

文章编号: 0258-0926(2022)02-0001-08; DOI:10.13832/j.jnpe.2022.02.0001

基于 PINN 深度机器学习技术求解 多维中子学扩散方程

刘 东^{1,2,3}, 罗 琦³, 唐 雷^{1,2}, 安 萍^{1,2}, 杨 帆¹

1. 中国核动力研究设计院核反应堆系统设计技术重点实验室, 成都, 610213; 2. 中核核能软件与数字化反应堆工程技术研究中心, 成都, 610213; 3. 中国核工业集团有限公司科技委, 北京, 100822

摘要: 阐述了基于物理信息指引的神经网络模型 (PINN), 构造深度神经网络作为试函数, 将其代入中子学扩散方程形成残差, 并作为机器学习的加权损失函数, 进而通过深度机器学习技术逼近中子学扩散方程数值解; 针对扩散方程的特点, 提出了特征值方程加速收敛方法、有效增殖系数 (k_{eff}) 高效并行搜索技术、学习样本网格点不均匀分布策略等创新性关键技术, 并对神经网络深度、神经元数量、边界条件损失函数权重等关键参数进行了敏感性分析。验证计算结果表明, 该方法具有良好的精度, 提出的关键技术具有显著的成效, 为中子学扩散方程的数值求解探索出了新的技术途径。

关键词: 深度机器学习; 基于物理信息指引的神经网络模型 (PINN); 中子学扩散方程; 加速收敛; 有效增殖系数 (k_{eff})

中图分类号: TL334 文献标志码: A

Solving Multi-Dimensional Neutron Diffusion Equation Using Deep Machine Learning Technology Based on PINN Model

Liu Dong^{1,2,3}, Luo Qi³, Tang Lei^{1,2}, An Ping^{1,2}, Yang Fan¹

1. Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, Chengdu, 610213, China;
2. CNNC Engineering Research Center of Nuclear Energy Software and Digital Reactor, Chengdu, 610213, China;
3. Committee on Science and Technology of China National Nuclear Corporation, Beijing, 100822, China

Abstract: This paper elaborates the physics-informed neural network model (PINN), constructs a deep neural network as a trial function, substitutes it into the neutron diffusion equation to form a residual, and takes it as the weighted loss function of machine learning, and then approaches the numerical solution of the neutron diffusion equation by deep machine learning technique; According to the characteristics of diffusion equation, this paper puts forward innovative key technologies such as accelerated convergence method of eigenvalue equation, efficient parallel search technology of effective multiplication coefficient (k_{eff}), learning sample grid point uneven distribution strategy, and analyzes the sensitivity of key parameters such as neural network depth, neuron number, boundary condition loss function weight and so on. The verification calculation results show that the method has good accuracy, and the proposed key technology has remarkable results, and explores a new technical approach for the numerical solution of the neutron diffusion equation.

Key words: Deep machine learning, Physics-informed neural networks (PINN), Neutron diffusion equation, Accelerated convergence, Effective multiplication factor (k_{eff})

收稿日期: 2021-12-07; 修回日期: 2022-01-26

基金项目: 第五批国家高层次人才特殊支持计划科技创新领军人才基金项目

作者简介: 刘 东 (1973—), 男, 博士, 研究员, 现主要从事核能数值计算、专业软件开发、数值仿真、人工智能应用方面的研究,

E-mail: liudong73@yahoo.com

0 引言

在核反应堆设计研发领域,当前核反应堆工程堆芯中子学设计计算主要采用基于中子输运理论模型及其散射角度各向同性近似的**扩散模型**。其中,中子扩散模型计算速度较快,并且具有工程设计可接受的精度,在反应堆堆芯方案搜索、燃料管理等实际应用中广泛采用^[1-2]。业内对求解中子扩散模型已经有了较为成熟的数值计算方法,主要包括有限差分法、有限元法等^[3-5]。其总体思路是从能量离散、空间离散、并行加速等方面开展数值计算方法研究与软件研发,并且已在当前实际工程中得到了大量成功应用。

近年来,随着人工智能技术的飞速发展,在反应堆设计、计算领域开展了大量以深度学习技术为代表的智能计算技术的研究,通过数据驱动实现人工神经网络建模的技术已经取得显著的成果^[6]。近年来,利用深度学习技术直接求解复杂偏微分方程的技术正在成为计算物理领域研究热点^[7-10],这些方法不需要预先收集大量的数据作为机器学习的训练集,而是在求解域上产生学习样本,通过控制方程与边界条件形成损失函数,进而调整神经网络参数进行函数逼近实现方程求解,在学习数据集的获得上具有明显优势。基于此,在反应堆中子学扩散计算方法研究上,文献[11]提出了边界条件相关的机器学习模型(BDM)实现了对多维问题求解,但其主要困难在于对边界条件需要事先构造解析关系式表示合理试函数,将边界特性完全“吸收”到神经网络函数形式中,这对于现实工程中的复杂问题,存在巨大挑战。

值得注意的是,一种基于物理信息指引的神经网络模型(PINN)微分方程求解方法^[12],由于其对复杂高阶多维方程可进行正向计算、空间没有几何限制、处理边界条件灵活、计算流程规范强等特点,已在多种复杂微分方程求解方法上取得了较大的突破,越来越受到业界的高度重视。

