

基于数值微分的物理信息神经网络方法 在非矩形区域上的应用

康豪, 张羽

西南科技大学数理学院, 四川 绵阳

收稿日期: 2023年4月21日; 录用日期: 2023年4月27日; 发布日期: 2023年6月21日

摘要

物理信息神经网络(PINN)是一种新兴的数据驱动的偏微分方程数值求解框架。该类方法不需要进行网格划分, 从而避免多余的计算消耗, 这使得其对高维问题有更好的拓展性。但传统神经网络技术基于自动微分实现, 依赖于大量的训练配点, 容易引起梯度爆炸或梯度消失现象, 且不能在非矩形区域上直接进行求解。文章简要阐述了基于数值微分的物理信息神经网络的基本原理, 使用数值微分替代自动微分以避免使用大量的训练配点, 避免梯度爆炸或梯度消失发生。同时, 利用区域分解思想对非矩形区域进行划分, 使划分后的子区域可以使用PINN直接求解, 最终用一个算例验证了该方法的可行性。结果表明, 基于数值微分的物理信息神经网络可以求解非矩形区域上的问题。

关键词

物理信息神经网络, 数值微分, 区域分解

Application of Physics Informed Neural Network Method Based on Numerical Differentiation in Non-Rectangular Regions

Hao Kang, Yu Zhang

School of Mathematics and Physics, Southwest University of Science and Technology, Mianyang Sichuan

Received: Apr. 21st, 2023; accepted: Apr. 27th, 2023; published: Jun. 21st, 2023

Abstract

Physics-Informed Neural Networks (PINN) is a novel data-driven numerical framework for solving

文章引用: 康豪, 张羽. 基于数值微分的物理信息神经网络方法在非矩形区域上的应用[J]. 流体动力学, 2023, 11(2): 70-79. DOI: 10.12677/ijfd.2023.112007

partial differential equations. Since this type of technique does not require meshing, it consumes fewer computer resources and is more scalable for high-dimensional issues. Traditional neural network technology, on the other hand, relies on a high number of training points and depends on automatic differentiation, which makes it easy to produce gradient expansion or gradient disappearance and prevents direct solution on non-rectangular areas. In order to avoid the requirement for many training points and to prevent gradient explosion or gradient disappearance, this paper uses numerical differentiation rather than automatic differentiation to briefly explain the fundamental concepts of Physics-Informed Neural Networks. In order for the divided subregion to be directly solved using PINN, the non-rectangular area is simultaneously divided using the domain decomposition idea. Finally, a calculation scenario is utilized to confirm the viability of the method. The outcomes demonstrate the ability of the numerically differentiated, Physics-Informed Neural Networks to address issues in non-rectangular locations.

Keywords

PINN, Numerical Differentiation, Domain Decomposition

Copyright © 2023 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/>



1. 引言

在众多机器学习算法取得快步发展之前, 工程领域通常采用的都是传统的物理模型。这些领域的大部分问题都以偏微分方程(Partial Differential Equation, PDE)的形式刻画或描述。直接求解物理模型可进行精准预测, 但复杂的场景下的偏微分方程的解通常难以显式地表示。传统数值求解方法在高维度的情况下, 计算代价相当大。相较于基于网格的方法, 深度学习能弥补传统数值方法求解偏微分方程的先天缺陷[1]。因此, 在理论研究和工程应用中开发设计出高效精确的基于深度学习的偏微分方程数值解法对降低计算成本和提高计算精度具有重要意义。

