) (best : 2.45e-02)
16层基于网格的PINN	5.06e-03 ($\pm 2.64\text{e-}03$) (best : 6.61e-04)	3.05e-01 ($\pm 2.38\text{e-}01$) (best : 7.47e-02)	1.77e-02 ($\pm 4.67\text{e-}03$) (best : 9.64e-03)	9.98e-04 ($\pm 3.70\text{e-}04$) (best : 4.88e-04)	9.48e-03 ($\pm 1.62\text{e-}03$) (best : 6.39e-03)
64层基于网格的PINN	1.95e-01 ($\pm 1.84\text{e-}01$) (best : 4.86e-02)	4.26e-01 ($\pm 3.10\text{e-}01$) (best : 1.05e-01)	1.90e-02 ($\pm 8.35\text{e-}03$) (best : 3.85e-03)	6.20e-04 ($\pm 2.09\text{e-}04$) (best : 3.85e-04)	4.69e-03 ($\pm 1.25\text{e-}03$) (best : 2.41e-03)
96层基于网格的PINN	1.53e-01 ($\pm 6.81\text{e-}02$) (best : 1.34e-02)	3.11e-01 ($\pm 1.43\text{e-}01$) (best : 1.70e-01)	1.63e-02 ($\pm 3.95\text{e-}03$) (best : 8.86e-03)	7.01e-04 ($\pm 3.60\text{e-}04$) (best : 3.20e-04)	6.19e-03 ($\pm 3.36\text{e-}03$) (best : 1.84e-03)

对于前向偏微分方程求解问题，基于网格的 PINN 相对于传统的内嵌物理神经网络在求解精度和收敛速度两方面都有进步，能够有效处理 Allen-Cahn 方程和 Helmholtz 方程，而传统的 PINN 无法精确模拟这两个方程的解。对于偏微分方程的逆问题，传统的物理神经网络容易受到随机数种子的不同选取的影响，而基于网格的 PINN 在预测方面表现出很强的鲁棒性，无论随机数种子如何，都在 95% 置信区间内。

总的来说，我们提出了一种基于双线性插值和内嵌物理神经网络的偏微分方程深度学习算法，该算法能够有效提高偏微分方程求解的精度和效率；研究了复杂偏微分方程问题的求解，拓展了深度学习算法的应用范围。我们的研究表明，基于网格的PINN方法在处理复杂PDE问题时具有明显的优势。其主要原因在于基于网格的方法能够更好地处理空间离散化和复杂边界条件的问题，从而提高了求解的精度和效率。与传统的PINN相比，基于网格的PINN更加灵活和适用于不同类型的PDE问题，为PDE深度学习算法的发展和应用提供了新的思路和方法。

参考文献

- [1] Yanzhong Yao, Jiawei Guo, Tongxiang Gu, A deep learning method for multi-material diffusion problems based on physics-informed neural networks
- [2]Anagnostopoulos, S. J., Toscano, J. D., Stergiopoulos, N., & Karniadakis, G. E. (2024). Residual-based attention in physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421, 116805.
- [3]Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561. <http://arxiv.org/pdf/1711.10561v1>
- [4]Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (1907). DeepXDE: A deep learning library for solving differential equations,(2019). URL <http://arxiv.org/abs/1907.04502>. <https://arxiv.org/abs/1907.04502>
- [5]Lu, L., Jin, P., & Karniadakis, G. E. (2019). Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv preprint arXiv:1910.03193.
- [6]Wang, S., Wang, H., & Perdikaris, P. (2021). Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science advances*, 7(40), eabi8605.
- [7] M. Raissi , P. Perdikaris, G.E. Karniadakis Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations
- [8] Colby L. Wight, Jia Zhao Solving Allen-Cahn and Cahn-Hilliard Equations using the Adaptive Physics Informed Neural Networks
- [9] Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, Michael W. Mahoney Characterizing possible failure modes in physics-informed neural networks
- [10] SIFAN WANG, YUJUN TENG,AND PARIS PERDIKARIS Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks

致谢

对于这次毕业实践报告的撰写，最需要感谢的是张晓平老师，他在整个过程中都给予了我充分的帮助和支持。老师不仅耐心地为之处毕业论文中的不足之处，对毕业论文的改进提出宝贵的建议，而且还在我遇到困难时尽心地进行指点与解答。在此借毕业论文完成之际，表示由衷的感谢和敬意。