

基于Runge-Kutta的自回归物理信息神经网络求解偏微分方程

韦 昌, 樊昱晨, 周永清, 张超群, 刘 欣, 王赫阳

SELF-REGRESSIVE PHYSICS-INFORMED NEURAL NETWORK BASED ON RUNGE-KUTTA METHOD FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

Wei Chang, Fan Yuchen, Zhou Yongqing, Zhang Chaoqun, Liu Xin, and Wang Heyang

在线阅读 View online: <https://doi.org/10.6052/0459-1879-24-106>

您可能感兴趣的其他文章

Articles you may be interested in

基于神经网络的差分方程快速求解方法

A FAST SOLVER BASED ON DEEP NEURAL NETWORK FOR DIFFERENCE EQUATION

力学学报. 2021, 53(7): 1912-1921

面内变刚度薄板弯曲问题的挠度弯矩耦合神经网络方法

DEFLECTION-BENDING MOMENT COUPLING NEURAL NETWORK METHOD FOR THE BENDING PROBLEM OF THIN PLATES WITH IN-PLANE STIFFNESS GRADIENT

力学学报. 2021, 53(9): 2541-2553

基于人工神经网络的非结构网格尺度控制方法

UNSTRUCTURED MESH SIZE CONTROL METHOD BASED ON ARTIFICIAL NEURAL NETWORK

力学学报. 2021, 53(10): 2682-2691

基于人工神经网络的湍流大涡模拟方法

ARTIFICIAL NEURAL NETWORK-BASED SUBGRID-SCALE MODELS FOR LARGE-EDDY SIMULATION OF TURBULENCE

力学学报. 2021, 53(1): 1-16

基于人工神经网络的声子晶体逆向设计

INVERSE DESIGN OF PHONONIC CRYSTALS BY ARTIFICIAL NEURAL NETWORKS

力学学报. 2021, 53(7): 1992-1998

基于组合神经网络的雷诺平均湍流模型 多次修正方法

A COMBINED NEURAL NETWORK AND MULTIPLE MODIFICATION STRATEGY FOR REYNOLDS-AVERAGED NAVIER-STOKES TURBULENCE MODELING

力学学报. 2021, 53(6): 1532-1542



关注微信公众号, 获得更多资讯信息

基于 Runge-Kutta 的自回归物理信息神经网络求解偏微分方程

韦 昌* 樊昱晨* 周永清* 张超群† 刘 欣*,† 王赫阳*,1)

* (天津大学机械工程学院, 天津 300072)

† (烟台龙源电力技术股份有限公司, 山东烟台 264006)

摘要 物理信息神经网络离散时间模型 (PINN-RK) 是深度学习技术与龙格库塔方法相结合的产物, 在求解偏微分方程时具有非常出色的稳定性和较高的求解精度。但是, 受到龙格库塔算法本身的限制, PINN-RK 模型仅能实现单步时间预测, 且计算效率较低。因此, 为了实现多时间步长预测和提高模型的计算效率, 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)。该模型基于自回归时间步进机制, 改进了神经网络的训练流程和网络结构, 相比 PINN-RK 模型, 大幅减少了神经网络的训练参数, 提高了模型的计算效率。此外, 在自回归机制的作用下, 该模型通过对标签数据的动态更新, 成功实现了对偏微分方程解的多时间步长预测。为了验证文中模型的求解精度和计算效率, 分别求解了 Allen-Cahn 方程和 Burgers 方程, 并与文献中的基准解进行了对比。结果表明, 模型预测解与基准解之间具有很高的 consistency, 求解 Allen-Cahn 方程和 Burgers 方程的最大相对误差均低于 0.009。

关键词 物理信息神经网络, 自回归时间步进机制, 偏微分方程, Allen-Cahn 方程, Burgers 方程

中图分类号: TP183 文献标识码: A doi: 10.6052/0459-1879-24-106

SELF-REGRESSIVE PHYSICS-INFORMED NEURAL NETWORK BASED ON RUNGE-KUTTA METHOD FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

Wei Chang* Fan Yuchen* Zhou Yongqing* Zhang Chaoqun† Liu Xin*,† Wang Heyang*,1)

* (School of Mechanical Engineering, Tianjin University, Tianjin 300072, China)

† (Yantai Longyuan Power Technology Co., Ltd., Yantai 264006, Shandong, China)

Abstract Physics-informed neural networks discrete-time model (PINN-RK) is a product of combining deep learning techniques with Runge-Kutta method, which has excellent stability and high accuracy in solving partial differential equations. As an emerging computational tool, PINN-RK has been widely applied in solving various complex problems in scientific and engineering fields. However, due to the limitations of the Runge-Kutta algorithm itself, the PINN-RK model can only achieve single-step time prediction and has low computational efficiency. To achieve multi-step time prediction and improve the computational efficiency of the model, this paper proposes a novel self-regressive physics-informed neural networks model based on the Runge-Kutta method (SR-PINN-RK). The SR-PINN-RK model builds

2024-03-04 收稿, 2024-04-18 录用, 2024-04-19 网络版发表。

1) 通讯作者: 王赫阳, 教授, 主要研究方向为智能燃烧算法和工程应用数值模拟。E-mail: heyang.wang@tju.edu.cn

引用格式: 韦昌, 樊昱晨, 周永清, 张超群, 刘欣, 王赫阳. 基于 Runge-Kutta 的自回归物理信息神经网络求解偏微分方程. 力学学报, 2024, 56(8): 2482-2493

Wei Chang, Fan Yuchen, Zhou Yongqing, Zhang Chaoqun, Liu Xin, Wang Heyang. Self-regressive physics-informed neural network based on Runge-Kutta method for solving partial differential equations. *Chinese Journal of Theoretical and Applied Mechanics*, 2024, 56(8): 2482-2493

upon the PINN-RK model by incorporating a self-regressive time-advancing mechanism which allows the SR-PINN-RK model to learn the temporal dynamics of the partial differential equation more effectively, resulting in improved training performance and accuracy. In SR-PINN-RK model, except for the label data at the initial time given by the user, all other training labels are provided by the neural network model itself. The new PINN-RK model is a significant improvement over the PINN-RK model, with a much smaller number of training parameters and a significant boost in computational efficiency. This makes SR-PINN-RK model much faster and easier to train, while still maintaining the same level of accuracy. The SR-PINN-RK model uses a self-regressive mechanism to dynamically update the label data, which allows it to successfully achieve multi-step time prediction of partial differential equation solutions. This represents a remarkable improvement compared to the PINN-RK model, which is limited to single-step predictions. In order to verify the accuracy and computational efficiency of the SR-PINN-RK model, the Allen-Cahn equation and Burgers equation are solved using the SR-PINN-RK model and the predicted results are compared with the benchmark solutions in the literature. It is shown that the predicted solution of the SR-PINN-RK model is highly consistent with the benchmark solution qualitatively without significant differences. When it comes to the quantitative analysis, it can be observed that the maximum relative error in solving the Allen-Cahn and Burgers equations are both below 0.009, thereby exhibiting a remarkable level of precision in solving partial differential equations.

Key words physics-informed neural network, self-regressive time-advancing mechanism, partial differential equations, Allen-Cahn equation, Burgers equation

引言

近 10 年来, 随着大数据的海量积累和 GPU 算力的革命性提升, 人工智能领域迎来了蓬勃的发展. 在诸多人工智能技术中, 深度学习技术更是在自动驾驶^[1]、图像识别^[2]和自然语言处理^[3]等方面取得了显著的成果. 此外, 深度学习技术作为一种新兴的计算手段, 其理论和方法已被广泛应用于解决科学领域^[4]和工程应用^[5]中的各种复杂问题. 深度学习技术提供了一个强大的信息处理框架^[6], 不仅可以通过算法解析数据, 而且还能够挖掘数据中潜在的隐藏信息, 建立相应的代理模型, 实现对特定科学或工程问题的推理预测.