本文基于PINN模型,对多个典型多维扩散问题进行研究,提出了新颖的针对临界条件下特征值扩散方程加速收敛方法、机器学习环境下有效增殖系数(k_{eff})高效并行搜索技术、几何空间样本网格点不均匀分布策略,并针对神经网络的层数、隐藏单元数量、边界条件损失函数权重等

关键的参数进行系统性的敏感性分析与评价。

1 理论原理

1.1 中子学扩散方程模型

核反应堆工程堆芯中子学设计计算中采用中子扩散模型的方程形式如式(1)所示:

$$\frac{1}{v} \frac{\partial \phi(r, E, t)}{\partial t} = \nabla \cdot D \nabla \phi(r, E, t) - \Sigma_t(r, E) \phi(r, E, t) + x(E) \int_0^\infty \vartheta(E') \Sigma_f(r, E') \phi(r, E', t) dE' + S(r, E, t) + \int_0^\infty \Sigma_s(r, E' \rightarrow E) \phi(r, E', t) dE' \quad (1)$$

式中, $\phi(r, E, t)$ 是 t 时刻在 r 位置处能量为 E 的中子注量率; v 为中子速率; D 为中子扩散系数; ϑ 为每次裂变放出的平均中子数; $x(E)$ 是裂变中子的能谱分布; Σ_t 、 Σ_s 和 Σ_f 分别是总的反应截面、散射截面和裂变截面; $S(r, E, t)$ 为外源。其单能群无外源扩散方程形式^[1]可写为:

$$\frac{1}{Dv} \frac{\partial \phi(r, t)}{\partial t} = \nabla^2 \phi(r, t) + \frac{k_\infty - 1}{L^2} \phi(r, t) \quad (2)$$

式中, L 为扩散长度; k_∞ 为无限介质增殖系数。

当系统处于稳态时,式(2)形式^[1]为:

$$\nabla^2 \phi(r) + \frac{k_\infty/k_{\text{eff}} - 1}{L^2} \phi(r) = 0 \quad (3)$$

当系统处于**临界状态**($k_{\text{eff}}=1$)时,式(3)的形式^[1]为:

$$\nabla^2 \phi(r) + B_g^2 \phi(r) = 0 \quad (4)$$

式中, B_g^2 为系统临界时的几何曲率,与系统的几何特性相关,临界时等于材料曲率。

式(1)~式(4)可描述为统一的两阶偏微分方程形式^[3]:

$$F[\vec{x}, \phi(\vec{x}), \nabla \phi(\vec{x}), \nabla^2 \phi(\vec{x})] = 0 \quad (5)$$

式中, \vec{x} 为扩散方程几何描述空间向量。

1.2 PINN 基本原理与数值计算流程

利用PINN深度机器学习技术求解扩散方程的基本原理与步骤^[12]是:将深度人工神经网络形成的逼近函数作为试函数,代入到中子扩散方程中,从而形成与扩散方程相对应的损失函数;将此方程损失函数以及边界和初始条件代入人工神经网络函数形成的损失函数,加权后作为统一的机器学习损失函数,进行神经网络深度学习;通过不断迭代优化网络权值,减小损失函数值,直至达到足够合理的机器学习精度;此时,该神经网络逼近函数就是扩散方程的数值解。该技术途

径类似求解偏微分方程的变分方法^[13]，不同之处在于选择了规范的深度神经网络函数作为试函数，并利用深度机器学习方法进行神经网络权重的优化，逐步减小损失函数值，不断逼近真实方程的解。

全连接深度神经网络架构见文献 [14]，其第 l 层神经网络函数形式为：

$$N^l(\vec{x}) = f^l(w^l f^{l-1}(w^{l-1} f^{l-2}(\dots f^1(w^1 \vec{x} + b^1) \dots + b^{l-2}) + b^{l-1}) + b^l) \quad (6)$$

令 $N(\vec{x}) = \phi(\vec{x})$ ，将式 (6) 代入式 (5) 中，其残差函数形式为：

$$F_{\text{Loss}} = F[\vec{x}, N(\vec{x}), \nabla N(\vec{x}), \nabla^2 N(\vec{x})] \quad (7)$$

将方程的边界条件值 $K_b(\vec{x}_b^i)$ 与初始值 $K_c(\vec{x}_c^k)$ 代入式 (6) 与式 (7) 共同形成不同权重的机器学习损失函数为：

$$f_{\text{Loss}} = \frac{P_f}{|i|} \sum_i F[\vec{x}^i, N(\vec{x}^i), \nabla N(\vec{x}^i), \nabla^2 N(\vec{x}^i)]^2 + \frac{P_b}{|j|} \sum_j [N(\vec{x}_b^j) - K_b(\vec{x}_b^j)]^2 + \frac{P_c}{|k|} \sum_k [N(\vec{x}_c^k) - K_c(\vec{x}_c^k)]^2 \quad (8)$$

