arXiv:2304.00909v1 [math.NA] 3 Apr 2023

Laplace-fPINNs：用于求解亚扩散正问题与逆问题的基于拉普拉斯变换的分数阶物理信息神经网络

Xiong-Bin Yan[a, d], Zhi-Qin John Xu[a, b, c, *], Zheng Ma[a, b, c, d, *]

[a]上海交通大学数学科学学院，中国上海[b]上海交通大学自然科学研究院，*MOE-LSC*，中国上海[c]上海交通大学清源研究院，中国上海[d]*CMA-Shanghai*，上海交通大学，中国上海

摘要

物理信息神经网络（PINNs）在求解分数阶扩散方程的正问题与反问题方面显示出良好前景。然而，由于自动微分不适用于分数阶导数，使用物理信息神经网络求解分数阶扩散方程需要应对额外挑战。为解决这一问题，本文提出了物理信息神经网络的一种扩展，称为基于拉普拉斯变换的分数阶物理信息神经网络（Laplace-fPINNs），可有效求解分数阶扩散方程的正问题与反问题。该方法避免引入大量辅助点，并简化了损失函数。我们通过多个示例验证了 Laplace-fPINNs 方法的有效性。数值结果表明，Laplace-fPINNs 方法能够有效求解高维分数阶扩散方程的正问题与反问题。

关键词： 物理信息神经网络、拉普拉斯变换、拉普拉斯变换数值逆变换、亚扩散

1. 引言

分数阶扩散方程因其在建模反常扩散现象方面的优越能力，已在工程学、物理学和数学文献中得到广泛研究。该模型与标准扩散模型的区别在于，它基于扩散遵循标准布朗运动这一基本假设，并已应用于动物皮毛图案和神经细胞信号。标准布朗运动的一个显著特征是，扩散物种的均方位移 $\langle x^2(t)\rangle$ 随时间线性增长，即 $\langle x^2(t)\rangle \sim K_1 t$。然而，在反常扩散中，均方位移随时间呈非线性幂律增长，即 $\langle x^2(t)\rangle \sim K_\alpha t^\alpha$，其中 $0 < \alpha < 1$ 表示亚扩散，而 $\alpha > 1$ 表示

*通讯作者电子邮箱： yanxb2015@163.com (Xiong-Bin Yan), xuzhiqin@sjtu.edu.cn(Zhi-Qin John Xu), zhengma@sjtu.edu.cn (Zheng Ma)

超扩散。在微观层面，此类反常扩散过程可由连续时间随机游走准确描述，其中连续两次粒子跃迁之间的等待时间服从均值发散的重尾分布。在宏观层面，反常扩散描述了粒子在给定空间位置 x 和时间 t 出现的概率密度函数的演化。分数阶扩散方程的成功应用已十分广泛，且仍在不断扩展。这些应用包括但不限于：非均质介质中的溶质输运 [1, 2]，分形区域上的热扩散 [3]，膜内的蛋白质输运 [4, 5]，以及高度非均质含水层中的流动 [6]。关于物理学建模及各种应用的全面综述可见 [7, 8]。

本文考虑有界区域 $\Omega \subset \mathbb{R}^d$ 上满足齐次狄利克雷边界条件的以下时间分数阶扩散方程

$$\begin{cases} \partial_{0+}^{\alpha} u(x,t) = \nabla \cdot (a(x)\nabla u(x,t)) + c(x)u(x,t) + f(x,t),\ x \in \Omega,\ t > 0, \\ u(x,0) = u_0(x),\ x \in \Omega, \\ u(x,t) = 0,\ x \in \partial\Omega,\ t > 0. \end{cases} \tag{1.1}$$

上述方程左端是阶数为 α的Caputo导数，其定义为

$$\partial_{0+}^{\alpha} u(x,t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} \frac{\partial u(x,\tau)}{\partial \tau} d\tau,\ 0 < \alpha < 1,$$

其中 $\Gamma(\cdot)$ 是伽马函数，$a(x)$、$c(x)$、$f(x,t)$ 和 $u_0(x)$ 分别表示扩散系数、反应系数、源项和初始值。

数值求解偏微分方程是一项众所周知的挑战，而对于涉及非局部算子的分数阶扩散方程，求解难度更大。近年来，应用机器学习技术求解偏微分方程正问题与逆问题的趋势日益增强。相关例子包括使用高斯过程回归 [9–11] 以及基于深度学习的方法 [12–17] 来求解此类问题。

本文重点探讨如何运用深度学习方法求解偏微分方程（PDEs），这类方法可分为两种途径。第一种途径利用大量通过数值模拟或控制方程获得的输入—输出数据对，让神经网络学习给定问题的解算子。在这种情况下，深度神经网络用于近似一个将输入（例如系数、初始值、源项和边界）映射为偏微分方程解的函数。更多相关研究见 [18–25]。第二种途径则利用深度神经网络强大的表达能力来近似偏微分方程的解。该方法借助自动微分领域的最新进展；自动微分是科学计算中的一项实用技术，可用于求出神经网络的导数并构造代价函数，再通过最小化代价函数来优化神经网络的可训练参数，

这一途径简洁而强大，并催生了用于求解偏微分方程的物理信息深度学习方法。此类方法包括 Deep Ritz 方法 [15, 26], 物理信息神经网络方法 [13, 14, 27–31],弱对抗网络方法 [16, 17], Deep Galerkin 方法 [32, 33], 以及多尺度深度神经网络 [34, 35]。

近年来，利用神经网络求解分数阶偏微分方程越来越受到关注 [25, 36, 37]。然而，经典链式法则不适用于分数阶微积分，这使得自动微分无法有效处理分数阶导数。为克服这一挑战，Pang 等人 [36] 提出了物理信息神经网络（PINNs）的一种扩展，称为分数阶物理信息神经网络。该方法通过对分数阶算子进行数值离散化来近似分数阶导数，并利用自动微分解析地获取神经网络的整数阶导数。类似地，Guo 等人 [37] 提出了一种基于蒙特卡罗采样的物理信息神经网络方法，通过蒙特卡罗采样计算神经网络输出的分数阶导数。然而，这两种方法都需要引入辅助点来计算分数阶导数，从而增加了神经网络训练的计算成本。

本文提出了一种基于拉普拉斯变换的分数阶物理信息神经网络，称为 Laplace-fPINNs，用于求解亚扩散的正问题与逆问题。所提出的方法利用拉普拉斯变换，将原始的时间分数阶扩散方程转换为拉普拉斯空间中的受限方程，随后可利用物理信息神经网络对其进行求解。拉普拉斯空间中的受限方程不包含分数阶导数，从而避免了对时间分数阶导数应用自动微分的问题。通过拉普拉斯变换数值逆变换，将物理信息神经网络在拉普拉斯空间中的解转换到时域。总之，本文的主要贡献如下：

- 我们提出了一种名为 Laplace-fPINNs 的新方法，用于求解时间分数阶扩散方程（1.1）。该方法显著降低了神经网络训练过程中损失函数的计算成本，并消除了以往研究中为近似时间分数阶导数而使用辅助点的需求 [36, 37]。

- 我们采用所提出的 Laplace-fPINNs 方法，解决根据给定测量数据识别高维扩散系数这一具有挑战性的任务。传统反演方法通常难以求解这一问题。

本文的结构安排如下。第2节回顾物理信息神经网络（PINNs）的基础知识，并总结此前利用物理信息神经网络求解分数阶偏微分方程的一些研究。第3节详细介绍基于拉普拉斯变换的物理信息神经网络，用于求解时间分数阶扩散方程（1.1）。第4节全面考察 Laplace-fPINNs 方法在求解