由神经网络的函数通用近似定理^[7-8]可知, 经过充分训练的深度神经网络模型可以作为非线性偏微分方程的一种求解器. 与有限元、有限差分 and 有限体积等传统数值求解方法不同, 基于深度学习的偏微分方程求解器是一种无网格化方法, 在求解物理和工程问题时无需网格^[9], 略去了离散化过程. 偏微分方程中的导数项可以通过基于链式求导法则的自动微分技术 (automatic differentiation, AD) 获得, 省时省力. 值得一提的是, 在智能科学计算领域中, AD^[10-11]扮演着不可或缺的重要角色. AD 可以高效地计算神经网络输出关于输入的导数, 为损失函数的构建和反向传播过程提供了关键的技术支撑^[5].

目前已经提出并发展了多种基于深度学习技术

的偏微分方程求解方法. 一些学者^[12-13] 基于高斯过程回归的方法建立了线性算子之间的函数映射, 并在经典的基准问题中进行了有效验证. 在后续的研究工作中, Raissi 等^[14] 通过对先前工作^[13] 的改进, 进一步实现了高斯过程对非线性算子的映射. 但是, 对于强非线性问题, 高斯过程模型的预测精度较低且鲁棒性较差. Khoo 等^[15] 基于深度卷积神经网络构造了一种新型的参数偏微分方程求解器, 该求解器可以有效地避免维度灾难的发生. Zhu 等^[16] 通过将偏微分方程嵌入到卷积神经网络的损失函数中, 提出了一个求解偏微分方程的代理模型和量化不确定性的方法. 该方法最大的特点是无监督学习, 在神经网络训练之前不需要预先获得偏微分方程解的信息. 虽然卷积神经网络在图像识别等领域^[9] 表现出色, 但是它们针对结构化数据的特点也限制了其在非结构化数据中的应用. 受到前人工作^[17] 的启发, Raissi 等^[18] 利用神经网络强大的非线性近似能力和 AD, 正式提出了物理信息神经网络 (physics-informed neural networks, PINN). PINN 的基本原理是神经网络的函数通用近似定理, 核心思想是将偏微分方程的残差形式嵌入到神经网络的损失函数中, 进而约束模型训练参数的求解空间, 指导神经网络的输出近似偏微分方程的解. 与传统的深度学习方法不同, PINN 嵌入了物理机理等先验知识, 可以实现数据与物理机理的深度融合, 因此能够更迅速

准确地解决实际遇到的科学和工程问题.

根据损失函数不同的构造方式, PINN 可以分为连续时间模型和离散时间模型, 离散时间模型也被称为基于龙格库塔法的物理信息神经网络模型^[18] (physics-informed neural networks based on Runge-Kutta method, PINN-RK). 然而, 在智能科学计算领域中, 学者们目前主要关注于 PINN 的连续时间模型^[19-25]. 相比于连续时间模型, PINN-RK 的实施步骤较为繁琐, 因此导致对 PINN-RK 的研究较少. PINN-RK 是深度学习技术与龙格库塔方法相结合的产物, 具有非常出色的稳定性和较高的求解精度. 但是, 对于时变偏微分方程而言, 由于受到龙格库塔算法本身的限制^[26], PINN-RK 模型仅能实现单步时间预测, 且计算效率较差. 因此, 为了提高 PINN-RK 的计算效率以及实现偏微分方程解的多时间步长预测, 本文在 PINN-RK 框架的基础上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 (self-regressive physics-informed neural network models based on the Runge-Kutta method, SR-PINN-RK). SR-PINN-RK 模型旨在解决多时间步长预测问题, 并通过简化神经网络结构和减少神经网络复杂度来提高模型的计算效率. 因此, 与 PINN-RK 相比, SR-PINN-RK 大幅度缩减了可训练参数数量, 减轻了神经网络在前向传播和反向传播过程中的计算负担, 提高了整个模型的计算效率, 节约了训练的时间成本. 在 SR-PINN-RK 模型的优化过程中, 本文引入了自回归机制, 将神经网络自身的输出值传递到训练集中, 以实现不断动态更新标签数据的目的. 通过这种方式, 偏微分方程的时间步长也随之向前推进, 直至达到所预设的时间节点. 在时间步长的推进更新下, 经过充分训练后的 SR-PINN-RK 模型可以实现偏微分方程解的多时间步长预测. 最后, 为了验证 SR-PINN-RK 的求解精度和计算效率, 本文使用 SR-PINN-RK 模型分别求解了 Allen-Cahn 方程和 Burgers 方程, 并将模型的预测值与文献^[18]中提供的基准解进行了对比.

1 物理信息神经网络模型

1.1 基于龙格库塔法的物理信息神经网络模型

本小节将简要介绍如何使用 PINN-RK 方法求解常见的非线性偏微分方程. 一般情况下, 非线性偏微分方程的通用形式为

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

其中, $u(t, x)$ 表示非线性偏微分方程的潜在解, 下标 t 表示函数 u 对时间的偏导数, $\mathcal{N}[\cdot]$ 表示非线性微分算子, Ω 为 $\mathbb{R}^D$ 的子集.

龙格库塔法是科学计算中常用的数值方法之一, 也是在工程应用中广泛使用的一种高精度单步算法. 龙格库塔法能够通过迭代的方式计算出微分方程的数值解, 具有很高的准确性和稳定性. 通过将 q 阶龙格库塔公式^[26] 应用到非线性偏微分方程 (1) 可以得到

$$u^{n+c_i} = u^n - \Delta t \sum_{j=1}^{q+1} a_{ij} \mathcal{N}[u^{n+c_j}], \quad i = 1, 2, \dots, q, q+1 \quad (2)$$

其中, q 为龙格库塔阶数, a_{ij} 为龙格库塔公式中的系数, Δt 为时间步长, c_i 和 c_j 分别表示龙格库塔节点, u^{n+c_i} 和 u^{n+c_j} 为龙格库塔采样值

$$u^{n+c_j}(x) = u(t^n + c_j \Delta t, x), \quad j = 1, 2, \dots, q, q+1 \quad (3)$$

龙格库塔公式有两种常见的形式, 分别为显式格式和隐式格式. 由于隐式格式具有更好的稳定性, 因此在本文后续的计算中, 神经网络的损失函数均采用隐式龙格库塔公式构造. 为了方便书写表达, 式 (2) 可以被转化为以下形式

$$u^n = u_i^n, \quad i = 1, 2, \dots, q, q+1 \quad (4)$$

其中

$$u_i^n := u^{n+c_i} + \Delta t \sum_{j=1}^{q+1} a_{ij} \mathcal{N}[u^{n+c_j}], \quad i = 1, 2, \dots, q, q+1 \quad (5)$$

在 PINN-RK 中, 神经网络输出层的神经元数量等于龙格库塔阶数 $q+1$, 具体形式如下式所示

$$u^{n+c_i} = [u^{n+c_1}(x), u^{n+c_2}(x), \dots, u^{n+c_q}(x), u^{n+1}(x)] \quad (6)$$

其中, $u^n(x)$ 表示非线性偏微分方程在当前时刻的解, $u^{n+1}(x)$ 表示非线性偏微分方程下一时刻的解.

PINN-RK 模型通过将式 (5) 作用于待求解的非线性偏微分方程, 可以构造出特定的神经网络损失函数. PINN-RK 以最小化损失函数为目标, 不断更新神经网络的权重和偏差, 最终形成非线性偏微分方程的代理模型. 与连续时间模型不同, PINN-RK 的输入量仅为空间坐标 x , 不包含任何时间量 t . 另

外, PINN-RK 的输出值为向量形式, 由 q 阶龙格库塔采样值和下一时刻偏微分方程的解组成, 如式 (7) 所示

$$\mathbf{u}_{q_i}^n = [u_1^n(x), u_2^n(x), \dots, u_q^n(x), u_{q+1}^n(x)]^T \quad (7)$$

1.2 基于龙格库塔法的自回归物理信息神经网络模型

PINN-RK 是一种用于求解非线性偏微分方程的神经网络模型, 该模型可以通过当前时刻偏微分方程的解准确地推测出一定时间步长后的解. 尽管这种方法具有高度的灵活性和适应性, 但需要较大的神经网络架构, 包括较深的隐藏层数和较多的神经元个数, 因此需要占用大量的计算资源和显存空间. 另外, 受到龙格库塔法是单步算法的限制, PINN-RK 也仅能输出单个时间步长的解, 无法直接对未来多个时间步长的解进行预测. 为此, 本文利用自回归的思想, 优化了 PINN-RK 的训练流程和网络结构, 提高了模型的计算效率, 缩短了训练时间, 并实现了多时间步长预测的目的.