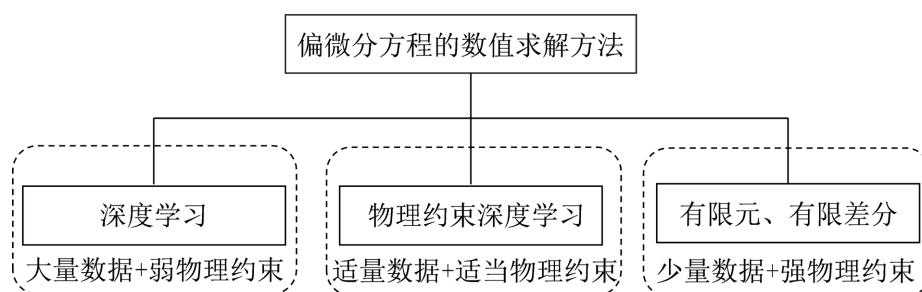


Figure 1. Three numerical methods for solving partial differential equations

图 1. 求解偏微分方程的三种数值方法

2017 年, Raissi 教授和他的合作者们提出了一套深度学习算法框架, 并将其命名为“Physics-informed Neural Networks”(物理信息神经网络) [2]。2021 年, Karniadakis, Lu 等研究人员在著名的 Nature 子刊 Nature Reviews Physics 上发表了一篇综述性文章[3], 值得注意的是, 作者将物理约束和数据驱动结合的方法, 归纳统称为物理信息约束的深度学习方法(Physics-informed Deep Learning, PIDL), 见图 1。

在深度学习方法中, 增加网络的深度和复杂性可能会导致损失函数的参数空间中具有梯度爆炸或梯

度消失的区域。如果出现梯度爆炸或消失, 模型就会变得非常不稳定, 并使先前的训练进展变得徒劳无功。传统深度学习方法以链式求导法则和自动微分技术为基础, 能够在网络结构中前馈和反向传播梯度信息。但基于自动微分的 PINN 只能在大量训练配点的情况下实现高精度求解, 否则它们很容易偏向于完全的数据驱动方案, 进而优化到不满足物理条件的解[4]。为了提升 PINN 在稀疏样本下的训练效率, 可以采用数值微分来定义损失函数。基于数值微分的损失函数能够强连接相邻的搭配点, 以便在稀疏样本状态下实现有效的训练。

原始的 PINN 方法需要根据问题的定解条件在一个给定的矩形区域上进行求解。但实际情况中常常因为解函数存在区域内的某些部分无法进行直接观测, 从而无法获得该部分的观测信息, 进而导致只能在非矩形区域上求解。2019 年提出的 DeepXDE 软件库中提供了 7 种基本几何结构[5], 可以帮助使用者利用基于自动微分的 PINN 在一些更复杂的问题区域上进行求解和模拟。

受此启发, 本文结合区域分解的思想, 将基于数值微分的 PINN 方法应用到非矩形区域上, 以扩展原有方法的适用范围。文章第一节简要叙述了 PINN 的发展历程; 第二节给出 PINN 的一般模型、基于数值微分的 PINN 的基本原理、区域分解技术和 KdV 方程的问题模型; 第三节展开数值实验, 使用基于数值微分的 PINN 方法对一个非矩形区域上的 KdV 方程进行求解, 以验证方法的有效性; 第四节总结了方法的优缺点, 并提出了潜在的应用前景。

2. 基本原理

2.1. 物理信息神经网络

在物理信息神经网络的框架下, 我们考虑具有如下形式的偏微分方程:

$$u_t + \mathbf{N}[u] = 0, x \in \Omega, t \in [0, T] \quad (1)$$

其中, u_t 代表了方程的隐藏解, $\mathbf{N}[\cdot]$ 是一个非线性微分算子, 而 Ω 是 R^n 的子集, T 是终止时刻, 下标表示对时间或空间的微分运算。在给定初始、边界条件和物理参数且无法求得解析解时, 可以利用有限差分、有限元等传统数值方法求解。PINN 则是考虑建立一个神经网络来逼近偏微分方程的解。假设 $u^N(t, x)$ 即为用于逼近解函数 $u(t, x)$ 的神经网络, 偏微分方程的残差即为:

$$r(t, x) = u^N(t, x) + \mathbf{N}[u^N(t, x)] \quad (2)$$

并定义物理信息神经网络的损失函数为:

$$Loss = Loss_u + Loss_r \quad (3)$$

其中, $Loss_u$ 表示损失函数中的数据驱动部分。数据驱动部分以在初始、边界条件上, 根据实验观测或物理仿真取得的训练数据进行训练。这部分损失函数依赖观测数据作为标签指导模型进行训练, 属于监督学习过程。另一部分损失函数 $Loss_r$ 则表示损失函数中的物理约束, 也即物理约束部分[6]。由于自动微分技术可以高效地获得方程的残差, 这部分损失函数在训练时不需要额外的标签数据, 属于无监督学习过程。