方程（1.1）相关正问题与逆问题时的性能。最后，我们在第5节总结全文。

2. 预备知识

2.1. 物理信息神经网络

本节首先回顾物理信息神经网络（PINNs）求解一般形式的整数阶非线性偏微分方程的基本思想

$$u_t + \mathcal{N}[u;\lambda] = 0,\ x \in \Omega,\ t \in [0,T], \tag{2.1}$$

其中，$u(x,t)$ 表示方程 (2.1) 的解，$\mathcal{N}[\cdot;\lambda]$ 是由 λ 参数化的非线性算子，Ω 是 $\mathbb{R}^d$ 中的有界区域。物理信息神经网络方法通过深度神经网络逼近 $u(x,t)$ 来求解上述方程 (2.1)。该网络以坐标 (x,t) 为输入，并输出标量 $u_{NN}(x,t;\theta)$，其中 θ 表示神经网络的所有可训练参数。近似解 $u_{NN}(x,t;\theta)$ 是可微的，可通过自动微分 [38]代入方程 (2.1)，从而得到

$$r_{NN}(x,t;\theta) := \partial_t u_{NN}(x,t;\theta) + \mathcal{N}[u_{NN}(x,t;\theta);\lambda]. \tag{2.2}$$

神经网络 $u_{NN}(x,t;\theta)$ 和 $r_{NN}(x,t;\theta)$ 共用相同的可训练参数，这些参数可通过最小化以下损失函数进行学习

$$\begin{aligned} L(\theta) &:= L_{bd}(\theta) + L_{eq}(\theta) \\ &= \frac{1}{N_u}\sum_{i=1}^{N_u} |u_{NN}(x_u^i, t_u^i;\theta) - u^i|^2 + \frac{1}{N_r}\sum_{i=1}^{N_r} |r_{NN}(x_r^i, t_r^i;\theta)|^2, \end{aligned}$$

其中，$\{x_u^i、\ t_u^i、\ u^i\}_{i=1}^{N_u}$ 表示$u(x,t)$的初始与边界训练数据，$\{x_r^i、\ t_r^i\}_{i=1}^{N_r}$ 指定了$r_{NN}(x,t;\theta)$的配点。图1展示了物理信息神经网络的示意图。

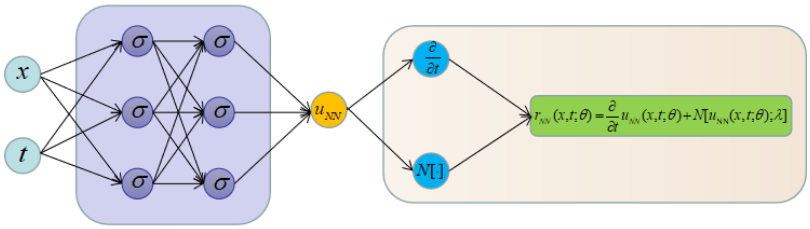


图 1：用于求解偏微分方程的物理信息神经网络。

2.2. 使用物理信息神经网络求解分数阶偏微分方程

物理信息神经网络是一种利用神经网络求解偏微分方程（PDEs）的新方法。然而，在处理涉及非整数阶导数的分数阶偏微分方程时，物理信息神经网络面临一些挑战。主要挑战之一是计算分数阶 Caputo 导数，该导数广泛应用于物理学和工程学。与整数阶导数不同，Caputo导数取决于函数的整个历史，这意味着随着模拟时间的推进，数值积分的计算成本和内存占用会不断增加。此外，Caputo导数不遵循经典链式法则，这意味着无法直接应用物理信息神经网络中的标准自动微分技术。因此，需要开发新的方法和算法来克服这些困难，使物理信息神经网络在求解分数阶偏微分方程时更加高效、稳健。

受此前使用物理信息神经网络求解整数阶偏微分方程的方法启发，Pang 等人在 [36] 中提出了一种用于求解分数阶偏微分方程的分数阶物理信息神经网络方法。具体而言，分数阶物理信息神经网络采用自动微分解析地求出神经网络输出的整数阶导数，并通过数值离散近似分数阶导数。此外，为降低分数阶物理信息神经网络进行数值离散所产生的高昂成本，Guo 等人在 [37] 中提出了一种基于蒙特卡洛采样的物理信息神经网络（MC-fPINNs）方法，用于求解分数阶偏微分方程。该方法将时间分数阶导数简化为一个函数的期望，并通过引入概率密度函数，使用蒙特卡洛方法求解该期望。尽管分数阶物理信息神经网络和 MC-fPINNs 在求解分数阶偏微分方程的正问题与逆问题时均表现出较高精度，但它们都需要辅助点来求出神经网络输出的时间分数阶导数，这增加了使用神经网络近似分数阶导数的计算成本。

3. Laplace-fPINNs

为解决上述问题，我们提出了一种名为基于拉普拉斯变换的分数阶物理信息神经网络（*L-fPINNs*）的新方法，该方法利用拉普拉斯变换解析地计算神经网络输出的时间分数阶导数。该方法无需辅助点和数值离散，从而降低了利用神经网络求解分数阶偏微分方程的计算成本和复杂度。所提出的方法包含三个要素：首先，对分数阶偏微分方程进行拉普拉斯变换；其次，在拉普拉斯域中应用物理信息神经网络方法；最后，通过数值反演获得最终解。我们表明，应用拉普拉斯变换可以将分数阶偏微分方程转化为拉普拉斯域中的常微分方程，并可用物理信息神经网络高效求解。随后，我们采用数值反演技术，在原始域中恢复解。

3.1. 拉普拉斯变换

首先，我们给出一维拉普拉斯变换的定义，该变换将函数从时域转换到拉普拉斯空间域。设$f(t)$为一个依赖于时间变量$t \geq 0$的函数，$f(s)$表示其在拉普拉斯空间中的像，其定义为

$$\tilde{f}(s) = \mathcal{L}[f(t)](s) = \int_0^\infty e^{-st} f(t)dt, \quad (s \in \mathbb{C}). \tag{3.1}$$

如果积分（3.1）在某一点 $s_0 \in \mathbb{C}$处收敛，那么对于满足 $\Re(s) > \Re(s_0)$ 的任意 $s \in \mathbb{C}$，该积分都绝对收敛，其中 $\Re(s)$ 表示复变量 s 的实部。时间分数阶 Caputo 导数 $\partial_{0+}^{\alpha} f(t)$ 的拉普拉斯变换可写为

$$\begin{aligned}\mathcal{L}[\partial_{0+}^{\alpha} f](s) &= s^{\alpha}\mathcal{L}[f(t)](s) - s^{\alpha-1}f(0) \\ &= s^{\alpha}\tilde{f}(s) - s^{\alpha-1}f(0), \quad t \geq 0,\ 0 < \alpha < 1.\end{aligned}$$

基于上述介绍，我们对方程（1.1）两边应用拉普拉斯变换，得到

$$\begin{cases} s^{\alpha}\tilde{u}(x,s) - \nabla\cdot(a(x)\nabla\tilde{u}(x,s)) - c(x)\tilde{u}(x,s) = s^{\alpha-1}u_0(x) + \tilde{f}(x,s),\ x \in \Omega,\ s \in \mathbb{C}, \\ \tilde{u}(x,s) = 0,\ x \in \partial\Omega,\ s \in \mathbb{C}, \end{cases} \tag{3.2}$$