图 1 展示了 SR-PINN-RK 模型的结构示意图. 在该模型中, 除了初始时刻的标签数据由用户给定外, 其余训练标签均由神经网络模型自身提供, 因此该模型被称为自回归模型. SR-PINN-RK 改进了 PINN-RK 的训练流程, 通过将神经网络当前时刻的输出值作为下一时刻的初始条件加入到训练集中, 不断动态更新训练集, 实现了自回归地向前推进时间步长, 最终达到预测多时间步长解的目的. 与 PINN-RK 相比, SR-PINN-RK 仅需较浅的网络层数和少量的神经元个数就能够实现对非线性偏微分方程的近似, 避免了神经网络深度增加所带来的问题, 如梯度消失和梯度爆炸等, 提高了模型的鲁棒性. 与 PINN 连续时间模型相比, SR-PINN-RK 模型在处理时间变量时采用了不同的策略. PINN 连续时间模型未设定明确的时间间隔或时间步长, 将时间视为连

续的变量. 而在 SR-PINN-RK 模型中, 时间被划分为离散的间隔, 模型通过在这些离散的时间点上进行计算来描述系统的演化过程. 这种离散时间的处理方式更符合实际应用中的情况.

SR-PINN-RK 模型的训练流程如图 2 所示. 在 SR-PINN-RK 模型训练过程中, 神经网络首先使用初始训练数据集进行训练, 这些数据集主要由非线性偏微分方程的初始条件和边界条件组成. 为了尽可能地减小 SR-PINN-RK 的优化误差, 本文采用了 Adam 算法^[27] 和 L-BFGS 算法^[28] 相结合的训练方式. 神经网络经过不同算法的双重优化后, 当迭代次数达到所设定的最大迭代次数或损失函数低于所设定的阈值时, SR-PINN-RK 模型将进入自回归阶段. 在 SR-PINN-RK 模型中, 神经网络通过对初始训练集的学习, 已经初步完成了对数据集的特征提取工作. 使用这些特征作为后续时间步长训练任务的起点, 神经网络能够更好地进行优化训练, 实现更快的收敛速度, 降低所需的标签数据数量和训练时间, 提高模型计算效率.

在自回归训练阶段, SR-PINN-RK 模型展现出了独特的能力. 神经网络的当前输出被用作下一个时间步长的输入, 形成了一个自回归的时间序列训练过程. 对于 SR-PINN-RK 模型来说, 这种自回归机制尤为关键, 因为它允许模型在一系列的时间步长中逐步学习和预测. 每一个时间步长的输出不仅是对当前输入的处理结果, 还是下一个时间步长的训练集. 随着 SR-PINN-RK 模型不断地更新训练集, 偏微分方程的时间步长也相应地向前推进. 每一次训练集的更新都意味着模型在时间序列上向前迈进了一步, 从而掌握了更多关于系统动态的信息. 在达到预设的时间节点之前, SR-PINN-RK 模型会不断地重新加载训练集, 并重复之前的训练过程. 这种迭代

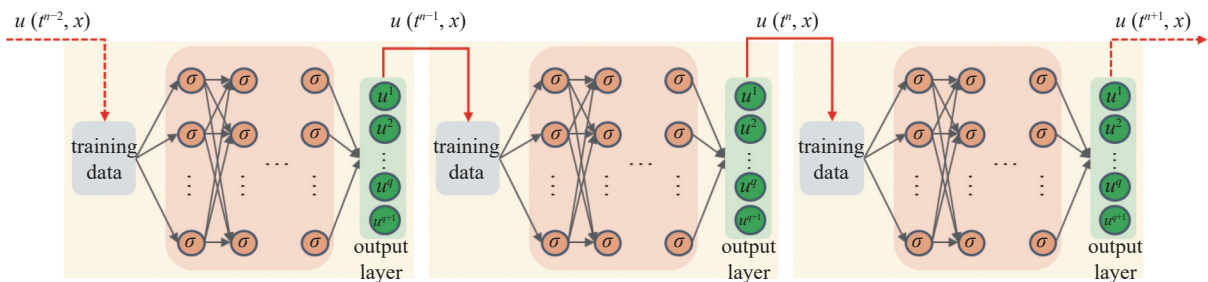


图 1 SR-PINN-RK 模型结构示意图

Fig. 1 Schematic diagram of the structure of the SR-PINN-RK model

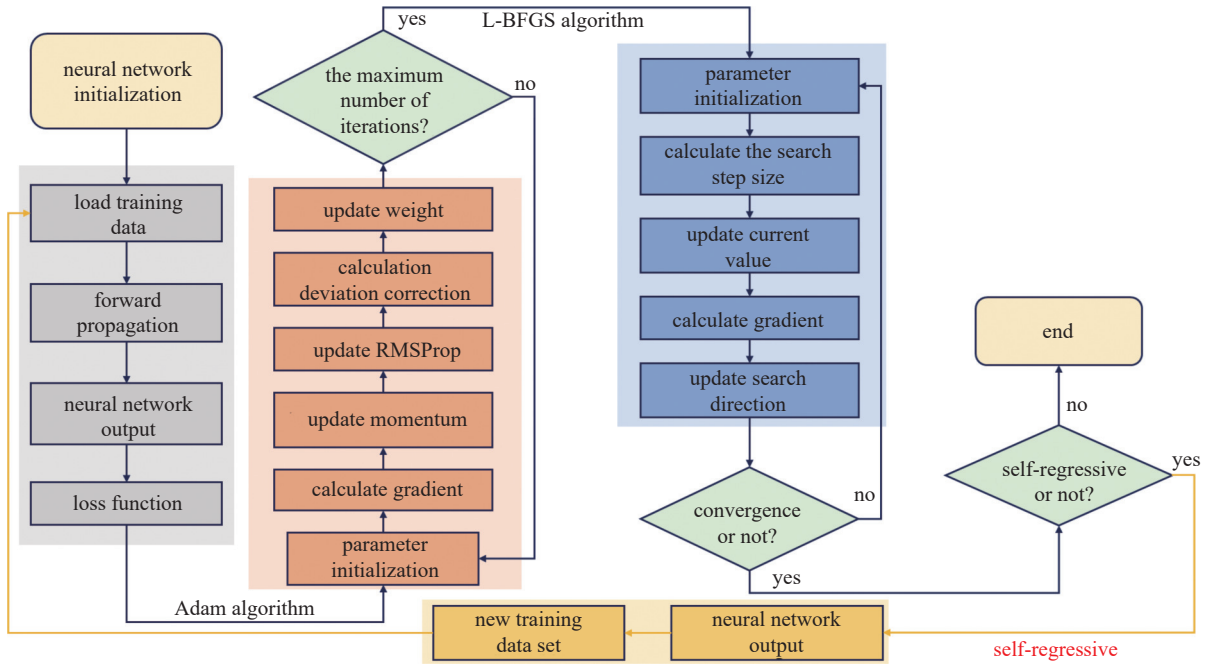


图 2 SR-PINN-RK 训练流程图

Fig. 2 The flowchart illustrating the training procedure of the SR-PINN-RK model

式的训练方式有助于模型逐步积累经验和知识, 从而不断提高其预测能力.

经过充分训练后, SR-PINN-RK 模型可以同时输出不同时间步长的解. 与 PINN-RK 模型相比, SR-PINN-RK 模型通过嵌入自回归机制, 不仅可以最大化地利用神经网络训练初期所积累的知识, 缩短在后续时间步长上的训练时间, 提高模型的训练效率, 而且还能具备多时间步长预测的能力.