式中， i 为 $\phi(\vec{x})$ 进行机器学习的空间离散点； j 、 k 分别为边界条件和初始条件的空间离散点； P_f 、 P_b 、 P_c 分别为控制方程、边界条件、初始条件的残差权重； \vec{x}_b 是边界条件的几何描述空间向量； \vec{x}_c 是初始条件的几何描述空间向量； f_{Loss} 为控制方程残差函数均方差 $[\sigma_{\text{MSE}, F_{\text{Loss}}}]$ 与边界条件、初始条件代入 $N(\vec{x})$ 形成的均方差 $(\sigma_{\text{MSE}(BC, IC)})$ 按照不同权重构成的机器学习损失函数之和。

机器学习过程是调整式 (6) 中的 w^l 与 b^l ，当 f_{Loss} 趋近于 0 时， $N(\vec{x})$ 趋近于 $\phi(\vec{x})$ 。其计算流程见文献 [12]，实践中设定合理的 ε ，当 f_{Loss} 小于 ε ，停止学习迭代。

2 利用 PINN 求解扩散方程的关键技术

利用 PINN 基本模型求解微分方程具有一定的普适性，但将其直接应用于中子学扩散方程，精度与收敛性上会出现一定的困难，需要针对扩散方程的特点进行改进与完善。

2.1 扩散方程特征向量加速收敛方法

当扩散方程是稳态临界状态时，扩散方程为式 (3)、式 (4) 形式，这在数理方法上属于特征值方程。对同一特征值来说，特征向量是无穷

多个^[1-2]。实验证明，这种特征值方程解的不确定性会对神经网络收敛过程造成大幅度振荡，收敛时间会大幅度增加。同时，神经网络初始值的不同会造成收敛特征向量的不同，而且，在同样的初始值下，因为学习率、批处理值的差异等因素，收敛的特征向量也可能不同，相同的初始值难以重现计算结果。

一种改进的方法是将扩散方程的特殊空间点、多个区域平均值/积分值与预设值的均方差作为 f_{Loss} 的额外加权因素，采用与初始条件、边界条件相似的处理方法进行加权处理，将大幅度提高机器学习收敛效率，即便采用不同的神经网络初始值参数也会收敛到相近的结果。

特定空间点/区域理论上可选择任意注量率不为零的点/区域，实践中推荐选取求解几何域的中心处，或者具有相同注量率对称分布的不同几何点/区域，预设值可设定为任意固定的归一化注量率值。

此外，采用已经收敛的神经网络作为类似问题的神经网络初始值，将很大程度加速收敛过程，这是由于深度神经网络具备一定的迁移学习特性所导致的^[15]。

2.2 k_{eff} 并行搜索方法

稳态方程式 (3) 需要引入 k_{eff} ，并进行搜索达到临界。由于式 (3) 是特征值方程，常用的是源迭代方法^[1, 5]。基于 PINN 模型的深度学习方程求解方法，可以直接作为其内迭代环节进行 k_{eff} 搜索计算。但由于内迭代次数较多，若每一次内迭代均需要完整的机器学习，迭代计算量比较大， k_{eff} 的搜索效率很低。

本文采用式 (2) 进行 k_{eff} 的搜索，将式 (3) 写成式 (2) 的形式：

$$\frac{1}{D\nabla} \frac{\partial \phi(r, t)}{\partial t} = \nabla^2 \phi(r, t) + \frac{k_{\infty}/k - 1}{L^2} \phi(r, t) \quad (9)$$

给出 k 与 $\phi(r, t_0)$ 的任意初始值，考察经过一定时间 t_r ，即 $t > t_r$ 时， $\phi(r, t)$ 对 t 的偏导数是否接近 0，从而进行临界判断。可设定合适的 ε ，若 $\partial \phi(r, t) / \partial t > \varepsilon$ ，判断为超临界状态，需要增大 k 值；若 $\partial \phi(r, t) / \partial t < -\varepsilon$ ，为次临界状态，需要减小 k 值。当 $-\varepsilon < \partial \phi(r, t) / \partial t < \varepsilon$ ，则认为系统已达临界，此时 $k = k_{\text{eff}}$ ， $\phi(r, t > t_r)$ 即为稳态时系统的 $\phi(r)$ 分布 [式 (3)]。

t_s 的取值应大于注量率从任意初始状态到达“稳定”变化状态的时间, 根据扩散方程中反应堆类型 [表现为式 (9) 中 $D\nu$] 与初始边界条件形式会有所不同; k 值的搜索方法可采用二分法或者将 k 等分成一定数量后, 大规模并行搜索, 可有效提高效率。 k_{eff} 的并行搜索流程如图 1 所示。