其中 $\tilde{f}(x,s) = \mathcal{L}[f(x,t)](x,s) = \int_0^\infty e^{-st} f(x,t)dt$。值得注意的是，拉普拉斯空间中的时间分数阶扩散方程不再包含时间分数阶导数 $\partial_{0+}^{\alpha} u(x,t)$，这与原始的分数阶扩散方程（1.1）不同。这意味着，如果我们直接通过方程（3.2）求解数值解 u ，就可以大幅降低与时间分数阶导数离散化相关的计算开销。

3.2. 拉普拉斯域中的物理信息神经网络

接下来，我们提出一种基于拉普拉斯变换的物理信息神经网络方法，用于近似求解方程（3.2）。我们将方程（3.2）重新表述如下：

$$\begin{cases} \mathcal{A}[\tilde{u}](x,s) = F(x,s),\ x \in \Omega,\ s \in \mathbb{C}, \\ \tilde{u}(x,s) = 0,\ x \in \partial\Omega,\ s \in \mathbb{C}, \end{cases} \tag{3.3}$$

其中 $\mathcal{A}[\tilde{u}] = s^{\alpha}\tilde{u}(x,s) - \nabla\cdot(a(x)\nabla\tilde{u}(x,s)) - c(x)\tilde{u}(x,s)$，$F(x,s) = s^{\alpha-1}u_0(x) + \tilde{f}(x,s)$。我们采用深度神经网络 $\tilde{u}_{NN}(x,s;\theta)$ 来近似方程 (3.3) 中的解 $\tilde{u}$，其中 θ 表示神经网络的所有可训练参数。随后，将 $\tilde{u}_{NN}(x,s;\theta)$ 代入方程 (3.3)，得到相应的残差

$$r_{NN}(x,s;\theta) := \mathcal{A}[\tilde{u}_{NN}](x,s) - F(x,s).$$

基于拉普拉斯变换的物理信息神经网络可通过最小化以下复合损失函数进行训练

$$L^{lp}(\theta) := w_{bd}L^{lp}_{bd}(\theta) + w_{eq}L^{lp}_{eq}(\theta), \tag{3.4}$$

其中

$$L^{lp}_{eq}(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r} |r_{NN}(x^i_r, s^i_r; \theta)|^2,$$

$$L^{lp}_{bd}(\theta) = \frac{1}{N_{bd}}\sum_{i=1}^{N_{bd}} |\tilde{u}_{NN}(x^i_{bd}, s^i_{bd}; \theta)|^2.$$

批次大小N_r 和N_{bd} 分别表示用于计算残差$r_{NN}(x, s;\ \theta)$（x ∈属于 Ω）和边界$\mathrm{u}_{NN}(x, s; \theta)$（$x$ ∈ ∂属于 Ω）的配点数量。在随机下降算法的每次迭代中，这些配点都会在计算区域和边界内随机采样。此外，损失函数$L^{lp}(\theta)$中的权重系数w^{eq}和w_{bd} 可以平衡各损失项不同的学习率。权重系数w^{eq}、w_{bd} 以及批次大小N_r、N_{bd} 的具体取值将在实验中给出。

值得注意的是，式（3.4）中的损失函数不包含时间分数阶 Caputo 导数 ∂^{α}_{0+}，仅包含项$s^{\alpha}\tilde{u}_{NN}(x, s;\ \theta)\ - s^{\alpha-1}u_0(x)$。这意味着，在训练过程中，无需对时间分数阶导数 ∂^{α}_{0+} 进行离散化以最小化式（3.4）中的损失函数。因此，也不需要如先前研究 [36, 37]所建议的那样使用辅助点。

3.3. 拉普拉斯变换的数值逆变换

通过使用随机梯度下降算法最小化损失函数 (3.4)，我们得到了方程 (3.3) 的近似解，记作 $\tilde{u}_{NN}(x, s; \theta^*)$。接下来，我们讨论如何将拉普拉斯空间域中的近似解 $\tilde{u}_{NN}(x, s; \theta^*)$ 转换为原始问题 (1.1) 的解 u，这需要使用拉普拉斯变换数值逆变换 (NILT)。拉普拉斯变换的数值求逆是一个众所周知的问题，且其病态性十分突出。数值求逆过程可能不稳定，其难点在于对舍入误差高度敏感。为应对这一挑战，研究人员提出了多种拉普拉斯变换数值求逆算法，例如傅里叶级数法 [39], 、Talbot算法 [40],和 Gaver-Stehfest算法 [41]。本研究选择使用 Gaver-Stehfest数值逆拉普拉斯变换算法，这是一种成熟且易于实现的方法。此外，该逆变换算法的求和权重和节点不依赖复数。利用这一算法，我们能够将拉普拉斯空间域中的近似解 $\tilde{u}_{NN}(x, s; \theta^*)$ 转换为方程 (1.1) 中的含时解 u。一个近似解

Stehfest数值逆拉普拉斯变换算法在方程 (1.1) 的特定时刻 t 给出的解为 $u_{NN}(x,t;\theta^*)$

$$u_{NN}(x,t;\theta^*) = \frac{\ln 2}{t}\sum_{i=1}^{M}\mu_i \tilde{u}_{NN}(x, \frac{\ln 2}{t}i;\theta^*), \tag{3.5}$$

其中系数 μ_i 由下式给出

$$\mu_i = (-1)^{\frac{M}{2}+i}\sum_{k=[\frac{i+1}{2}]}^{\min(i,\frac{M}{2})}\frac{k^{\frac{M}{2}}(2k)!}{(\frac{M}{2}-k)!(k)!(k-1)!(i-k)!(2k-i)!},$$

其中 $[C]$ 表示小于或等于C的最大整数，式 (3.5) 中的 M 为偶数 .

根据式 (3.5)，对于给定的 t，我们可以通过对序列 $\{\tilde{u}_{NN}(x, s_i;\theta^*)\}_{i=1}^{M}$进行线性组合，得到近似解 $u_{NN}(x, t;\theta^*)$，其中 $s_i = \ln 2\, t\, i$，$i = = 1,\ldots,M$。序列 $\{\tilde{u}_{NN}(x, s_i;\theta^*)\}_{i=1}^{M}$是将配点 $\{(x, s_i)\}_{i=1}^{M}$ 输入神经网络近似解 $\tilde{u}_{NN}(x, s;\theta^*)$后得到的，该近似解是变量 x 和 s 的函数。

值得注意的是，根据拉普拉斯变换的定义 (3.1)，神经网络输出 $\tilde{u}_{NN}(x, s;\theta^*)$需要近似方程 (3.3) 中的解 $\tilde{u}(x, s)$，其中 $x \in \Omega$，$s \in \mathbb{C}$。然而，训练点 $s \in \mathbb{C}$ 的选择并非神经网络训练过程所必需。实际上，要使用式 (3.5) 中的 Stehfest 数值拉普拉斯逆变换方法求得近似解 $u_{NN}(x,t;\theta^*)$，我们只需在 $s_i = \ln 2\, t\, i$ 处计算神经网络输出 $\tilde{u}_{NN}(x, s_i;\theta^*)$，其中 $i = 1,\ldots,M$。因此，在神经网络训练过程中，我们选择训练点 $s \in \mathcal{S} \subset \mathbb{R}^+$；它是$\mathbb{R}^+$ 中的一个有界闭区间，具体将在下一节说明。