这种结合监督学习和无监督学习的设计使得损失函数不但可以保证和已知数据的误差尽可能小, 而且可以尽可能地满足偏微分方程隐含的物理约束。在输入需要的时空数据后, 首先使用一个全连接神经网络来逼近解函数, 然后利用自动微分技术求出残差放入损失函数, 最后使用梯度下降等优化算法获得全连接神经网络的权重参数, 最终训练出可以满足偏微分方程的神经网络参数以逼近解函数。

2.2. 嵌入数值微分技术

作为基于神经网络的无网格方法, PINN 相较传统基于网格的数值算法可以节省大量计算开销, 这就使其在高维问题上有了用武之地, 并且已经证明这种方法在求解椭圆型方程时可以依范数收敛到问题的

真实解[7]。但自动微分技术需要使用大量的训练配点, 增加网络深度又可能导致神经网络在训练过程中出现梯度爆炸或梯度消失的现象[8], 对此问题, 采用基于数值微分的神经网络技术更为恰当。

在基于数值微分的 PINN 方法中, 使用差分方法而非自动微分来实现数值微分过程。通过选择合适的点并利用泰勒级数展开消除误差项, 可以得到数值微分导数项。将函数 $u(x+\Delta x)$, $u(x-\Delta x)$ 进行泰勒展开, 联立得到 $u(x)$ 的一阶导数 $u_x(x)$ 的二阶中心差分公式为:

$$u_x(x) = \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} \quad (4)$$

类似的, 得到 $u(x)$ 的二阶导数 $u_{xx}(x)$ 的二阶中心差分公式为:

$$u_{xx}(x) = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} \quad (5)$$

$u(x)$ 的三阶导数 $u_{xxx}(x)$ 的二阶中心差分公式为:

$$u_{xxx}(x) = \frac{0.5 * u(x+2\Delta x) - u(x+\Delta x) + u(x-\Delta x) - 0.5 * u(x-2\Delta x)}{(\Delta x)^3} \quad (6)$$

本文使用上述差分公式进行数值微分, 以替代神经网络框架中的自动微分过程。如图 2 所示, 上述差分格式能够以不同权重强耦合相邻的训练配点, 以便在稀疏样本状态下的模型仍能获得足够信息, 进而实现有效的训练。

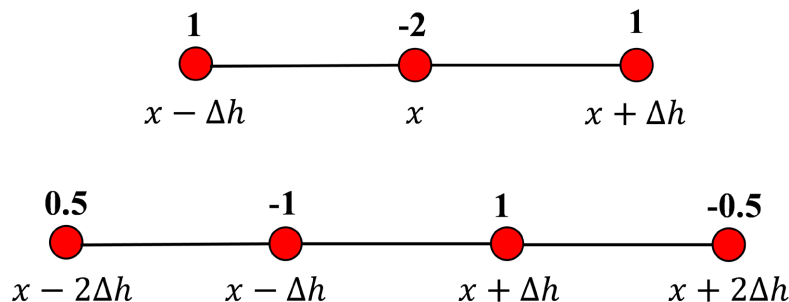


Figure 2. Schematic diagram of the central difference format of the second and third derivatives

图 2. 二阶导数和三阶导数的中心差分格式示意图

2.3. 区域分解求解思想

区域分解技术基于分治思想, 其本身是一类迭代方法。具体来说, 区域分解技术基于计算区域的划分:

$$\Omega = \bigcup_{i=1}^n \Omega_i \quad (7)$$

然后将计算过程和计算量分配到子区域上进行。对子区域进行划分能够极大地提高算法的可并行性, 使得利用并行计算技术加速求解成为可能; 同时可以结合各子区域的特点, 在各子区域中使用不同的数学模型、计算方法和求解格式, 以加快运算速度。受迭代方法中的区域分解思想启发, 软件库 DeepXDE 中的 PINN 方法首先定义了一些基本几何图形, 通过将复杂区域分解为几种这些基本几何图形来实现将原问题在复杂区域进行求解。但这种基于区域分解的求解思想尚未在有数值微分的 PINN 方法中有过具体应用。

3. 模型建立

KdV 方程是一个描述浅水孤立波运动的物理模型, KdV 方程与孤子解在流体力学、光学等许多领域得到了广泛的研究和应用。考虑 KdV 方程:

$$\frac{\partial u}{\partial t} + 6\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial t^3} = 0 \tag{8}$$