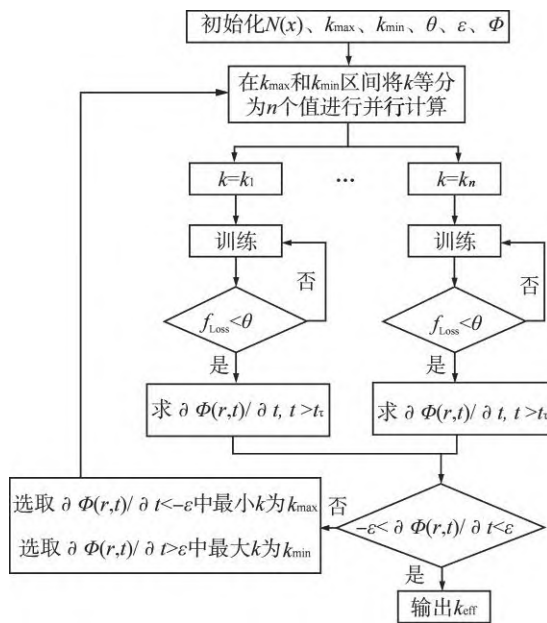


图 1 k_{eff} 并行搜索流程图

Fig. 1 Flow Chart of k_{eff} Parallel Search

k_{max} 、 k_{min} —选取范围的最大值和最小值

2.3 几何空间网格点密度不均匀分布策略

式 (1) ~ 式 (4) 中空间网格几何点 i 的分布与数量, 对模型精度与机器学习效率的影响很大。通常来说, 网格点的数量越多, 计算精度越高, 但计算量越大, 收敛效率降低。同时, 初始条件、边界条件及 $\phi(x)$ 变化很大的地方网格点密度对模型的精度与收敛效率影响很大。

为了平衡机器学习训练精度与效率, 推荐的策略是在相同网格点数量情况下, $\phi(x)$ 变化大的区域网格点相对紧密, 变化平滑的区域相对稀疏, 以 $\phi(x)$ 在几何空间区域网格点偏导数的变化值作为网格点密度设定的判别依据。

据此, 在方程的边界条件和初始条件附近, 或者系统 $\phi(x)$ “畸性”比较大处 (即区域偏导较大处), 应加密网格点提高精度, 而在变化平缓处 (即几何区域点偏导较小处), 应减小网格密度以提高收敛效率。

2.4 深度机器学习的超参数设置

深度机器学习人工神经网络系统包含一系列超参数^[14], 主要有神经网络架构深度/层数、中间隐藏神经元的个数、机器学习率、批量学习样本数等共性参数, 以及与特定的机器学习算法相关的超参数。这些超参数对模型的收敛精度与效率有很大的影响, 而且对于不同的应用系统, 超参数的敏感度不同。目前, 并没有完善的机器学习理论来指导超参数的设计, 不同的应用通常要依靠先验经验、实验进行超参数的选择与设置。

3 数值计算验证

为了验证计算结果, 选取针对特定几何有解析解的扩散方程进行数值验证, 相关结论也可供其他形式的方程与几何形式参考。验证计算神经网络架构均采用全连接方式, 激活函数选取具有高阶导数连续特点的双曲正切函数 \tanh , 其形式为 $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$, 网络初始值权重 $\{\vec{w}, \vec{b}\}$ 采用高斯分布随机采样^[12]。

3.1 临界条件下稳态扩散方程的验证

平板、球、圆柱、方形几种典型几何的稳态方程见文献 [1]。平板的解析解为: $C \cdot \cos(x \cdot \pi / a)$; 球的解析解为: $C / r \cdot \sin(\pi \cdot r / R)$; 圆柱的解析解为: $C \cdot J_0(r \cdot 2.405 / R) \cos(z \cdot \pi / H)$; 方形的解析解为: $C \cdot \cos(x \cdot \pi / a) \cos(y \cdot \pi / b) \cos(z \cdot \pi / c)$ 。平板、球、圆柱、方形几种典型几何参数中 C 为常数; J_0 是第一类零阶修正贝塞尔函数; R 为圆柱半径; H 为圆柱高; a 为长; b 为宽; c 为高。

验证计算神经网络的超参数设定为: 深度 $l=16$, 中间层隐藏神经单元数量 $s=20$, 边界权重 $P_b=100$, $C=0.5$, 几何网格点随机均布, 学习率从 0.001 开始, 训练至损失函数值 f_{Loss} 在 100 次学习内不再下降结束, 结果如表 1 所示。其中

表 1 临界条件下稳态扩散方程计算结果

Tab. 1 Calculation Results of Steady-state Diffusion Equation under Critical Conditions

类型	网格点 (i)	学习次 数(n)	几何参 数/m	$\sigma_{\text{MSE},l}$	$\sigma_{\text{SE},\text{max}}$	$\sigma_{\text{MSE},\text{mc}}$
平板	900	3500	$a=1$	1.4850×10^{-9}	4.9266×10^{-9}	4.8897×10^{-6}
球	2000	3500	$R=1$	3.4875×10^{-10}	1.9138×10^{-9}	6.5248×10^{-6}
圆柱	9000	5000	$R=1, H=2$	4.5593×10^{-6}	4.1536×10^{-5}	1.3897×10^{-5}
方形	9000	5000	$a=1, b=1, c=1$	6.2277×10^{-5}	7.0309×10^{-4}	5.2182×10^{-5}

$\sigma_{\text{MSE},1}$ 为学习好的神经网络函数 $N(\vec{x})$ 在所有网格点上泛化计算结果与解析解的均方差； $\sigma_{\text{SE},\max}$ 是网格点集合中每个点处的 $N(\vec{x})$ 与解析解的方差最大值； $\sigma_{\text{MSE},\text{mc}}$ 是蒙特卡洛方法软件 RMC 计算结果与解析解均方差。