2 数值实验

2.1 Allen-Cahn 方程

2.1.1 问题描述

Allen-Cahn 方程^[29]是描述相场模型的一个经典偏微分方程, 该方程在材料科学和生物学等领域被广泛应用. 通过对 Allen-Cahn 方程的分析和求解, 可以研究相变的动力学和热力学特性, 预测材料的形态和微观结构. Allen-Cahn 方程的研究对于深入理解其中的物理现象和开展相关领域的科研工作具有重要意义. 本文将使用 SR-PINN-RK 方法对 Allen-Cahn 方程进行求解, 并与文献 [18] 中提供的基准解进行对比.

Allen-Cahn 方程及其初边界条件如下所示

$$\left. \begin{aligned} u_t - 0.0001u_{xx} + 5u^3 - 5u &= 0, \quad x \in [-1, 1], \quad t \in [0, 1] \\ u(0, x) &= x^2 \cos(\pi x) \\ u(t, -1) &= u(t, 1) \\ u_x(t, -1) &= u_x(t, 1) \end{aligned} \right\} \quad (8)$$

其中, 下标 t 表示函数 u 对时间的偏导数, 下标 x 和 xx 分别表示函数 u 对空间坐标的 1 阶和 2 阶偏导数.

通过将式 (5) 应用于 Allen-Cahn 方程, 可以得到下列损失函数. SR-PINN-RK 模型损失函数形式与 PINN-RK 相同, 如式 (9)~式 (11) 所示. 损失函数采用误差平方和 (sum of squared errors, SSE) 形式, 由周期性边界条件 SSE_b 和物理约束 SSE_n 两部分组成

$$SSE = SSE_n + SSE_b \quad (9)$$

$$\left. \begin{aligned} SSE_n &= \sum_{j=1}^{q+1} \sum_{i=1}^{N_n} \left| u^{n+c_j, i} + \Delta t \sum_{j=1}^{q+1} a_{ij} [PDE] - \hat{u}^{n, i} \right|^2 \\ PDE &= -0.0001u_{xx}^{n+c_j, i} + 5(u^{n+c_j, i})^3 - 5u^{n+c_j, i} \end{aligned} \right\} \quad (10)$$

$$\begin{aligned} SSE_b &= \sum_{j=1}^{q+1} |u^{n+c_j}(-1) - u^{n+c_j}|^2 + \\ &\quad \sum_{j=1}^{q+1} |u_x^{n+c_j}(-1) - u_x^{n+c_j}|^2 \end{aligned} \quad (11)$$

其中, $\hat{u}^{n, i}$ 表示训练集中的标签数据, $u^{n+c_j, i}$ 表示 SR-

PINN-RK 的输出值, i 表示不同的空间采样点, 其取值范围由采样点数量决定, j 表示龙格库塔阶数。

2.1.2 神经网络参数配置

SR-PINN-RK 模型中的训练数据并非固定不变的, 自回归机制的嵌入可以实现标签数据的动态更新。在保持训练数据量相同的情况下, 在单个训练周期内, 与 PINN-RK 模型需要一次性接受全部数据量不同, SR-PINN-RK 模型只需少量数据即可进行训练, 并能够在后续训练过程中通过动态更新数据集的方式实现标签数据的分批式投入, 减少单次训练数据的大小。对于较小的训练数据集, 采用较小的神经网络结构不仅可以避免过拟合, 提高泛化性能, 而且还可以减少训练时间, 提高训练效率。因此, 与 PINN-RK 相比, SR-PINN-RK 模型中的神经网络结构更加简洁。

在实际的训练过程中, 标签数据对于模型的预测性能具有显著影响。首先, 标签数据对于任何监督学习模型来说都是至关重要的, 因为它们为模型提供了学习和预测的依据。因此, 随着标签数据数量的增加, 两个模型的预测精度均有所提高。相反, 随着标签数据量的减少, 两个模型的预测性能均有不同程度的下降。相较于 PINN-RK 模型, SR-PINN-RK 模型凭借其独特的优化算法, 可以从少量数据中提取出有效的特征信息, 并给出相对准确的预测结果。因此当标签数据数量有限时, SR-PINN-RK 模型表现出更强的鲁棒性和适应性。

在对 Allen-Cahn 方程求解时, SR-PINN-RK 模型中神经网络隐藏层数量设置为 3, 每个隐藏层设置 50 个神经元, 龙格库塔阶数为 50, 总训练参数为 7750 个。而在 PINN-RK 模型^[18]中, 神经网络为 4 个隐藏层, 每个隐藏层则有 200 个神经元, 龙格库塔阶数为 100, 总训练参数高达 121400 个。可以看到, 由于 SR-PINN-RK 模型的网络结构更加简洁, 因此可训练参数仅为 PINN-RK 的 6%, 在保证性能的同时大幅度缩减了神经网络的计算量, 显著提升了模型的训练效率。

神经网络强大的非线性拟合能力主要得益于神经元中的非线性激活函数。常见的激活函数主要有 ReLU 激活函数^[30]、Tanh 激活函数^[31]和 Sigmoid 激活函数^[32]等。由于 Tanh 激活函数具有无限可微的特点, 因此文中所有隐藏层均采用 Tanh 激活函数。此外, 为了不约束神经网络输出值的范围, 输出层采用

线性激活函数。SR-PINN-RK 权重初始化采用 Glorot 正态分布初始化^[33], 网络结构为全连接神经网络。

在 SR-PINN-RK 的训练初期阶段, 神经网络损失值通常较高, 为了保证后续训练集的精度, 设置 Adam 算法首次迭代次数为 5000 代, 学习率为 0.001, L-BFGS 算法最大迭代次数为 5000 代。在 SR-PINN-RK 的后续训练区间内, 设置 Adam 迭代次数为 1000 代, 学习率保持不变, L-BFGS 算法最大迭代次数为 1700 代。L-BFGS 算法^[28]的优点是在较少的迭代次数内收敛, 且无需手动设置学习率。但是, L-BFGS 算法在非凸优化问题中容易陷入局部极小值。相比之下, Adam 算法是一种自适应学习率算法^[27], 自适应学习率可以帮助算法在参数空间内跳出局部极小值, 而动量可以使算法在梯度方向上保持一定的惯性, 从而更容易找到全局最优解。因此, 本文采用 Adam 算法和 L-BFGS 算法相结合的训练方式, 旨在先借助 Adam 算法逼近全局最优点, 再利用 L-BFGS 算法加快神经网络的收敛速度, 减少 SR-PINN-RK 的优化误差, 提高模型预测精度。

2.2 Burgers 方程

2.2.1 问题描述

Burgers 方程^[34]是一种经典的非线性偏微分方程, 可以将其视为 N-S 方程的一种简化模型, 常被用于模拟激波传播和反射现象。虽然 Burgers 方程不存在解析解, 但是通过数值方法可以获得 Burgers 方程的数值解。本文将使用 SR-PINN-RK 方法对 Burgers 方程进行求解, 并与文献^[18]中提供的基准解进行对比。

Burgers 方程及其初边界条件为

$$\left. \begin{aligned} u_t + uu_x - (0.01/\pi)u_{xx} &= 0, \quad x \in [-1, 1], \quad t \in [0, 1] \\ u(0, x) &= -\sin(\pi x) \\ u(t, -1) = u(t, 1) &= 0 \end{aligned} \right\} \quad (12)$$

将式 (5) 应用于 Burgers 方程, 可以得到下列 SSE 损失函数。SSE 损失函数由狄利克雷边界条件 SSE_b 和物理约束 SSE_n 两部分组成

$$SSE = SSE_n + SSE_b \quad (13)$$

$$\left. \begin{aligned} SSE_n &= \sum_{j=1}^{q+1} \sum_{i=1}^{N_n} \left| u^{n+c_j, i} + \Delta t \sum_{j=1}^{q+1} a_{ij} [PDE] - \hat{u}^{n, i} \right|^2 \\ PDE &= u^{n+c_j, i} u_x^{n+c_j, i} - (0.01/\pi) u_{xx}^{n+c_j, i} \end{aligned} \right\} \quad (14)$$

$$SSE_b = \sum_{i=1}^{q+1} \left[|u^{n+c_i}(-1)|^2 + |u^{n+c_i}(1)|^2 \right] \quad (15)$$

其中, $\hat{u}^{n,i}$ 表示训练集中的标签数据, $u^{n+c_j,i}$ 表示 SR-PINN-RK 模型的输出值, i 表示不同的空间采样点, 取值范围由采样点数量决定, j 表示龙格库塔阶数.