给定不同的定解条件, 该方程有不同的解。其一种孤立波解的解析形式为:

$$u(x,t) = \frac{c}{2} \operatorname{sech}^2\left(\frac{\sqrt{c}}{2}(x-ct-x_0)\right) \tag{9}$$

该解析解由 Stephen Anco 博士给出。其中, c 代表波速。取 $c = 2$, $x_0 = -1$, 定义解空间为:

$$\begin{cases} x \in [-3, 3], & t \in [0, 1] \\ x \in [-3, 0], & t \in [1, 2] \end{cases} \tag{10}$$

将该 L 型区域分解为两个子区域 Ω_1 和 Ω_2 , 其中 Ω_1 为 $x \in [0, 3]$, $t \in [0, 1]$ 的矩形, Ω_2 为 $x \in [-3, 0]$, $t \in [0, 2]$ 的矩形。在两个子区域上分别定义一个物理信息神经网络进行求解。

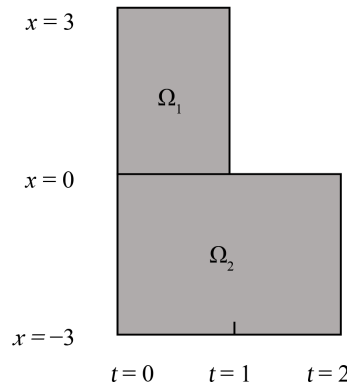


Figure 3. Domain decomposition
图 3. 区域分解示意图

根据方程的具体形式, 令:

$$r(t,x) = \frac{\partial u}{\partial t} + 6\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial t^3} \tag{11}$$

由式(3), 定义物理信息神经网络的损失函数为:

$$Loss = Loss_u + Loss_r = \frac{1}{N_u} \sum_{i=1}^{N_u} \left| u^N(t_u^i, x_u^i) - u^i \right|^2 + \frac{1}{N_r} \sum_{i=1}^{N_r} \left| r(t_u^i, x_u^i) \right|^2 \tag{12}$$

其中, N_u 为观测数据点的个数, N_r 为根据残差计算点的个数。设差分步长 $h = 0.01$, 使用式(4)和式(6)对方程(8)中的一阶导数和三阶导数进行近似。

4. 数值实验

为了验证上述方法的可行性, 本文设计了数值实验进行验证。

实验中的 KdV 方程的参数 $c = 2$, $x_0 = -1$, 求解的区域范围如式(12)所示, 区域分解方案如图 3 所示。

实验中的物理信息神经网络的拓扑结构为全连接网络, 输入层有 2 个神经元、输出层 1 个神经元, 隐藏层共 2 层, 每层 50 个神经元, 隐藏层间以双曲正切函数为激活函数。神经网络使用 Adam 优化器以 0.01 为学习率迭代 10,000 次, 对两个子区域都分别采用这样的神经网络进行求解。实验中的数值微分方法采用式(6)和式(8)的中心差分公式, 步长设置为 0.01。所有实验均在 Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz 上进行, 编程语言采用 Python。

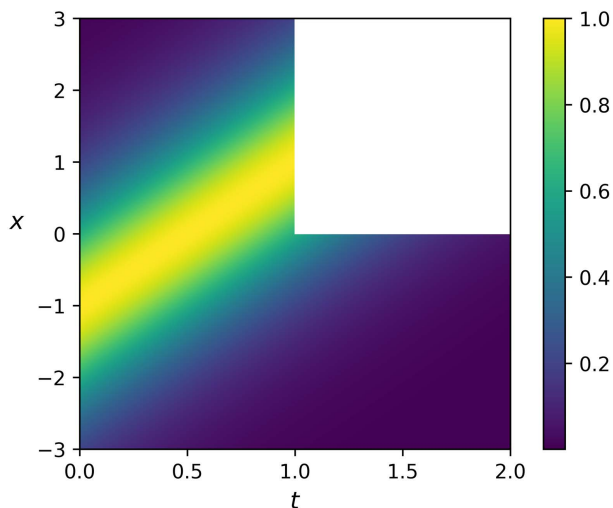


Figure 4. Solution of PINN based on numerical differentiation
图 4. 基于数值微分的 PINN 的求解结果