由上可知，总体上机器学习模型具有相当高的精度，但随着扩散方程的输入维度、复杂程度增加，系统的精度有所降低，模型与解析解的均方差分布具有一定程度的随机性，存在个别误差特别高的畸点，对模型的可靠性有一定影响，这是由于深度神经网络目前本身存在一定的不可解释性特性决定的^[14]。同时，在计算效率性能上，由表 1 中机器学习的次数可知，利用 PINN 方法求解这些扩散方程的整体训练时间通常会大于有限差分、蒙特卡洛等传统数值计算方法。但训练好后，表 1 中神经网络进行预测的时间，在普通的微型计算机上实测均小于 0.01 s，几乎可以忽略。

3.2 扩散方程特征向量加速收敛方法验证

验证计算的目标是：统计并分析不同初始权重值 $\{\vec{w}, \vec{b}\}$ 的神经网络 $N(\vec{x})$ ，在训练到相似精度时，所需要的收敛时间。

算例 1、算例 2 网络初始值权重 $\{\vec{w}, \vec{b}\}$ 随机选择^[12]；算例 3 也选择随机初始值，但将 $x=0$ 时， $\phi(0)$ 值 [式 (10) 解析解中的 C 值] 设定为 0.5，如 2.1 节所述作为 f_{Loss} 的组成部分；算例 4、算例 5、算例 6 初始状态为具有不同 C_0 值、已经事先训练好的精度小于 10^{-7} 网络，训练方式与算例 3 类似，将 $\phi(0)=0.5$ 作为加权损失函数组成部分进行训练。各算例训练精度小于 10^{-7} 即停止，记录所需要的训练次数及精度等相关参数，结果如表 2 所示。

可见，算例 1、算例 2 分别收敛到不同的

表 2 神经网络初始值敏感性分析结果

Tab. 2 Sensitivity Analysis Results of Initial Value of Neural Network

算例	网络初始值/ C_0 值	目标 C 值设定	$\sigma_{\text{MSE},1}/10^{-8}$	学习次数 (n)	学习完成时 C 值
1	随机/随机	无	9.9998	2405	0.015
2	随机/随机	无	9.1095	1606	0.0022
3	随机/随机	0.5	9.6124	379	0.4995
4	已训练/ $C_0=0.01$	0.5	9.1010	211	0.4996
5	已训练/ $C_0=0.05$	0.5	9.3309	55	0.4996
6	已训练/ $C_0=0.1$	0.5	9.5475	39	0.4995

C 值，训练次数较多，这说明了若不给定目标 C 值作为训练的加权 f_{Loss} ，由于特征值方程的特征向量不确定性，收敛得到的特征向量是不同的（表现为收敛时 C 值不同），且不易收敛。算例 3 训练次数大幅度下降，而算例 4、算例 5、算例 6 由于有预训练初始值，最终训练次数进一步降低，且初始 C 值越接近目标值 $C=0.5$ ，收敛越快，这表明了采用已训练好的神经网络作为相似问题神经网络初始值，具有良好的迁移学习特性，可大幅度加快收敛进程。

3.3 瞬态条件扩散方程求解及 k_{eff} 搜索方法验证

瞬态方程式 (2) 在平板几何真空边界条件下解析解为^[11]：

$$\phi(x,t) = \sum_{n=1}^{\infty} \frac{2}{a} \left[\int_{-a/2}^{a/2} \phi_0(x') \cos \frac{(2n-1)\pi}{a} x' dx' \right] \times \cos \frac{(2n-1)\pi}{a} x e^{-(k_n-1)t/l_0} \quad (10)$$

验证模型方程参数： $v=2.2 \times 10^3$ m/s， $D=0.211 \times 10^{-2}$ m， $L^2=2.1037 \times 10^{-4}$ m²， $a=1$ m，几何网格点数共 6000，其中 3000 点均匀分布，其余 3000 网格点在边界条件区域、初始条件区域进行局部加密，其他超参数选择与 3.1 节一致。不同的 k_0 值与不同初始条件下，训练 1500 次后，验证计算结果如表 3 所示。4 个算例在定义区域内 $\{D|-0.5 \leq x \leq 0.5, 0 \leq t \leq 0.015\}$ 的神经网络模型泛化计算结果 $N(\vec{x})$ 的散点图，如图 2 所示。

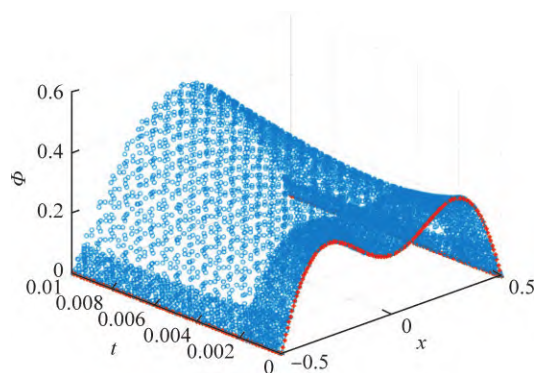
表 3 临界条件下稳态扩散方程数值解 $N(\vec{x})$ 误差分布

Tab. 3 Error Distribution of Numerical Solution of Steady-State Diffusion Equation under Critical Conditions $N(\vec{x})$