4. 实验

本节将通过若干数值算例验证所提出的 Laplace-fPINNs 方法的有效性。首先，我们评估 Laplace-fPINNs 方法求解二维和三维亚扩散正问题 (1.1) 的表现。随后，我们展示 Laplace-fPINNs 方法在三维亚扩散问题中识别扩散系数这一反问题上的效率。为量化 Laplace-fPINNs 方法的精度，我们考虑相对l_2 误差，其定义如下：

$$\text{Relative } l_2 \text{ error} = \frac{\|u(x,t) - u_{NN}(x,t)\|_2}{\|u(x,t)\|_2},$$

其中u 和 u_{NN} 分别表示精确解和近似解。

在所有实验中，我们使用带有 Swish激活函数的全连接神经网络来逼近方程 (3.3) 的解 $\tilde{u}(x,s)$。在所有数值实验中，神经网络可训练参数的初始值均设为默认值。该神经

网络使用 Adam优化器进行训练，学习率为 1×10^{-4}。优化器的超参数根据默认建议进行选择。

4.1. 二维正问题

4.1.1. T = 1 时的二维正问题

为研究Laplace-fPINNs近似的性能，我们考虑求解二维情形下的方程（1.1）正问题。不失一般性，取区域 $\Omega = [0,1]^2$，$T = 1$。扩散系数和反应系数分别设为$a(x,y) = 1$和 $c(x,y) = 0$。初始值和源项定义为

$$u_0(x,y) = \sin(\pi x)\sin(\pi y),$$
$$f(x,y) = 5\sin(2\pi x)\sin(3\pi y).$$

我们使用L_1型有限差分方法 [42]得到参考解u。数值方法中时间变量和空间变量的网格分辨率分别取为101和 101×101。

在本实验中，对于每个小批量，我们将用于计算方程损失函数 $L_{eq}^{lp}(\theta)$ 和边界损失 L^{lp} bd(θ) 的残差点数和边界点数分别设为1000和1600。每次迭代时，空间变量的残差点和边界点分别从 Ω 和 $\partial\Omega$ 上的均匀分布中采样。训练点 s 从区间$\mathcal{S} = [s_{min}、s_{max}]$中采样，其中 $s_{min} = \ln 2$ T，$s_{max} = \ln 2\, t_1$ M，$t_1 = 0.01$。(3.4) 中的权重 w^{eq} 和 w_{bd} 是超参数，可在训练过程中手动或自动调节。w^{eq} 和w_{bd} 的大小会显著影响神经网络的收敛速度。然而，讨论权重 w^{eq} 和w_{bd} 如何影响Laplace-fPINNs的收敛速度，超出了本文的讨论范围。因此，在本例中，我们以启发式方式设置权重 $w_{eq} = 1$ 和$w_{bd} = 2000$。

我们首先讨论式 (3.5) 中用于 Stehfest数值逆拉普拉斯变换算法的参数 M 如何影响 Laplace-fPINNs 方法的精度。为此，我们将神经网络的隐藏层数及每个隐藏层中的神经元数分别设为 5 和 256，并将迭代次数设为 1.2×10^5。为减少神经网络参数随机初始化对数值精度的影响，我们运行 Laplace-fPINNs 代码五次，并报告相对l_2 误差的均值和标准差，结果见表 1。

如表 1 所示，我们的结果表明，要获得最佳数值精度，M 的最优值为 4，增大M 并不会降低相对l_2 误差。这一现象可能是因为近似解 $\tilde{u}_{NN}(x, s; \theta)$ 对方程 (3.3) 的真实解近似得不够准确，而拉普拉斯变换数值逆变换过程可能会引入额外误差。先前的研究也报告了类似的观察结果，例如 [43, 44]。

为验证 Laplace-fPINNs 方法对神经网络深度和宽度的敏感性，我们计算了近似解 $u_{NN}(x,t;\ \theta^*)$ 与采用L_1 型有限差分方法得到的参考解 $u(x,\ t)$ 之间的相对l_2 误差。M 取前述分析确定的最优值 4。对于每种神经网络架构，我们都运行 Laplace-fPINNs 代码五次。相应的数值结果见表 2，其中包括相对l_2 误差的均值和标准差。从表中可以看出，神经网络的宽度对解的精度影响大于深度。增大神经网络的宽度可略微提高精度。

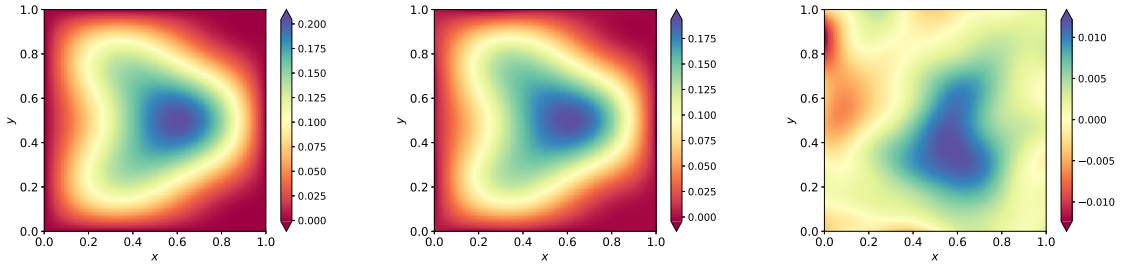
下面，我们通过可视化展示 Laplace-fPINNs 方法求解方程 (1.1) 二维正问题的精度。根据表 1 和表 2 中给出的前述数值结果，我们选择一个每个隐藏层含 512 个神经元的 7 层全连接神经网络来求近似解，并设定 $M\ =4$。图 2 展示了精确解 u、近似解 u_{NN} 及其对应的残差 $u\ -u_{NN}$。从图 2 (c)、(f) 所示的残差可以看出，Laplace-fPINNs 方法得到的近似解与真实值高度吻合。

M	2	4	6
相对l_2 误差	0.0845±0.0158	0.0348±0.0103	0.5212±0.1770

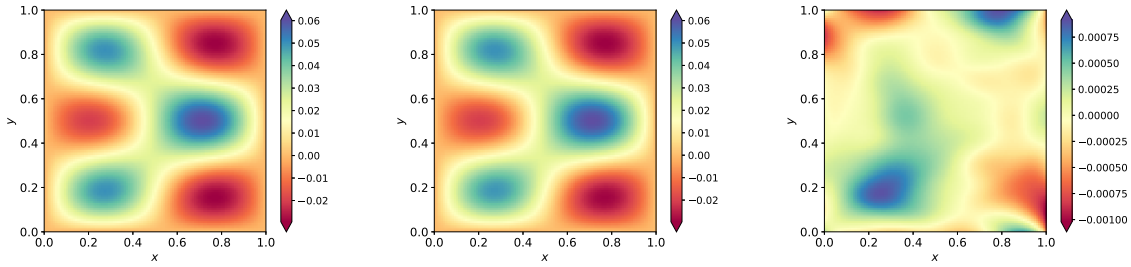
表 1：二维正问题中，不同M下近似解$u_{NN}(x,t;\ \theta^*)$相对l_2 误差的均值和标准差。神经网络的深度和宽度分别为 5 和 256。

深度	宽度		
	128	256	512
5	0.0822±0.0261	0.0348±0.0103	0.0349±0.0084
6	0.0994±0.0388	0.0358±0.0051	0.0261±0.0053
7	0.1055±0.0186	0.0375±0.0129	0.0204±0.0062

表 2：不同神经网络架构的近似解l_2 与参考解u 之间相对误差的均值和标准推导。此处固定$M\ =4$。



(a) 参考解$u_{\circ}$ (b) 近似解$u_{NN\circ}$ (c) $u - u_{NN\circ}$



(d) 参考解$u_{\circ}$ (e) 近似解$u_{NN\circ}$ (f) $u - u_{NN}$.