2.2.2 神经网络参数配置

在求解 Burgers 方程时, SR-PINN-RK 模型的参数设置与求解 Allen-Cahn 方程时的设置相同. 而在 PINN-RK 模型^[18]中, 神经网络拥有 4 个隐藏层, 每个隐藏层有 50 个神经元, 龙格库塔阶数为 500, 总训练参数为 32 800 个. 可以看到, SR-PINN-RK 模型的可训练参数仅为 PINN-RK 的 20%, 减少了优化学习所需的计算量, 提高了计算效率.

在 SR-PINN-RK 训练的早期阶段, 设置 Adam 首次迭代次数为 1000 代, 学习率为 0.001, L-BFGS 算法的最大迭代次数为 1700 代. 在 SR-PINN-RK 的后续训练区间内, Adam 迭代次数和学习率保持不变, L-BFGS 算法的最大迭代次数降为 350 代. 文中所有的计算代码均基于 TensorFlow2.6 版本的 Python 库开发, 并在 GeForce 730 显卡上完成运算. 本文的主要目的是证明所提出模型的可行性, 而非追求最佳的神经网络结构和训练参数. 因此, 文中所采用的神经网络结构与训练参数配置均根据过往经验进行设定.

3 结果与讨论

3.1 Allen-Cahn 方程

本小节将应用 SR-PINN-RK 模型对 Allen-Cahn 方程进行求解, 并对损失函数变化曲线和模型预测结果展开深入讨论. 在 SR-PINN-RK 学习 Allen-Cahn 方程的过程中, 其损失函数变化曲线如图 3 所示. 从整体趋势看, 随着迭代次数的不断增加, 损失函数值持续减小并逐渐逼近于 0. 图中红色方框表示自回归位置, 即训练集更新的时刻. 在 SR-PINN-RK 训练期间, 共经历了 7 次自回归过程. 从初始时刻起, 时间步长共推进 $8\Delta t$, 整个训练流程被划分为 8 个训练区间. 每个训练区间又可分为两个阶段, 第 1 阶段采用 Adam 训练算法, 如图中绿色星号所示, Adam 训练算法起始位置与自回归位置相同. 第 2 阶段采用 L-BFGS 算法, 如图中橘色圆形所示.

从图 3 中可以观察到, 神经网络损失函数在训练的初始阶段出现了小幅度的振荡, 这主要是由于 Adam 算法自身的特性所导致的. 为了更好地适应不同参数的更新情况, Adam 算法会在训练过程中根据历史梯度信息自动调整学习率. 但当学习率过大时, 更新的步长可能会过大, 从而导致算法在神经网络的训练参数空间中跳过最优解, 引起损失函数值的突然增加. 另外, 从图 3 中损失函数的局部放大图可以看到, 每当神经网络经历自回归位置时, 损失函数值都会出现阶跃现象. 造成这种现象的原因主要是

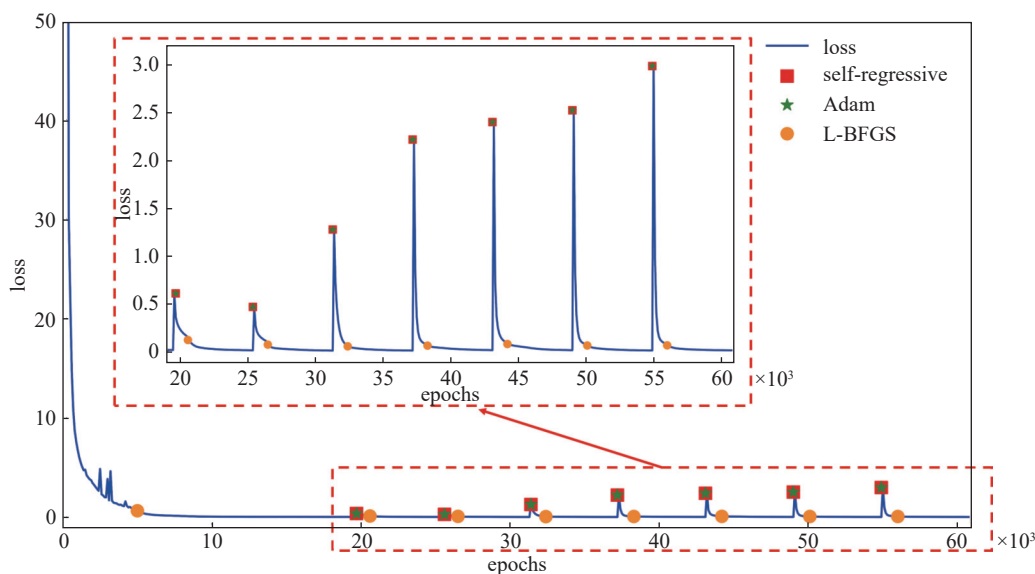


图 3 SR-PINN-RK 求解 Allen-Cahn 方程的损失函数变化曲线

Fig. 3 The change in loss function for solving the Allen-Cahn equation using the SR-PINN-RK model

由于训练集的突然更新和预测目标的改变所导致的. 在 SR-PINN-RK 的训练过程中, 除了第 1 个训练区间中的训练集使用初始条件外, 后续训练阶段所采用的训练数据集均由神经网络自身产生, 即自回归过程. 由于 SR-PINN-RK 的预测结果与真实结果之间存在微小误差, 使后续的训练数据标签也会存在误差. 所以, 随着自回归次数的不断增加, 损失函数的增加幅度也在不断升高. 但是, 从损失函数的局部放大图中可以看到, 在 Adam 算法和 L-BFGS 算法的不断优化下, 在每个训练区间末尾, 损失值最终都会趋向于 0. 另外, 虽然训练数据的更新造成了损失函数的升高, 但是从全局看并未对整个训练过程产生较大影响. 这是因为不同训练集背后所隐藏的物理规律相同, 全部由 Allen-Cahn 方程控制. 面对后续时间的推测任务, SR-PINN-RK 模型会在相似但不同的数据集上对神经网络参数进行微调, 以提

高模型的预测准确性和加快收敛速度. 相较于 PINN-RK 模型, SR-PINN-RK 模型可以更高效地利用模型训练初期所学习到的特征来解决新的问题和任务, 具有更好的训练效率.

图 4 中展示了 SR-PINN-RK 模型对 Allen-Cahn 方程的预测解与基准解在不同时刻的对比结果. 云图展示了 Allen-Cahn 方程在时空域内解的分布情况. 其中, 绿色实线表示 Allen-Cahn 方程的初始条件, 即初始训练集; 红色虚线表示 SR-PINN-RK 对当前时刻的预测解; 黄色实线表示上一个时刻 SR-PINN-RK 的预测解, 也即当前时刻神经网络的训练集. 线图展示了模型预测解与基准解在不同时刻的相符程度, 其中红色虚线表示模型的预测解, 蓝色实线表示由数值方法得出的基准解^[18].