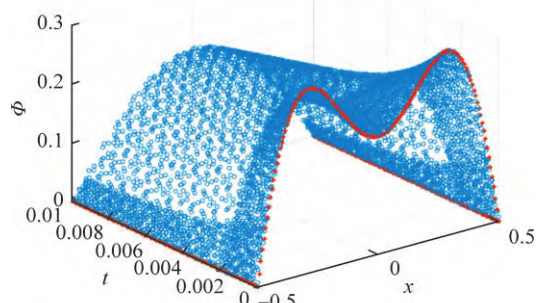
算例	k_0	初始条件	$\sigma_{\text{MSE},1}$
1	1.0041	$\cos(\pi \cdot x/a) - 0.4 \cos(2\pi \cdot x/a) - 0.4$	3.8005×10^{-7}
2	1.0001	$\cos(\pi \cdot x/a) - 0.4 \cos(2\pi \cdot x/a) - 0.4$	6.3758×10^{-7}
3	1.0041	$0.5[\cos(2\pi \cdot x/a) + 1]$	1.5564×10^{-6}
4	1.0001	$0.5[\cos(2\pi \cdot x/a) + 1]$	1.2230×10^{-6}

由图 2 可见，随着时间的变化，注量率值发生明显的变化，表明算例 1 和算例 3 处于超临界，算例 2 和算例 4 处于次临界状态。

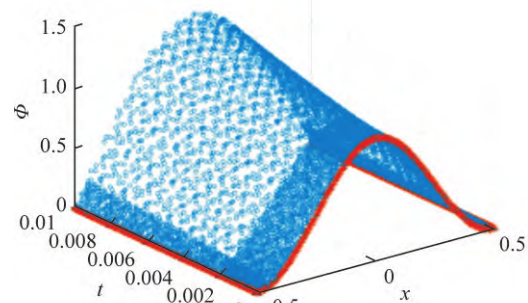
按照 2.2 节所述，式 (9) 采用二分法对上述算例 1、算例 4 进行 k_{eff} 搜索， ε 值从 0.01 开始，逐渐减小。计算结果如表 4 所示。



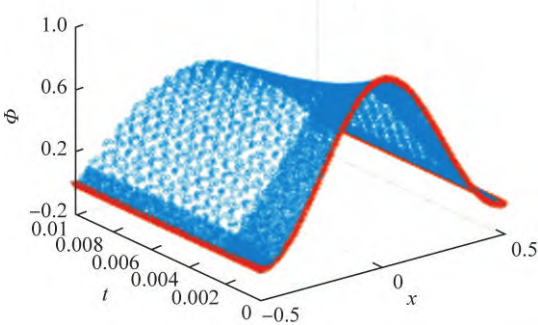
a 算例1散点图



b 算例2散点图



c 算例3散点图



d 算例4散点图

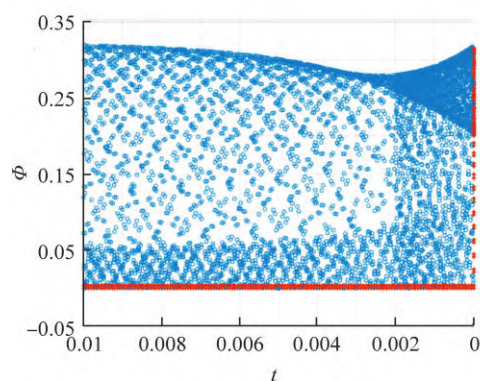
图2 式(2)验证结果图(轴侧视角)
Fig. 2 Verification Result of Equation (2) (Axial View)

表4 k_{eff} 搜索计算结果误差Tab. 4 Error of k_{eff} Search Calculation Results

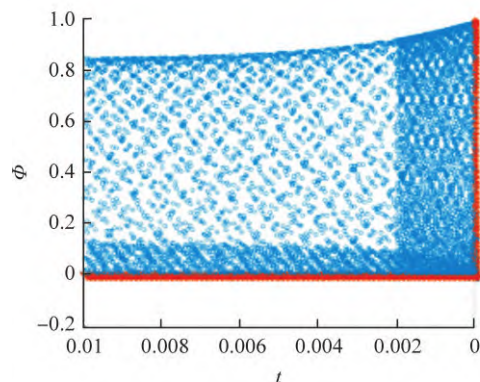
算例	算例5	算例6
k_{∞}	1.00410	1.00010
k_{eff}	1.00202	0.99803
初始条件	$\cos(\pi \cdot x/a) - 0.4\cos(2\pi \cdot x/a) - 0.4$	$0.5\cos(2\pi \cdot x/a) + 0.5$
$\sigma_{\text{MSE},a}$	8.9108×10^{-7}	4.4229×10^{-6}
$\sigma_{\text{MSE},b}$	3.8236×10^{-6}	1.8132×10^{-5}
$\sigma_{\text{MSE},c}$	3.6584×10^{-7}	8.9051×10^{-6}

$\sigma_{\text{MSE},a} = N(x,t)$ 在全域 $\{D|-0.5 \leq x \leq 0.5, 0 \leq t \leq 0.015\}$ 上计算结果与解析解式(10)的均方差; $\sigma_{\text{MSE},b} = N(x,0.015)$ 在 $\{x|-0.5 \leq x \leq 0.5\}$ 上与解析解式(10)的均方差; $\sigma_{\text{MSE},c} = N(x,0.015)$ 在 $\{x|-0.5 \leq x \leq 0.5\}$ 上与平板解析解的均方差