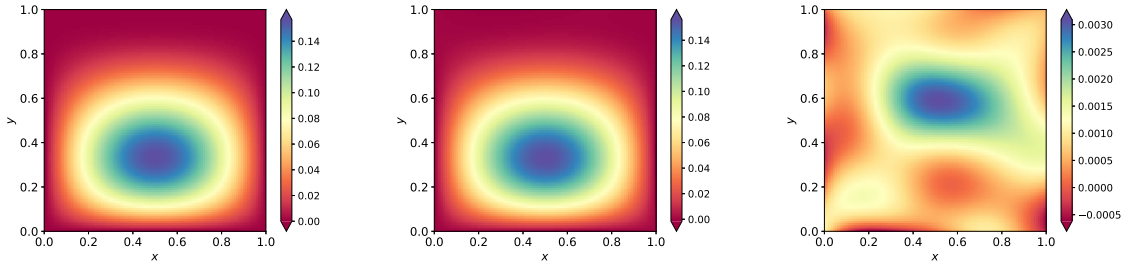
图 2：T = 1 时二维正问题的参考解 u(x, y, t) 与近似解 u_{NN}(x, y, t; θ^*) 的比较。上图分别显示参考解 u(x, y, 0.02)、近似解 u_{NN}(x, y, 0.02; θ^*) 和误差 u(x, y, 0.02) − u_{NN}(x, y, 0.02; θ^*)。下图分别显示参考解 u(x, y, 1)、近似解 u_{NN}(x, y, 1; θ^*) 和误差 u(x, y, 1) − u_{NN}(x, y, 1; θ^*)。

4.1.2. $T = 10$ 时的二维正问题

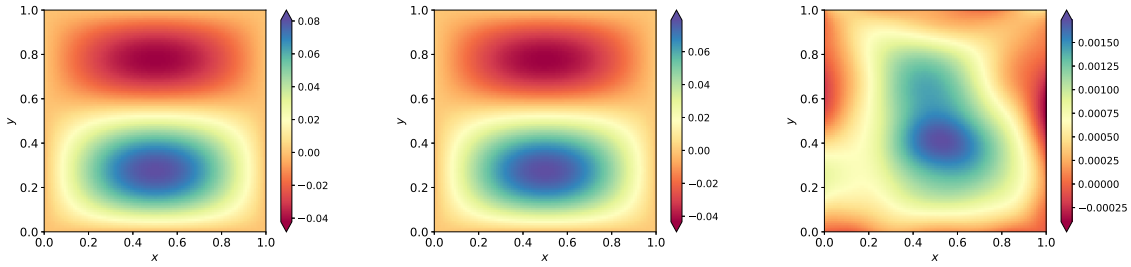
本部分中，我们考虑另一个使用 Laplace-fPINNs 方法求解二维正问题的例子。同样，扩散系数和反应系数分别为 $a(x,y) = 1$、$c(x,y) = 0$。初始值和源项给定如下：

$$u_0(x, y) = 3\sin(\pi x)\sin(\pi y),$$
$$f(x, y) = 3\sin(\pi x)\sin(2\pi y).$$

在本例中，我们考虑区域 $\Omega = [0, 1]^2$ 和$T = 10$。神经网络架构、优化器和采样点设置均与前一例相同。对于通过L_1型有限差分格式近似时间分数阶导数的 fPINNs 方法 [36], , 在较大的T下，最小化损失函数的过程需要更多辅助点，因此计算成本可能显著增加。然而，采用本文提出的 Laplace-fPINNs 方法时，每次迭代的计算成本不会随着T增大而增加。图 3 所示的数值结果表明，即使T较大，Laplace-fPINNs 方法仍能准确求解亚扩散方程 (1.1) 的正问题。



(a) 参考解$u_{\circ}$ (b) 近似解$u_{NN\circ}$ (c) $u - u_{NN}$.



(d) 参考解$u_{\circ}$ (e) 近似解$u_{NN\circ}$ (f) $u - u_{NN}$.

图 3：T = 10 时二维正问题的参考解 u(x, y, t) 与近似解 u_{NN}(x, y, t; θ^*) 的对比。上图分别展示参考解 u(x, y, 0.05)、近似解 u_{NN}(x, y, 0.05; θ^*) 以及误差 u(x, y, 0.05) – u_{NN}(x, y, 0.05; θ^*)。下图分别展示参考解 u(x, y, 10)、近似解 u_{NN}(x, y, 10; θ^*) 以及误差 u(x, y, 10) –u_{NN}(x, y, 10; θ^*)。

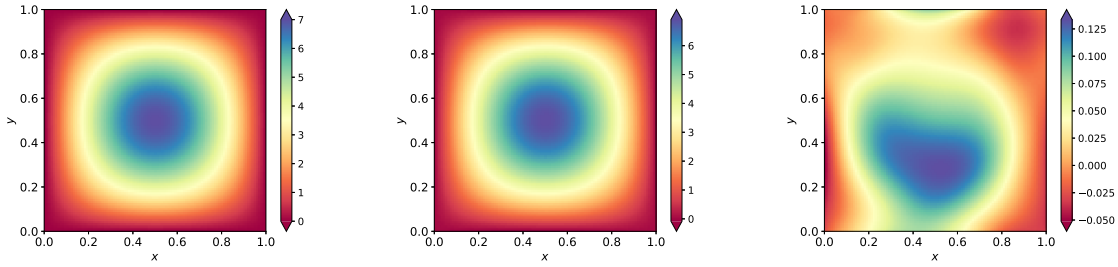
4.2. 三维正问题

本节研究使用 Laplace-fPINNs 方法求解方程 (1.1) 中三维正问题的精度。取区域 $\Omega = [0,1]^3$、$T = 1$、扩散系数$a(x, y, z) = 1$以及反应系数$c(x,y,z) = 0$。制造解取为 $u(x, y,z,t) = (2t + 5)\sin(\pi x)\sin(\pi y)\sin(\pi z)$，相应的初始值和源项为

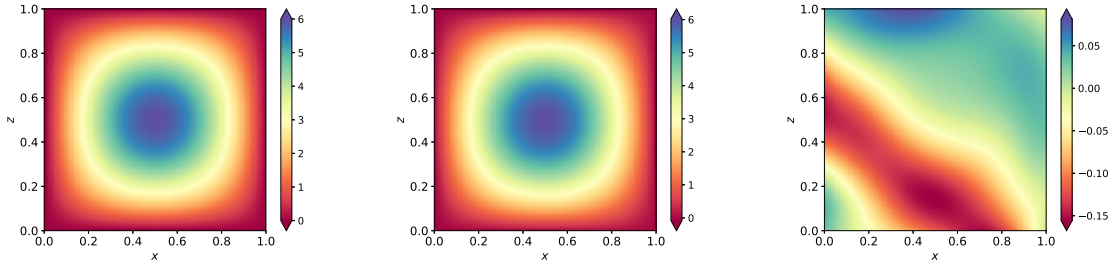
$$u_0(x, y, z) = 5\sin(\pi x)\sin(\pi y)\sin(\pi z),$$

$$f(x, y, z, t) = [\frac{2}{\Gamma(2-\alpha)}t^{1-\alpha} + (2t+5)(3\pi^2 a(x, y, z) - c(x, y, z))]\sin(\pi x)\sin(\pi y)\sin(\pi z).$$