由图 4 可以看到, 经过训练后, SR-PINN-RK 模型可以同时输出多个时间步长的解. 相比之下, PINN-

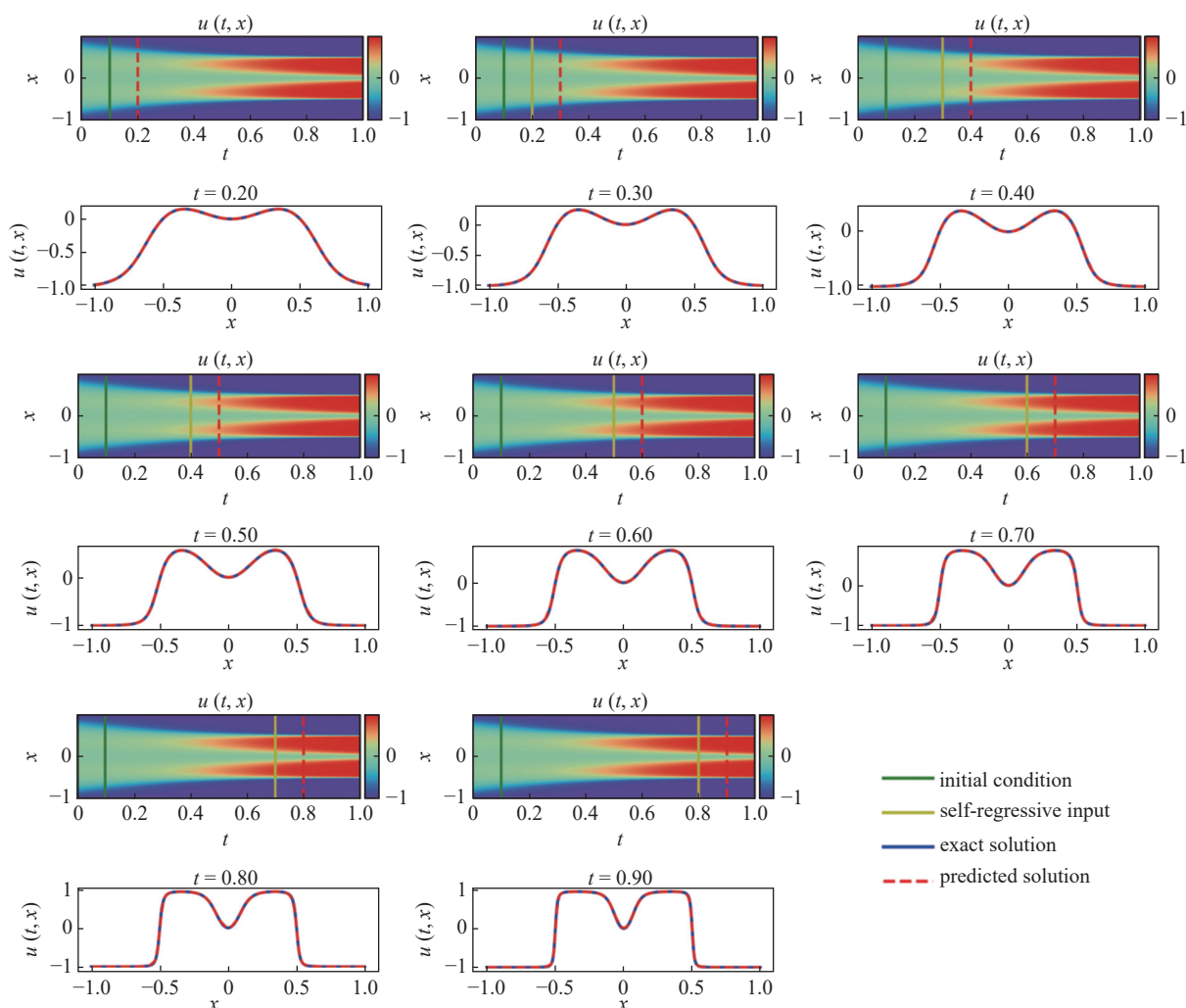


图 4 Allen-Cahn 方程 SR-PINN-RK 模型预测解与基准解对比

Fig. 4 Comparison between the SR-PINN-RK solution and the benchmark solution of the Allen-Cahn equation

RK 模型^[18] 仅能预测单步时间解, 无法随着训练的迭代更新向前推进时间步长. 另外, 尽管 SR-PINN-RK 模型在结构上更为简洁, 但它的预测精度并未受到明显影响. SR-PINN-RK 模型所获得的预测解与精确解之间具有很好的一致性, 可以很好地捕获 Allen-Cahn 方程在求解域中心位置不同时刻的变化情况. 并且, 随着时间步长的不断推进, SR-PINN-RK 的预测解并未发生偏移, 仍然可以根据神经网络上一时刻的输出值推测当前时刻的解, 这证明了文中所提出模型的有效性. 同时, 与 PINN-RK 模型相比, 在不牺牲模型预测精度的前提下, SR-PINN-RK 模型具有更少的隐藏层数、更少的神经元个数以及更低的龙格库塔阶数, 节省了大量计算资源和训练时间, 具有更高的计算效率.

为了更好地评估 SR-PINN-RK 模型在求解 Allen-Cahn 方程时的预测精度, 使用下式计算了模型在不同自回归次数下的 L2 相对误差

$$\text{relative error} = \frac{1}{N} \sum_{i=1}^N \frac{\|\hat{u}_i - u_i\|_2}{\|u_i\|_2} \quad (16)$$

其中, $\hat{u}_i$ 表示 SR-PINN-RK 模型的预测值, u_i 表示精确值. 在深度学习领域, L_2 相对误差是衡量模型预测解与基准解差异的一种通用指标, 对 L_2 相对误差的可视化, 可直观地展现神经网络相对误差的变化趋势, 如图 5 所示.

可以看出, 尽管随着自回归次数的增加和时间步长的推进, SR-PINN-RK 预测解与基准解之间的

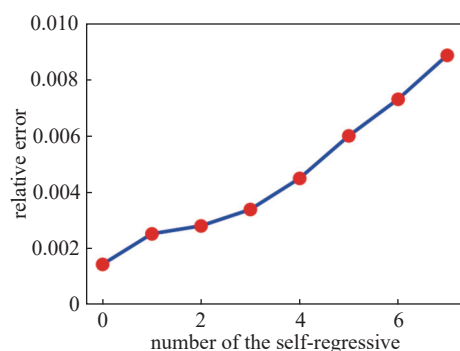


图 5 SR-PINN-RK 求解 Allen-Cahn 方程的相对误差图

Fig. 5 The relative error plot for solving the Allen-Cahn equation using the SR-PINN-RK model

相对误差在不断增加, 但最大误差仍保持在低于 0.01 的水平. 值得注意的是, 自回归训练过程会导致误差的累积和传递. 由于每个时间步长的输出都作为下一个时间步长的输入, 任何小的预测误差都可能被放大并传递到后续的时间步长中. 当误差累计较大时, 可以通过正则化方法或优化算法等措施来减轻这种误差累积的影响.

3.2 Burgers 方程

本小节将使用 SR-PINN-RK 模型对 Burgers 方程进行求解, 并对损失函数变化曲线和神经网络预测结果展开深入讨论. 在 SR-PINN-RK 学习 Burgers 方程的过程中, 其损失函数变化曲线如图 6 所示. 图中红色方框表示自回归位置, 绿色星号表示 Adam 训练算法起始位置, 橘色圆形表示 L-BFGS 算法起始位置. 从图 6 中可以发现, 神经网络的损失函数整

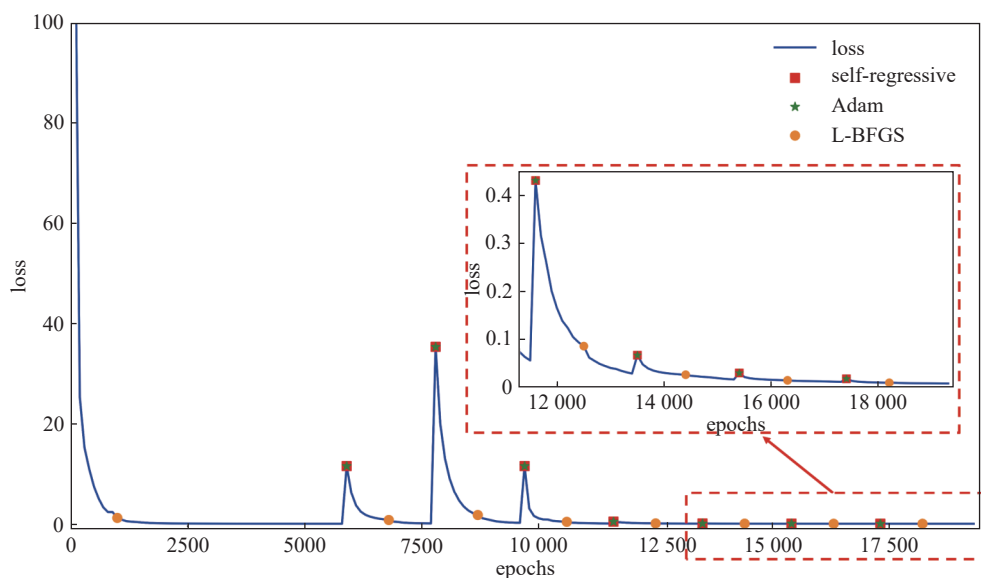


图 6 SR-PINN-RK 求解 Burgers 方程的损失函数变化曲线

Fig. 6 The change in loss function for solving the Burgers equation using the SR-PINN-RK model

体上呈现出不断降低的趋势,但是在自回归初期却出现了大幅度的振荡.此外,进一步观察可以看到,SR-PINN-RK 进入自回归阶段时,损失函数呈现出先增大后减小的变化趋势.