图 4 展示了基于数值微分的物理信息神经网络在 L 型区域上求解 KdV 方程的求解结果。总体来看, 基于数值微分的 PINN 在 L 型区域上比较好地描述了孤立波解的基本性态, 可以清晰看到孤立波的运动过程。为了分析在进行区域分解后的 PINN 的求解效果, 下面分别单独讨论 PINN 在子区域 Ω_1 , Ω_2 上的求解效果。

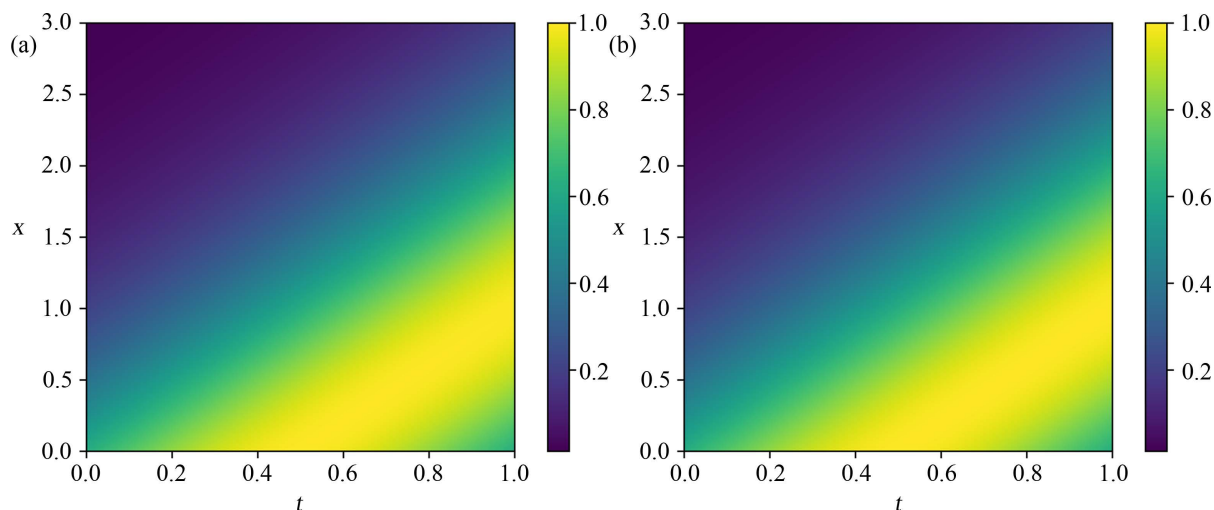


Figure 5. Comparison of solution on Ω_1
图 5. 区域 Ω_1 上的求解结果对比

单独考虑子区域 Ω_1 , 图 5 将 PINN 的求解结果和真实解的图像进行了对比。图 5(a)为 PINN 求解结

果示意图, 图 5(b)为真实解示意图。直接观察图像, 两者的解函数性态并无较大差异, PINN 较为清晰地展现了孤立波在该区域上的运动过程。观察 PINN 在区域 Ω_1 上的绝对误差, 见图 6(a), 在 $t=0$ 到 $t=0.8$ 秒的区间上, PINN 的绝对误差都保持在较低的水平且分布比较平均, 在接近 $t=1$ 秒时, PINN 的误差开始变大。如图 6(b), 从收敛曲线来看, 基于数值的 PINN 的收敛非常迅速, 在迭代初期就迅速下降到了 10^{-2} 的数量级上, 且迭代过程整体保持平稳, 并未出现剧烈震荡。