我们选择一个包含 5 层、每层 256 个神经元的全连接神经网络，并将迭代次数设为 1.5×10^5。训练模型得到的数值结果如图 4 所示。图 4 上图展示了精确解u 与近似解u_{NN}的对比，其中固定$t = 1$、$z = 0.5$。类似地，图 4 下图展示了精确解u 与近似解u_{NN}的对比，其中固定$t = 0.5$、$y = 0.5$。从图 4 可以看出，即使对于方程 (1.1) 中的三维正问题，Laplace-fPINNs 方法也能达到较高的精度。



(a) $t = 1$、$z = 0.5$ 处的精确解 (b) $t = 1, z = 0.5$ 处的近似解 = (c) $t = 1$、$z = 0.5$ 处的误差



(d) $t = 0.5$、$y = 0.5$ 处的精确解(e) $t = 0.5、y = 0.5$ 处的近似解 = (f) $t = 0.5$、$y = 0.5$ 处的误差

图 4：三维正问题的参考解 u 与近似解 $u_{NN}(x, y, z, t; \theta^*)$ 的比较。上图分别显示参考解 $u(x, y, 0.5, 1)$、近似解 $u_{NN}(x, y, 0.5, 1; \theta^*)$ 以及误差 $u(x, y, 0.5, 1) - u_{NN}(x, y, 0.5, 1; \theta^*)$。下图分别显示参考解 $u(x, 0.5, z, 0.5)$、近似解 $u_{NN}(x, 0.5, z, 0.5; \theta^*)$ 以及误差 $u(x, 0.5, z, 0.5) - u_{NN}(x, 0.5, z, 0.5; \theta^*)$。

4.3. 反问题

本部分研究使用 Laplace-fPINNs 求解时间分数阶扩散方程 (1.1) 的反问题时的性能。我们考虑一个构造解$u(x,y, z, t) = (3t + 5)\sin(2\pi x)\sin(\pi y)\sin(\pi z)$，并分别取区域 $\Omega = [0, 1]^3$、$T = 1$。根据 Caputo导数的定义，可以解析地计算时间分数阶导数，因此初始值和源项为

$$u_0(x, y, z) = 5\sin(2\pi x)\sin(\pi y)\sin(\pi z),$$

$$f(x, y, z, t) = [\frac{3}{\Gamma(2-\alpha)}t^{1-\alpha} + (3t + 5)(6\pi^2 a(x, y, z) - c(x, y, z))]\sin(2\pi x)\sin(\pi y)\sin(\pi z),$$

其中扩散系数 $a(x, y, z) = 0.5 + e^{-(x+y+z)}$，反应系数 $c(x, y,z) = 0$。本文考虑这样一个反问题：根据对 u 的附加测量，在已知初始值 u_0, 、源项 f 和分数阶阶数 α 的情况下，识别扩散系数 a 和解u ,

并且反应系数 $c = 0$ 已知。具体而言，附加测量数据为

$$h(x,y,z,t) = u(x,y,z,t) + \epsilon u(x,y,z,t), \quad (x,y,z) \in \omega \subset \Omega,\ t \in (0,T),$$

其中我们取 $\epsilon = 0.001$, $\omega = [0.3, 0.7]^3$。

为解决本文所研究的反问题，我们使用神经网络分别对解 u 和扩散系数a进行参数化。具体而言，我们使用一个每层含 256 个神经元的 5 层全连接神经网络，对方程 (3.3) 中的解$\tilde{u}$ 进行参数化，记为 $\tilde{u}_{NN}(x, y, z, s; \theta_1)$。此外，我们将未知扩散系数 a 表示为一个每层含 64 个神经元的 4 层全连接神经网络，记为 $a_{NN}(x,y,z; \theta_2)$。此外，我们假设扩散系数的边界已知。通过最小化以下损失函数，对近似解 θ_1, θ_2的参数 $\tilde{u}_{NN}(x,y,z, s; \theta_1)$以及近似扩散系数$a_{NN}(x,y, z; \theta_2)$进行优化

$$\begin{aligned}(\theta_1^*, \theta_2^*) &= \arg\min L^{lp}(\theta_1, \theta_2) \\ &= \arg\min \{w_{eq} L_{eq}^{lp}(\theta_1, \theta_2) + w_{bd} L_{bd}^{lp}(\theta_1) + w_{obs} L_{obs}^{lp}(\theta_1) + w_{prior} L_{prior}^{lp}(\theta_2)\},\end{aligned} \tag{4.1}$$

其中

$$\begin{aligned}L_{eq}^{lp}(\theta_1, \theta_2) &= \frac{1}{N_r} \sum_{i=1}^{N_r} |r_{NN}^{ip}(x_r^i, y_r^i, z_r^i, s_r^i; \theta_1, \theta_2)|^2, \\ L_{obs}^{lp}(\theta_1) &= \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} |\tilde{u}_{NN}(x_{obs}^i, y_{obs}^i, z_{obs}^i, s_{obs}^i; \theta_1) - h(x_{obs}^i, y_{obs}^i, z_{obs}^i, s_{obs}^i)|^2, \\ L_{prior}^{lp}(\theta_2) &= \frac{1}{N_{prior}} \sum_{i=1}^{N_{prior}} |a_{NN}(x_{prior}^i, y_{prior}^i, z_{prior}^i) - a(x_{prior}^i, y_{prior}^i, z_{prior}^i)|^2.\end{aligned}$$

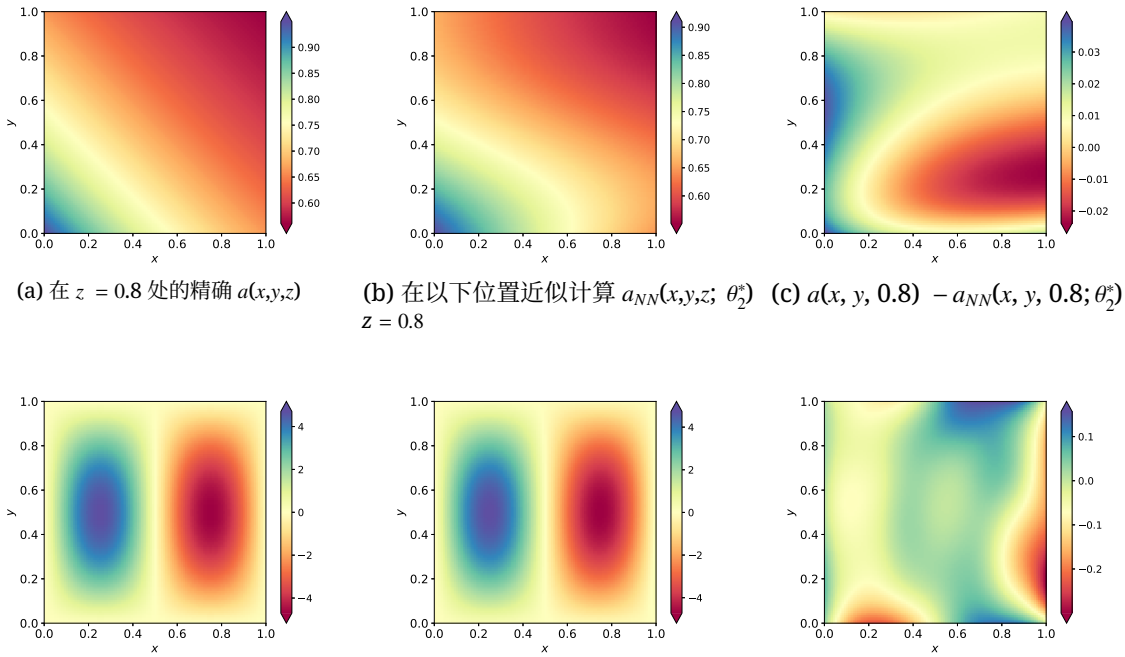
由 (3.4) 定义的 $L^{lp}{}_{bd}(\theta_1)$，以及由下式给出的 $r^{ip}{}_{NN}(x^r, y_r, z_r, s_r; \theta_1, \theta_2)$