上一小节已经分析过,损失函数的突然增大主要是由于训练数据的更新和标签数据中存在误差.另外,Burgers 方程的自身特性也是造成损失函数幅值增大的原因之一.在自回归初始阶段,Burgers 方程的解变化剧烈,且由连续分布向不连续分布过渡,导致 SR-PINN-RK 模型的训练难度增加,损失函数值变大.但是,在 Adam 算法和 L-BFGS 算法的双重优化下,损失函数仍可收敛于 0.虽然 Burgers 方程的解存在不连续性,但是随着时间步长的不断推进,Burgers 方程的解不再发生变化.因此,在 SR-PINN-RK 模型训练的后期阶段,不同训练数据集之间只发生了微小的改变,神经网络训练难度下降,导致损失

函数随着回归次数的增加呈现出不断减小的趋势.

SR-PINN-RK 对 Burgers 方程在不同时刻的预测解与基准解之间的对比如图 7 所示.其中,绿色实线表示 Burgers 方程的初始条件,即初始训练集;红色虚线表示 SR-PINN-RK 对当前时刻的预测解;黄色实线表示上一个时刻 SR-PINN-RK 的预测解,也即当前时刻神经网络的训练集.线图展示了模型预测解与基准解在不同时刻的相符程度,其中红色虚线表示模型的预测解,蓝色实线表示由数值方法得出的基准解^[18].

PINN-RK 模型仅能预测单个时间步长的解,无法处理连续解和非连续解共存的情形.在自回归机制的作用下,SR-PINN-RK 模型实现了连续解与不连续解的同时预测.具体说来,通过观察图 7 可看到,即使 Burgers 方程中存在连续解与不连续解,SR-PINN-RK 模型仍然可以同时输出多个时间步长的

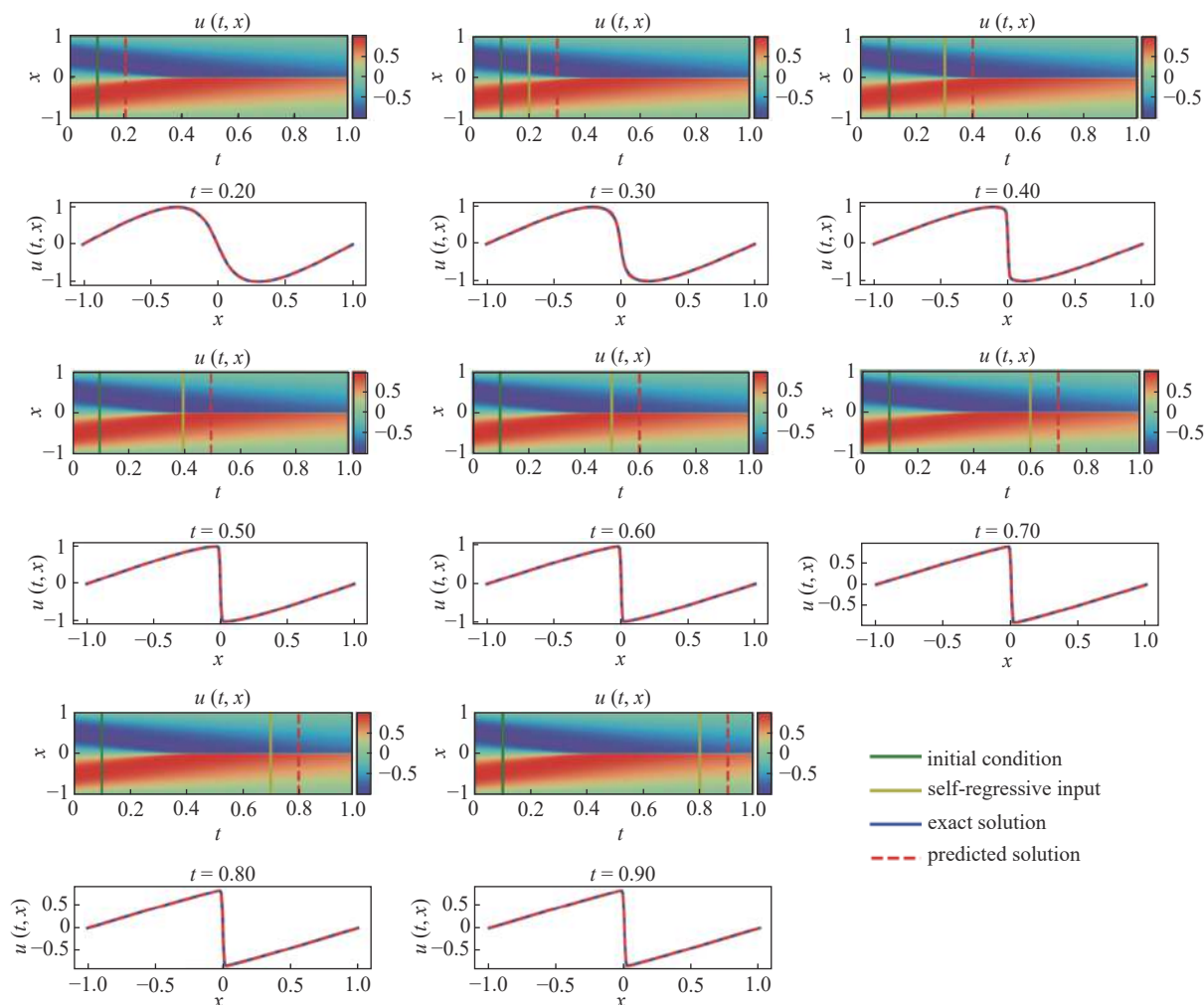


图 7 Burgers 方程 SR-PINN-RK 预测解与基准解对比

Fig. 7 Comparison between the SR-PINN-RK solution and the benchmark solution of the Burgers equation

预测结果,且预测解与精确解高度契合,两者之间没有明显的差异.此外,在连续解到不连续解的过渡中,SR-PINN-RK 模型仍能准确捕获解的动态特性,模型性能并未受到影响.这些结果进一步证明了 SR-PINN-RK 模型的高效性和稳定性,可以在解决复杂问题时发挥出良好的效果.值得一提的是,在 SR-PINN-RK 模型中,自回归机制的嵌入极大地拓宽了模型的应用范围,对于处理更加复杂的偏微分方程具有非常重要的意义.另外,由图 7 可得,在 $t = 0.5$ 之后, Burgers 方程的解并未发生显著的变化,这也再次解释了图 6 中神经网络损失函数逐渐减小的现象.

SR-PINN-RK 求解 Burgers 方程的预测解与精确解之间的相对误差变化曲线如图 8 所示.总体而言,相对误差随着自回归次数的增加呈现出先增大后减小的趋势,与图 6 损失函数变化趋势相同.此外,神经网络预测解与精确解之间的最大相对误差约为 0.008,这表明该模型具有很高的预测精度.尽管 SR-PINN-RK 的可训练参数更少,网络结构更简洁,但其预测精度并未降低.与 PINN-RK^[18] 相比,两模型产生的相对误差具有相同数量级.

从图 5 和图 8 中可以看到,自回归训练过程存在误差累计现象,但是否存在临界值或收敛值则与所求解的偏微分方程相关.求解 Allen-Cahn 方程时,随着自回归次数的增加, L_2 相对误差一直呈现上升趋势,并不存在临界值;但在求解 Burgers 方程时,随着自回归次数的增加, L_2 相对误差则呈现出先上升后下降的趋势,存在一个最大临界值.