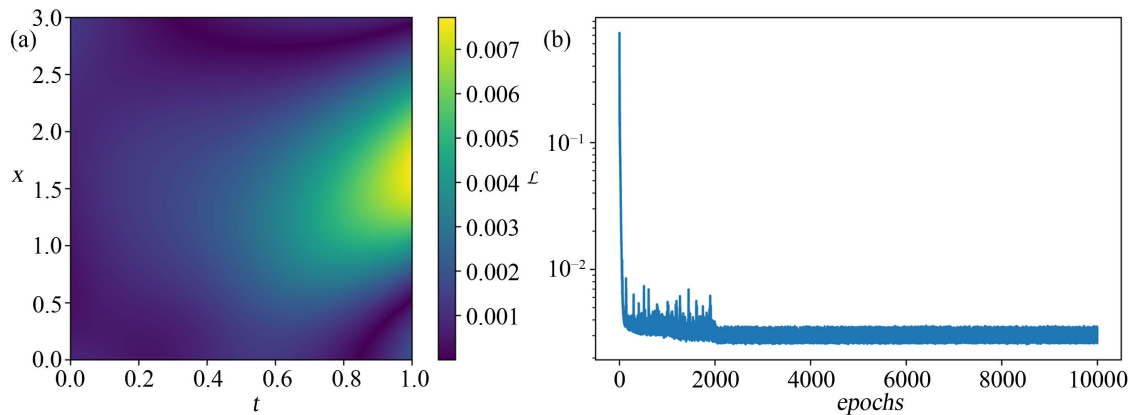


Figure 6. Convergence curve of absolute error and loss function on Ω_1
图 6. 区域 Ω_1 上的绝对误差和损失函数收敛曲线

单独考虑子区域 Ω_2 , 从解出的图像来看, PINN 的结果仍然比较清晰。但与 PINN 在区域 Ω_1 上的绝对误差表现略有不同, 基于数值微分的 PINN 方法在 Ω_2 上的绝对误差分布并不平均, 且随着 t 的增大, 绝对误差也有增大的趋势。从收敛曲线来看, 基于数值的 PINN 的损失函数在区域 Ω_2 上仍然迅速下降到了 10^{-2} 的数量级, 但在 2000 步前出现了一些震荡, 2000 步后才保持平稳。

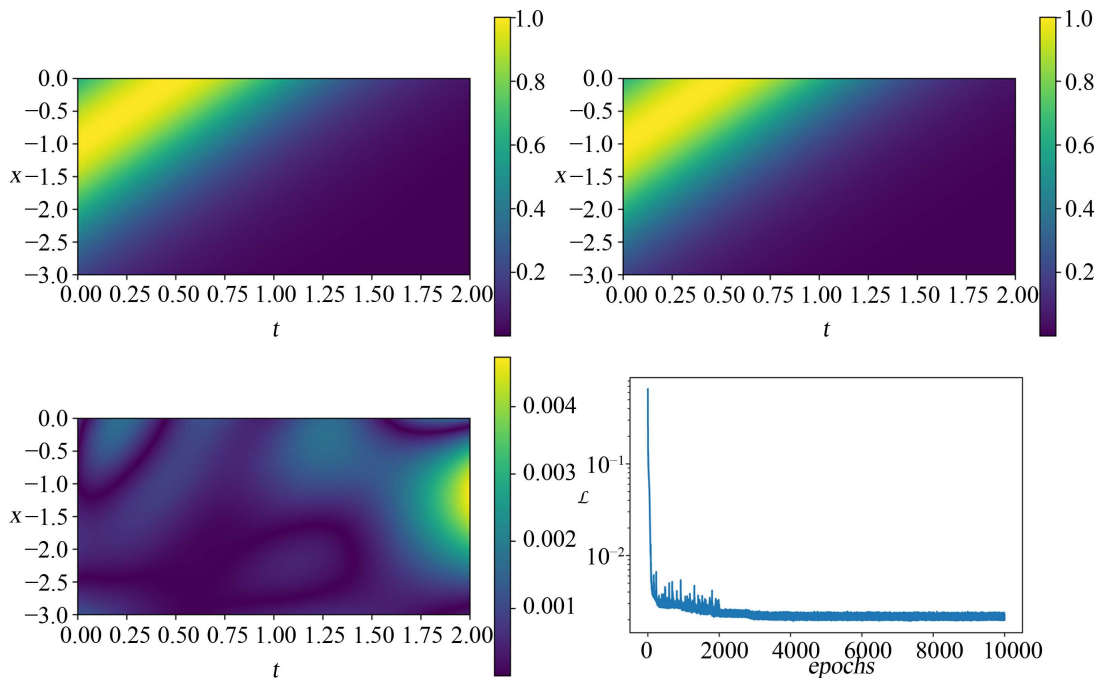


Figure 7. Solution and convergence process of PINN on Ω_2
图 7. PINN 在区域 Ω_2 上的求解结果和收敛过程示意图