$$\begin{aligned}r_{NN}^{ip}(x,y,z,s;\theta_1,\theta_2) := {} & s^{\alpha}\tilde{u}_{NN}(x,y,z,s;\theta_1) - \nabla \cdot (a_{NN}(x,y,z;\theta_2)\nabla \tilde{u}_{NN}(x,y,z,s;\theta_1)) \\ & - c(x,y,z)\tilde{u}_{NN}(x,y,z,s;\theta_1) - s^{\alpha-1}u_0(x,y,z) - \tilde{f}(x,y,z,s).\end{aligned}$$

N_{obs} 和 N_{prior} 分别表示与测量数据对应的观测点批次大小，以及扩散系数 $a(x, y, z)$ 的已知边界点批次大小。每次迭代中，我们从 Ω 和 $\partial\Omega$ 上的均匀分布中分别随机生成 1000 个残差点和 400^3 边界点。此外，在训练期间，我们固定观测点 $N_{obs} = 31^3$ 和先验点 $N_{prior} = 51^3$。式 (4.1) 中的权重系数取为 $w_{eq} = 1$, $w_{bd} = 2000$, $w_{obs} = 1000$,$w_{prior} = 100$。

在 1.5×10^5之后 经过训练轮次的训练后，训练所得模型的数值结果

如图 5 所示。具体而言，图 5 (b) 展示了 z =0.8 处重建的扩散系数，与图 5 (a) 中所示的精确扩散系数一致。此外，图 5 (c) 展示了 $z = 0.8$ 处重建的扩散系数 $a_{NN}(x,y,z;\ \theta_2^*)$ 与精确扩散系数 $a(x,y,z)$ 之间的差异。误差较小，表明 Laplace-fPINNs 方法能够有效识别三维扩散系数。图 5 第二行比较了 $z = 0.8$、$t = 1$ 处的精确解 $u(x,y,z,t)$ 与近似解 $u_{NN}(x,y,z,t;\ \theta_1^*)$。结果表明，通过纳入额外的测量数据$h(x,y,z,t)$，Laplace-fPINNs 方法能够同时重建扩散系数a 并预测解u 。



(a) 在 $z = 0.8$ 处的精确 $a(x,y,z)$　(b) 在以下位置近似计算 $a_{NN}(x,y,z;\ \theta_2^*)$ $z = 0.8$　(c) $a(x,y,0.8) - a_{NN}(x,y,0.8;\theta_2^*)$

(d) 在 $z = 0.8$、$t = 1$ 处的精确 $u(x,y,z,t)$；(e) 在 $z = 0.8$、$t = 1$ 处的 $u_{NN}(x,y,z,t;\ \theta_1^*)$；(f) $u(x,y,0.8,1) - u_{NN}(x,y,0.8,1;\ \theta_1^*)$

图 5：上图展示了在 $z = 0.8$ 处，精确扩散系数 $a(x,y,z)$ 与近似扩散系数 $a_{NN}(x,y,z;\ \theta_2^*)$ 的比较。下图展示了在 $z = 0.8$、$t = 1$ 处，参考解$u(x,y,z,t)$ 与近似解 $u_{NN}(x,y,z,t;\ \theta_1^*)$ 的比较。

5. 结论

本文提出了一种用于求解亚扩散正问题与逆问题的深度学习方法。通过结合时间分数阶导数的拉普拉斯变换、物理信息神经网络和拉普拉斯变换数值逆变换，我们提出了 Laplace-fPINNs 方法，可从时间分数阶扩散方程 (1.1) 推断解。该方法从根本上避免了使用 [36]中引入的辅助点，这些辅助点原本用于离散时间分数阶 Caputo 导数。一系列数值结果表明，该方法可用于求解亚扩散正问题。此外，我们还将该方法用于求解一个反问题

该问题旨在确定三维扩散系数。由于问题维度较高，传统反演技术难以求解。最后，我们指出，Laplace-fPINNs 方法的思想可轻松扩展至物理信息算子学习方法，用于求解亚扩散方程 (1.1)。

致谢 本工作得到国家重点研发计划项目（编号：2022YFA1008200（Z. X.）和编号：2020YFA0712000（Z. M.））、上海市扬帆计划（Z. X.）、上海市自然科学基金项目（编号：20ZR1429000（Z. X.））、国家自然科学基金项目（编号：62002221（Z. X.））、国家自然科学基金项目（编号：12101401（Z. M.））、国家自然科学基金项目（编号：12031013（Z. M.））、上海市科技重大专项（编号：2021SHZDZX0102）的资助，并获得数学科学学院高性能计算和学生创新中心，以及由上海交通大学高性能计算中心支持的思源一号集群的计算支持。

参考文献

[1] B. Berkowitz、A. Cortis、M. Dentz 和 H. Scher。将地质构造中的非菲克输运建模为连续时间随机游走。《地球物理学评论》，44(2)，2006年。[2] M. Dentz、A. Cortis、H. Scher 和 B. Berkowitz。非均质介质中溶质输运的时间行为：从反常输运向正常输运的转变。《水资源进展》，27(2):155–173，2004年。[3] R. R. Nigmatullin。在具有分形几何的介质中实现广义传递方程。《物理学现状（*B*辑）》，133(1):425–430，1986年。[4] S. C. Kou。纳米尺度生物物理学中的随机建模：蛋白质内的亚扩散。《应用统计年鉴》，2(2):501–535，2008年。[5] K. Ritchie、X. Y. Shan、J. Kondo、K. Iwasawa、T. Fujiwara 和 A. Kusumi。通过单分子追踪检测细胞膜中的非布朗扩散。《生物物理学杂志》，88(3):2266–2277，2005年。[6] B. Berkowitz、J. Klafter、R. Metzler 和 H. Scher。非均质介质中输运的物理图景：平流-弥散、随机游走和分数阶导数表述。《水资源研究》，38(10):9–1–9–12，2002年。[7] J. P. Bouchaud 和 A. Georges。无序介质中的反常扩散：统计机制、模型及物理应用。《物理报告》，195(4):127–293，1990年。