至此,已经利用 SR-PINN-RK 模型成功地求解了 Allen-Cahn 方程和 Burgers 方程,验证了模型的有效性.为了更好地理解自回归机制对模型性能提

升所带来的影响,将其与迁移学习进行比较.迁移学习^[35]是指将神经网络在一个任务中学到的知识迁移到不同但相关的任务中.通过这种学习方式,可以有效地减少神经网络在新任务中所需的标签数据和训练时间,提高模型训练效率.迁移学习首先采用预训练模型从数据集提取有用的特征表示,使后续任务可更好地利用这些特征进行训练.在 SR-PINN-RK 模型中,经过首个训练阶段后,此时的神经网络如同迁移学习的预训练模型,自回归机制的嵌入可以使 SR-PINN-RK 模型在面对后续任务时能够直接利用预训练模型的网络参数作为训练起点.相比从头开始训练学习,不仅可以更快地获得理想结果,并且还可以减少对数据和计算资源的占用.在后续时间的推测任务中,SR-PINN-RK 模型会在相似但不同的数据集上对神经网络参数进行微调,提高了模型的预测准确性,降低了过拟合的风险,并且能够以更快的速度收敛到最优解.在自回归机制的作用下,SR-PINN-RK 模型与迁移学习相同,可以更加高效地利用先前的训练模型去解决新问题和任务.

4 结论

针对 PINN-RK 模型仅能预测单个时间步长的问题,受到自回归思想的启发,本文提出了一种 SR-PINN-RK 模型.在 SR-PINN-RK 模型的优化过程中,本文引入了自回归机制,将神经网络自身的输出值传递到训练集中,以实现不断动态更新标签数据的目的.通过这种方式,偏微分方程的时间步长也随之向前推进,直至达到所预设的时间节点.与 PINN-RK 模型相比,SR-PINN-RK 模型具有以下优势:

- (1) 克服了龙格库塔单步算法所带来的局限性,可以实现对偏微分方程解的多时间步长预测;
- (2) 简化了神经网络结构,大幅度缩减了可训练参数数量,提高了神经网络的计算效率,降低了优化训练的时间成本;
- (3) 减少了对训练数据量的依赖性,提高了模型的泛化性能.

为验证 SR-PINN-RK 模型的有效性,本文选择 Allen-Cahn 方程和 Burgers 方程两个经典的基准问题作为测试案例.测试结果表明,定性上,SR-PINN-RK 模型预测解与基准解完全吻合,并无明显差异;定量上,SR-PINN-RK 预测解与基准解之间的最大相对误差仅为 0.009,具有很高的求解精度.因此,本

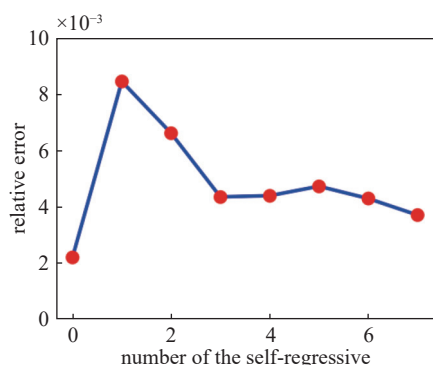


图 8 SR-PINN-RK 求解 Burgers 方程的相对误差图

Fig. 8 The relative error plot for solving the Burgers equation using the SR-PINN-RK model

文所提出的 SR-PINN-RK 模型作为一种新型的偏微分方程求解器, 具有很大的发展潜力。

参 考 文 献

- 1 Yurtsever E, Lambert J, Carballo A, et al. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 2020, 8: 58443-58469
- 2 He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(9): 1904-1916
- 3 Afkanpour A, Adeel S, Bassani H, et al. BERT for long documents: A case study of automated ICD coding. arXiv Preprint, arXiv: 2022, 221102519
- 4 Baker N, Alexander F, Bremer T, et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. USDOE Office of Science (SC), Washington DC, United States, 2019
- 5 Karniadakis GE, Kevrekidis IG, Lu L, et al. Physics-informed machine learning. *Nature Reviews Physics*, 2021, 3(6): 422-440
- 6 Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 2020, 52(1): 477-508
- 7 Lu L, Jin P, Pang G, et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 2021, 3(3): 218-229
- 8 Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 1989, 2(5): 359-366
- 9 Lu L, Meng X, Mao Z, et al. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 2021, 63(1): 208-228
- 10 Margossian CC. A review of automatic differentiation and its efficient implementation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2019, 9(4): e1305
- 11 Baydin AG, Pearlmutter BA, Radul AA, et al. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 2018, 18: 1-43
- 12 Rasmussen CE, Williams CK. Gaussian Processes for Machine Learning. MIT Press Cambridge MA, 2006
- 13 Raissi M, Perdikaris P, Karniadakis GE. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 2017, 348: 683-693
- 14 Raissi M, Perdikaris P, Karniadakis GE. Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 2018, 40(1): A98-A172
- 15 Khoo Y, Lu J, Ying L. Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics*, 2021, 32(3): 421-435
- 16 Zhu Y, Zabaras N, Koutsourelakis PS, et al. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 2019, 394: 56-81
- 17 Dissanayake M, Phan-Thien N. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 1994, 10(3): 195-201
- 18 Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 2019, 378: 686-707
- 19 宋家豪, 曹文博, 张伟伟. FD-PINN: 频域物理信息神经网络. *力学学报*, 2023, 55(5): 1195-1205 (Song Jiahao, Cao Wenbo, Zhang Weiwei. FD-PINN: Frequency domain physics-informed neural network. *Chinese Journal of Theoretical and Applied Mechanics*, 2023, 55(5): 1195-1205 (in Chinese))
- 20 Jagtap AD, Kawaguchi K, Karniadakis GE. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 2020, 404: 109136
- 21 Jin X, Cai S, Li H, et al. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 2021, 426: 109951
- 22 查文舒, 李道伦, 沈路航等. 基于神经网络的偏微分方程求解方法研究综述. *力学学报*, 2022, 54(3): 543-556 (Zha Wenshu, Li Daolun, Shen Luhang, et al. Review of neural network-based methods for solving partial differential equations. *Chinese Journal of Theoretical and Applied Mechanics*, 2022, 54(3): 543-556 (in Chinese))
- 23 Raissi M, Karniadakis GE. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 2018, 357: 125-141
- 24 蒋子超, 江俊扬, 姚清河等. 基于神经网络的差分方程快速求解方法. *力学学报*, 2021, 53(7): 1912-1921 (Jiang Zichao, Jiang Junyang, Yao Qinghe, et al. A fast solver based on deep neural network for difference equation. *Chinese Journal of Theoretical and Applied Mechanics*, 2021, 53(7): 1912-1921 (in Chinese))
- 25 Yuan L, Ni YQ, Deng XY, et al. A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *Journal of Computational Physics*, 2022, 462: 111260
- 26 Iserles A. A First Course in the Numerical Analysis of Differential Equations. Cambridge University Press, 2009
- 27 Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv Preprint, arXiv: 2014, 1412.6980
- 28 Byrd RH, Lu P, Nocedal J, et al. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 1995, 16(5): 1190-1208
- 29 Hussain S, Shah A, Ayub S, et al. An approximate analytical solution of the Allen-Cahn equation using homotopy perturbation method and homotopy analysis method. *Heliyon*, 2019, 5(12): e03060
- 30 Agarap AF. Deep learning using rectified linear units (relu). arXiv Preprint, arXiv: 2018, 1803.08375
- 31 Fan E. Extended tanh-function method and its applications to nonlinear equations. *Physics Letters A*, 2000, 277(4-5): 212-218
- 32 Yin X, Goudriaan J, Lantinga EA, et al. A flexible sigmoid function of determinate growth. *Annals of Botany*, 2003, 91(3): 361-371
- 33 Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks//Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010
- 34 Bakodah H, Al-zaid N, Mirzazadeh M, et al. Decomposition method for solving Burgers' equation with Dirichlet and Neumann boundary conditions. *Optik*, 2017, 130: 1339-1346
- 35 Torrey L, Shavlik J. Transfer learning//Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI Global, 2010: 242-264