图 4~7 展示了 PINN 求解 KdV 方程得到的图像, 可以直接观察 PINN 的求解效果。为了展示更为详细的求解结果, 分别以 $\Delta t = 0.08$ 和 $\Delta x = 0.15$ 为粒度进行采样, 将使用上述物理信息神经网络在 L 型区域上求解 KdV 方程的部分结果数据汇总如下。 $t \in [0, 1]$, $x \in [-3, 3]$ 时的求解结果在表 1 中给出; $t \in [1, 2]$, $x \in [-3, 3]$ 时的求解结果在表 2 中给出, 二者共同组成 L 型区域上的求解结果。例如, $x = 0$, $t = 0.08$ 时, 神经网络预测值 $\hat{u} = 0.174$, 以此类推。神经网络的求解结果可以通过设置不同的粒度进行输出, 极大方便了该方法在各种实际场景下的应用。

Table 1. Display of data sampling of PINN's solution

表 1. 物理信息神经网络求解结果数据采样展示

\hat{u}	t											
	0.080	0.160	0.240	0.319	0.399	0.479	0.559	0.639	0.719	0.798	0.878	0.958
0	0.174	0.142	0.115	0.093	0.075	0.060	0.049	0.039	0.031	0.025	0.020	0.016
0.15	0.253	0.208	0.170	0.138	0.112	0.090	0.073	0.059	0.047	0.038	0.030	0.024
0.30	0.360	0.299	0.247	0.203	0.165	0.134	0.109	0.088	0.071	0.057	0.046	0.037
0.45	0.496	0.420	0.352	0.293	0.241	0.198	0.161	0.131	0.106	0.085	0.069	0.055
0.60	0.655	0.568	0.487	0.412	0.344	0.286	0.235	0.193	0.157	0.127	0.103	0.083
0.75	0.814	0.730	0.644	0.558	0.477	0.403	0.337	0.279	0.229	0.188	0.153	0.124
0.90	0.940	0.879	0.803	0.720	0.633	0.548	0.467	0.394	0.329	0.272	0.224	0.183
1.05	0.998	0.977	0.933	0.870	0.793	0.709	0.622	0.537	0.458	0.385	0.321	0.265
1.20	0.970	0.996	0.997	0.973	0.926	0.861	0.783	0.698	0.611	0.527	0.448	0.377
1.35	0.864	0.929	0.975	0.998	0.995	0.968	0.919	0.852	0.773	0.687	0.601	0.517
x 1.50	0.712	0.797	0.873	0.936	0.979	0.998	0.993	0.963	0.911	0.843	0.763	0.677
1.65	0.551	0.636	0.723	0.807	0.882	0.942	0.982	0.999	0.991	0.958	0.904	0.833
1.80	0.405	0.480	0.561	0.647	0.734	0.816	0.890	0.948	0.985	0.999	0.988	0.952
1.95	0.287	0.346	0.414	0.490	0.572	0.658	0.744	0.826	0.897	0.953	0.988	0.999
2.10	0.198	0.242	0.294	0.354	0.423	0.499	0.582	0.668	0.754	0.835	0.905	0.958
2.25	0.135	0.166	0.204	0.249	0.301	0.363	0.432	0.509	0.592	0.679	0.764	0.844
2.40	0.090	0.112	0.138	0.170	0.209	0.255	0.308	0.371	0.441	0.519	0.603	0.689
2.54	0.060	0.074	0.093	0.115	0.142	0.175	0.214	0.261	0.316	0.379	0.450	0.529
2.69	0.040	0.049	0.061	0.077	0.095	0.118	0.146	0.180	0.220	0.267	0.323	0.387
2.84	0.026	0.032	0.040	0.051	0.063	0.079	0.098	0.122	0.150	0.184	0.225	0.274
2.99	0.016	0.021	0.026	0.033	0.042	0.052	0.066	0.082	0.101	0.125	0.154	0.189