[8] R. Metzler 和 J. Klafter。反常扩散随机游走指南：分数阶动力学方法。《物理报告》，339(1):1–77，2000年。[9] T. Graepel。利用高斯过程求解含噪线性算子方程：在常微分方程和偏微分方程中的应用。载于第二十届国际机器学习会议论文集，第234–241页，2003年。[10] M. Raissi、P. Perdikaris 和 G. E. Karniadakis。用于时变和非线性偏微分方程的数值高斯过程。《*SIAM*科学计算杂志》，40(1):A172–A198，2018年。[11] T. Lee、I. Bilionis 和 A. B. Tepole。利用多保真高斯过程回归，传播生长组织力学和生物学响应中的不确定性。《计算方法与应用力学和工程》，359:112724，2020年。[12] I.E. Lagaris、A. Likas 和 D.I. Fotiadis。用于求解常微分方程和偏微分方程的人工神经网络。《*IEEE*神经网络汇刊》，9(5):987–1000，1998年。[13] M. Raissi、P. Perdikaris 和 G. E. Karniadakis。物理信息神经网络：用于求解涉及非线性偏微分方程的正问题与逆问题的深度学习框架。《计算物理学杂志》，378:686–707，2019年。[14] L. Lu、X. H. Meng、Z. P. Mao 和 G. E. Karniadakis。DeepXDE：用于求解微分方程的深度学习库。《*SIAM*评论》，63(1):208–228，2021年。[15] W. E 和 B. Yu。深度Ritz方法：一种用于求解变分问题的深度学习数值算法。《数学与统计学通讯》，6(1):1–12，2018年。[16] Y. Zang、G. Bao、X. Ye 和 H. Zhou。用于高维偏微分方程的弱对抗网络。《计算物理学杂志》，411:109409，14，2020年。[17] G. Bao、X. Ye、Y. Zang 和 H. Zhou。利用弱对抗网络求解反问题。《反问题》，36(11):115003，31，2020年。[18] L. Lu、P. Jin 和 G. E. Karniadakis。DeepONet：基于算子通用逼近定理学习非线性算子以识别微分方程。*arXiv*预印本 *arXiv:1910.03193*，2019年。[19] P. Jin、S. Meng 和 L. Lu。MIONet：通过张量积学习多输入算子。*arXiv*预印本 *arXiv:2202.06137*，2022年。[20] Y. Zhu 和 N. Zabaras。用于代理建模和不确定性量化的贝叶斯深度卷积编码器-解码器网络。《计算物理学杂志》，366:415–447，2018年。

[21] Z. Li、N. B. Kovachki、K. Azizzadenesheli、B. liu、K. Bhattacharya、A. Stuart 和 A. Anand- kumar。用于参数偏微分方程的傅里叶神经算子。载于国际学习表征会议，2021年。[22] Z. Li、N. B. Kovachki、K. Azizzadenesheli、B. Liu、K. Bhattacharya、A. Stuart 和 A. Anand- kumar。神经算子：用于偏微分方程的图核网络。*arXiv* 预印本 *arXiv:2003.03485*，2020年。[23] Z. Li、N. B. Kovachki、K. Azizzadenesheli、B. Liu、K. Bhattacharya、A. Stuart 和 A. Anand- kumar。用于参数偏微分方程的多极图神经算子。载于第*34*届神经信息处理系统大会论文集，2020年。[24] L. Zhang、T. Luo、Y. Zhang、W. E、Z. J. Xu 和 Z. Ma。Mod-net：一种通过模型-算子-数据网络求解偏微分方程的机器学习方法。《计算物理学通讯》，32(2):299–335，2022年。[25] X. Yan、Z. J. Xu 和 Z. Ma。利用神经算子进行贝叶斯反演（bino），用于亚扩散建模：正问题与反问题。*arXiv:2211.11981*，2022年。[26] Y. Liao 和 P. Ming。深度尼采法：带本质边界条件的深度Ritz方法。《计算物理学通讯》，29(5):1365–1384，2021年。[27] Q. Lou、X. Meng 和 G. E. Karniadakis。通过玻尔兹曼-BGK公式，利用物理信息神经网络求解流动正问题与反问题。《计算物理学杂志》，447：论文编号110676，20页，2021年。[28] E. Kharazmi、Z. Zhang 和 G. E. Karniadakis。*hp*-VPINNs：结合区域分解的变分物理信息神经网络。《计算方法与应用力学和工程》，374：论文编号113547，25页，2021年。[29] L. Yang、X. Meng 和 G. E. Karniadakis。B-PINNs：用于含噪声数据的偏微分方程正问题与反问题的贝叶斯物理信息神经网络。《计算物理学杂志》，425：论文编号109913，23页，2021年。[30] K. Shukla、A. D. Jagtap 和 G. E. Karniadakis。通过区域分解实现并行物理信息神经网络。《计算物理学杂志》，447：论文编号110683，19页，2021年。[31] Z. Mao、L. Lu、O. Marxen、T. A. Zaki 和 G. E. Karniadakis。用于高超声速问题的 DeepM&Mnet：利用算子的神经网络近似，预测正激波后的耦合流动和有限速率化学反应。《计算物理学杂志》，447：论文编号110698，24页，2021年。

[32] J. Sirignano 和 K. Spiliopoulos。DGM：一种求解偏微分方程的深度学习算法。《计算物理学杂志》，375:1339–1364，2018年。[33] Y. Shang、F. Wang 和 J. Sun。用于求解偏微分方程的深度 Petrov-Galerkin 方法。*arXiv* 预印本 *arXiv:2201.12995*，2022年。[34] Z. Liu、W. Cai 和 Z. J. Xu。用于复杂区域泊松-玻尔兹曼方程求解的多尺度深度神经网络（mscalednn）。《计算物理学通讯》，28(5):1970–2001，2020年。[35] X. Li、Z. J. Xu 和 L. Zhang。用于多重尺度非线性椭圆方程的多尺度深度神经网络算法。《计算物理学通讯》，28(5):1886–1906，2020年。[36] G. Pang、L. Lu 和 G. E. Karniadakis。分数阶物理信息神经网络：分数阶物理信息神经网络。《*SIAM*科学计算杂志》，41(4):A2603–A2626，2019年。[37] L. Guo、H. Wu、X. Yu 和 T. Zhou。蒙特卡罗分数阶物理信息神经网络：求解涉及高维分数阶偏微分方程的正问题与反问题的深度学习方法。《计算方法与应用力学和工程》，400：论文编号115523，17页，2022年。[38] A. G. Baydin、B. A. Pearlmutter、A. A. Radul 和 J. M. Siskind。机器学习中的自动微分：综述。《机器学习研究杂志》，18(1):5595–5637，2017年。[39] B. Davies 和 B. Martin。拉普拉斯变换的数值反演：方法综述与比较。《计算物理学杂志》，33(1):1–32，1979年。[40] J. Abate 和 W. Whitt。拉普拉斯变换数值反演的统一框架。《*INFORMS*计算杂志》，18(4):408–421，2006年。[41] D. P. Gaver。观测随机过程及近似变换反演。《运筹学》，14(3):444–459，1966年。[42] Z. Z. Sun 和 X. Wu。扩散波系统的全离散差分格式。《应用数值数学》，56(2):193–209，2006年。[43] Z. J. Fu、W. Chen 和 H. T. Yang。用于拉普拉斯变换时分数阶扩散方程的边界粒子法。《计算物理学杂志》，235:52–66，2013年。[44] C. S. Chen、Y. F. Rashed 和 M. A. Golberg。线性扩散方程的无网格方法。《数值传热学》，B辑：469–486，1998年6月。