算例5和算例6神经网络函数 $N(x,t)$ 全域计算结果如图3所示。



a 算例5散点图



b 算例6散点图

图3 k_{eff} 搜索散点图(沿X方向正视图)Fig. 3 k_{eff} Search Scatter Diagram (Front View Along X Direction)

上述过程, 逐渐调整 k 值到系统临界状态。若系统已接近临界状态, 由于没有考虑缓发中子, 不同边界条件下, 系统都会在非常短的时间

(5~8 ms) 收敛到临界状态, 系统此时 $N(x)$ 值逼近平板稳态解析解, 误差如表 4 中 $\sigma_{\text{MSE},3}$ 所示。若系统不临界, 如图 2 所示, 也会在很短时间内 $\partial N(x,t)/\partial t$ 值有明显变化。由此特性, 可实现式 (3) 的 k_{eff} 的搜索。

3.4 网格点不均匀分布策略验证

对 2.3 节所述网格点分布影响进行验证, 选择表 4 中算例 5 作为对象, 学习参数保持与原算例一致, 训练 1000 次, 网格点总数 2000, 不同验证算例仅调整网格点分布。设为 5 个区域:

$$D_0: \{(x,t)|-0.5 \leq x \leq 0.5, 0 \leq t \leq 0.01\}$$

$$D_{C1}: \{(x,t)|-0.5 \leq x \leq 0.5, 0 \leq t \leq 0.002\}$$

$$D_{C2}: \{(x,t)|-0.5 \leq x \leq 0.5, 0 \leq t \leq 0.005\}$$

$$D_{B1}: \{(x,t)|-0.5 \leq x \leq -0.45, 0 \leq t \leq 0.01\}$$

$$D_{B2}: \{(x,t)|0.45 \leq x \leq 0.5, 0 \leq t \leq 0.01\}$$

各区域内的网格点随机分布。4 种算例计算结果 $N(x)$ 与解析解式 (10) 均方差如表 5 所示。

表 5 网格点分布敏感性分析结果

Tab. 5 Sensitivity Analysis Results of Grid Point Distribution

算例	网格点分布	$\sigma_{\text{MSE},1}$
1	$D=D_0$, $\text{card}(D_0)=2000$	5.1604×10^{-6}
2	$D=D_0 \cup D_{C1}$, $\text{card}(D_0)=1500$, $\text{card}(D_{C1})=500$	2.8265×10^{-6}
3	$D=D_0 \cup D_{C2}$, $\text{card}(D_0)=1500$, $\text{card}(D_{C2})=500$	1.2731×10^{-6}
4	$D=D_0 \cup D_{C2} \cup D_{B1} \cup D_{B2}$ $\text{card}(D_0)=1000$, $\text{card}(D_{C2})=500$, $\text{card}(D_{B1})=250$, $\text{card}(D_{B2})=250$	8.4288×10^{-7}

card—范围内的网格点数目

3.5 深度神经网络超参数敏感性验证

采用 3.1 节中平板几何算例进行神经网络深度、隐藏神经元数量、初始值关键超参数敏感性验证计算。验证过程中, 除特别说明, 其他参数取值与 3.1 节相同。

3.5.1 神经网络深度 (网络层数) 敏感性分析验证 网络隐藏神经单元 $s=20$, 迭代 2000 次, 网格点 $i=500$ (均匀分布), 对不同网络深度验证结果如表 6 所示。

可见, 随着网络深度加深, 神经网络模型精度明显提升, 但网络过深, 会出现梯度消失情况, 网络无法收敛。

3.5.2 隐藏神经元数量敏感性分析验证 神经网络深度 $l=5$, 其余参数同 3.5.1 节, 对神经网络

中间层不同数量隐藏单元验证计算结果如表 7 所示。

表 6 神经网络深度敏感性分析结果

Tab. 6 Sensitivity Analysis Results of Neural Network Depth

算例	l	$\sigma_{\text{MSE},1}$
1	3	1.9791×10^{-8}
2	5	1.1187×10^{-8}
3	9	2.6775×10^{-9}
4	16	2.1024×10^{-9}
5	32	1.6828×10^{-9}

表 7 隐藏神经元数量敏感性分析结果

Tab. 7 Sensitivity Analysis Results of Number of Hidden Neurons

算例	s	$\sigma_{\text{MSE},1}$
1	5	3.6171×10^{-6}
2	10	2.874×10^{-7}
3	20	1.1187×10^{-8}
4	50	1.0632×10^{-8}
5	100	1.1001×10^{-8}

可见, 随着隐藏神经元数量增加, 神经网络模型精度也明显提升, 但增加到一定数量, 精度不再提高, 而带来训练时间开销会大幅度增加的负面效应。

3.6 边界条件损失函数权重敏感性分析验证

验证算例采用 3.1 节中的平板算例, 神经网络深度 $l=5$, 网络隐藏神经单元 $s=20$, 每次计算均选择相同的随机初始函数, 除了式 (8) 中边界条件损失函数权重 P_b , 算例其他参数、初始网络设置均相同。验证计算不同权重 P_b 条件下, 训练达到 4 种精度所需要的训练次数 n , 结果如表 8 所示。