Table 2. Display of data sampling of PINN's solution
表 2. 物理信息神经网络求解结果数据采样展示

\hat{u}	t													
	1.038	1.118	1.198	1.277	1.357	1.437	1.517	1.597	1.677	1.756	1.836	1.916	1.996	
x	0	0.013	0.010	0.008	0.006	0.005	0.004	0.003	0.002	0.002	0.002	0.001	0.001	0.001
	0.15	0.019	0.015	0.012	0.010	0.008	0.006	0.005	0.004	0.003	0.002	0.002	0.002	0.001
	0.30	0.029	0.024	0.019	0.015	0.012	0.010	0.008	0.006	0.005	0.004	0.003	0.002	0.002
	0.45	0.044	0.036	0.029	0.023	0.018	0.015	0.012	0.009	0.007	0.006	0.004	0.003	0.003
	0.60	0.067	0.054	0.043	0.035	0.028	0.022	0.018	0.014	0.011	0.009	0.007	0.005	0.004
	0.75	0.100	0.081	0.065	0.052	0.042	0.034	0.027	0.022	0.017	0.014	0.011	0.008	0.006
	0.90	0.148	0.120	0.097	0.078	0.063	0.051	0.041	0.033	0.026	0.021	0.016	0.013	0.010
	1.05	0.218	0.178	0.144	0.117	0.094	0.076	0.061	0.049	0.039	0.031	0.025	0.020	0.016
	1.20	0.314	0.259	0.212	0.173	0.141	0.114	0.092	0.074	0.060	0.048	0.038	0.031	0.024
	1.35	0.439	0.369	0.306	0.253	0.207	0.169	0.137	0.111	0.090	0.072	0.058	0.047	0.037

5. 结论

偏微分方程的数值求解方法始终是一类重要的研究热点。传统数值求解方法包括有限差分、有限元、有限体积等。这些方法需要使用网格来近似偏微分方程的定义空间, 在高维度的情况下, 其计算代价相当大。与之不同的是, 经典的机器学习算法都以纯数据驱动为主。相较于基于网格的方法, 深度学习方法能使参数数量极大地降低, 从而带来更少的计算消耗, 使得其对高维问题有更好的扩展性, 这就能弥补传统数值方法求解偏微分方程的先天缺陷。

基于数值微分的 PINN 方法可以看作是传统数值方法与数据驱动算法的结合, 这类 PINN 方法既保留了物理信息神经网络的原有优势, 又避免了原始框架中存在的梯度爆炸等隐患, 并且能够在稀疏样本下进行训练。既往研究中求解的问题都是在矩形区域上进行的, 本文将这一方法应用到一个非矩形区域内。数值实验表明, 基于数值微分的物理信息神经网络方法能够有效应对非矩形区域的 KdV 方程的求解任务, 且其求解效果较好、收敛速度较快。针对其他类型的方程的求解还需要进一步研究。

基金项目

四川省大学生创新创业训练计划创新训练项目(S202210619108)。

参考文献

- [1] 李荣华. 偏微分方程数值解法[M]. 第 2 版. 北京: 高等教育出版社, 2010.
- [2] 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>
- [3] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., et al. (2021) Physics-Informed Machine Learning. *Nature Reviews Physics*, **3**, 422-440. <https://doi.org/10.1038/s42254-021-00314-5>
- [4] Chiu, P.-H., Wong, J.C., Ooi, C., et al. (2022) CAN-PINN: A Fast Physics-Informed Neural Network Based on

-
- Coupled-Automatic-Numerical Differentiation Method. *Computer Methods in Applied Mechanics and Engineering*, **395**, 114909. <https://doi.org/10.1016/j.cma.2022.114909>
- [5] Lu, L., Meng, X., Mao, Z., *et al.* (2019) DeepXDE: A Deep Learning Library for Solving Differential Equations. *SIAM Review*, **63**, 208-288. <https://doi.org/10.1137/19M1274067>
- [6] 李野, 陈松灿. 基于物理信息的神经网络: 最新进展与展望[J]. 计算机科学, 2022, 49(4): 254-262.
- [7] Mishra, S. and Molinaro, R. (2022) Estimates on the Generalization Error of Physics-Informed Neural Networks for Approximating PDEs. *IMA Journal of Numerical Analysis*, **43**, 1-43. <https://doi.org/10.1093/imanum/drab093>
- [8] Krishnapriyan, A.S., Gholami, A., *et al.* (2021) Characterizing Possible Failure Modes in Physics-Informed Neural Networks. *Proceedings of the Neural Information Processing Systems*, New York, September 2021, 26548-26560.