将表 8 训练次数取对数后, 权重敏感性分析特性如图 4 所示。

由此可见, P_b 值的大小对收敛速度的影响是明显的, 适当增加 P_b 值, 可以加速收敛, 但 P_b 值过大, 训练收敛次数反而变大。在不同精度水平, 引起最快收敛速度的 P_b 值也不同, 总体呈现随着精度水平增加, 逐渐增大趋势, 但精度高到一定程度, 会出现震荡。图 4 表明存在有规律的最佳权重区, P_b 最佳值随着学习精度的提升逐渐加大。

表 8 不同边界条件损失函数权重训练结果

Tab. 8 Weight Training Results of Loss Function under Different Boundary Conditions

P_b	训练次数 n			
	$\sigma_{\text{MSE},1}=5\times 10^{-8}$	$\sigma_{\text{MSE},1}=10^{-7}$	$\sigma_{\text{MSE},1}=10^{-6}$	$\sigma_{\text{MSE},1}=10^{-5}$
1	827	621	185	115
5	2257	367	129	50
10	2431	384	71	39
50	305	249	73	47
100	238	164	79	46
200	752	266	88	53
300	497	224	86	58
400	442	245	87	64
500	245	218	90	70

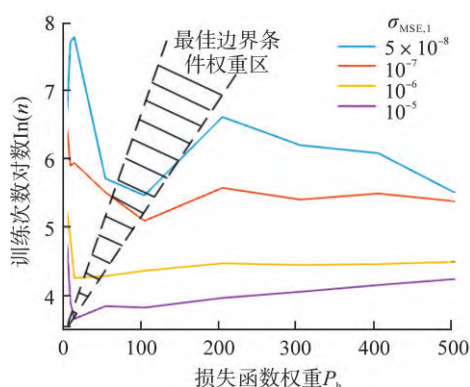


图 4 边界条件损失函数权重敏感性分析

Fig. 4 Sensitivity Analysis of Weight of Boundary Condition Loss Function

4 结 论

本文结合 PINN 深度机器学习技术求解微方程的基本原理,实现了多种形式中子学扩散方程的数值求解,数值计算验证结果表明该方法具有良好的精度。同时,针对扩散方程的特点,提出了特征值方程加速收敛方法、 k_{eff} 高效并行搜索方法、样本网格点不均匀分布策略等关键技术,并针对神经网络结构、边界条件损失函数权重等关键参数进行了敏感性分析,验证计算结果表明这些关键技术具有显著的效果。研究工作为中子学扩散方程的数值求解提供了新的技术途径,并可为进一步求解复杂几何多群多维扩散问题、中子学输运方程问题等更加复杂的反应堆物理中子学偏微分方程奠定良好的技术基础。

参考文献:

[1] 谢仲生, 曹良志, 张少泓. 核反应堆物理分析[M]. 第

五版. 西安: 西安交通大学出版社, 2020: 69-90.

[2] 曹良志, 谢仲生, 李云召. 近代核反应堆物理分析[M]. 北京: 中国原子能出版社, 2017: 8-209.

[3] 周蜀林. 偏微分方程[M]. 北京: 北京大学出版社, 2005: 1-16.

[4] 张强. 偏微分方程的有限差分方法[M]. 北京: 科学出版社, 2017: 77-114.

[5] 李治平. 偏微分方程数值解讲义[M]. 北京: 北京大学出版社, 2010: 202-282.

[6] GALIB S M, BHOWMIK P K, AVACHAT A V, et al. A comparative study of machine learning methods for automated identification of radioisotopes using NaI gamma-ray spectra[J]. Nuclear Engineering and Technology, 2021, 53(12): 4072-4079.

[7] VAN MILLIGEN B P, TRIBALDOS V, JIMÉNEZ J A. Neural network differential equation and plasma equilibrium solver[J]. Physical Review Letters, 1995, 75(20): 3594-3597.

[8] MAI-DUY N, TRAN-CONG T. Numerical solution of differential equations using multiquadric radial basis function networks[J]. Neural Networks, 2001, 14(2): 185-199.

[9] MAI-DUY N, TANNER R I. Solving high-order partial differential equations with indirect radial basis function networks[J]. International Journal for Numerical Methods in Engineering, 2005, 63(11): 1636-1654.

[10] MALL S, CHAKRAVERTY S. Single layer chebyshev neural network model for solving elliptic partial differential equations[J]. Neural Processing Letters, 2017, 45(3): 825-840.

[11] XIE Y C, WANG Y H, MA Y, et al. Neural network based deep learning method for multi-dimensional neutron diffusion problems with novel treatment to boundary[J]. Journal of Nuclear Engineering, 2021, 2(4): 533-552.

[12] RAISSI M, PERDIKARIS P, KARNIADAKIS G E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. Journal of Computational Physics, 2019, 378: 686-707.

[13] 刘宪高. 变分法与偏微分方程[M]. 北京: 科学出版社, 2016: 45-122.

[14] HAGAN M T, DEMUTH H B, BEALE M H, et al. Neural network design[M]. ZHANG Y, trans. 2nd ed. Beijing: China Machine Press, 2018: 1-25.

[15] 杨强, 张宇, 戴文渊, 等. 迁移学习[M]. 庄福振, 译. 北京: 机械工业出版社, 2020: 2-79.

(责任编辑: 黄可东)