

Numerical Methods for Porous Media Flow Equations and their Coupled Equations

A thesis submitted to
Northwest University
in partial fulfillment of the requirements
for the degree of Master
in Applied Mathematics

By
Supervisor:
June 2024

西北大学学位论文知识产权声明书

本人完全了解西北大学关于收集、保存、使用学位论文的规定。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。本人授权西北大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时授权中国科学技术信息研究所等机构将本学位论文收录到《中国学位论文全文数据库》或其它相关数据库。

保密论文待解密后适用本声明。

学位论文作者签名：_____指导教师签名：_____

年 月 日

年 月 日

西北大学学位论文独创性声明

本人声明：所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，本论文不包含其他人已经发表或撰写过的研究成果，也不包含为获得西北大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

年 月 日

摘要

智能求解算法相对于传统方法, 有着计算成本低、计算速度快等优点. 本文通过极限学习机求解二维混合形式的多孔介质流方程、非定常混合形式的多孔介质流方程; 通过物理信息神经网络, 求解流体在自由流区域和多孔介质区域耦合流动方程. 首先对于极限学习机, 我们结合区域分解, 给出极限学习求解偏微分方程的具体过程, 通过二维混合形式的多孔介质流方程的数值算例对数值方法进行验证; 其次, 我们考虑三维求解域, 通过非定常混合形式的多孔介质流方程的数值算例对数值方法进行验证, 并且分析了影响网络误差精度的网络设定. 最后对于物理信息神经网络, 我们运用区域分解的思想将问题划分为多孔介质流区域和自由流区域, 将时间维度进行剖分, 在每个小剖分上将物理信息数据作为约束条件对网络进行训练, 通过数值算例验证物理信息神经网络求解耦合方程的可行性.

关键词

物理信息神经网络, 极限学习机, 自由流区域和多孔介质区域耦合流动方程

Abstract

Intelligent solving algorithms have advantages such as low computational cost and fast computation speed compared to traditional methods. In this paper, we use Extreme Learning Machines (ELM) to solve the two-dimensional mixed form of porous media flow equations and the unsteady mixed form of porous media flow equations. They also employ Physics-Informed Neural Networks (PINNs) to solve the coupled flow equations in both the free-flow region and the porous media region.

Regarding ELM, the authors combine it with domain decomposition to provide a specific process for solving partial differential equations using ELM. They validate the numerical methods using numerical examples of the two-dimensional mixed form of porous media flow equations. Subsequently, they extend the analysis to a three-dimensional solution domain, verify the numerical methods using numerical examples of the unsteady mixed form of porous media flow equations, and analyze the network settings that affect the accuracy of the network's error.

Regarding the PINNs approach, the authors employ the idea of domain decomposition to divide the problem into porous media flow regions and free-flow regions. They further divide the time dimension and use physical information data as constraints to train the network on each small subdivided portion. The feasibility of using PINNs to solve the coupled equations is verified through numerical examples.

Keywords

Physics-Informed Neural Networks, Extreme Learning Machines, the coupled flow equations in both the free-flow region and the porous media region

目 录

| | |
|------------------------------|----|
| 中文摘要 | i |
| 英文摘要 | ii |
| 第一章 绪论 | 1 |
| §1.1 研究背景 | 1 |
| §1.2 研究现状 | 2 |
| §1.2.1 多孔介质流方程 | 2 |
| §1.2.2 流体在自由流区域和多孔介质区域耦合流动方程 | 2 |
| §1.2.3 智能求解算法 | 3 |
| §1.2.4 本文结构 | 4 |
| 第二章 极限学习求解混合形式的多孔介质流方程 | 5 |
| §2.1 模型介绍 | 5 |
| §2.2 极限学习的数值格式 | 5 |
| §2.3 带有区域分解的极限学习的数值格式 | 7 |
| §2.4 数值实验 | 8 |
| §2.4.1 数值算例 1 | 9 |
| §2.4.2 数值算例 2 | 10 |
| §2.5 本章总结 | 11 |
| 第三章 极限学习求解非定常的混合多孔介质流方程 | 13 |
| §3.1 模型介绍 | 13 |
| §3.2 带有区域分解的极限学习的数值格式 | 13 |
| §3.3 数值实验 | 15 |
| §3.3.1 数值算例 1 | 16 |
| §3.3.2 数值算例 2 | 16 |
| §3.4 本章总结 | 18 |

| | |
|---------------------------------------|----|
| 第四章 物理信息神经网络 (PINNs) 处理Stokes-Darcy方程 | 19 |
| §4.1 模型介绍 | 19 |
| §4.1.1 Stokes方程 | 19 |
| §4.1.2 Darcy方程. | 19 |
| §4.1.3 交界面条件 | 20 |
| §4.1.4 数值格式 | 20 |
| §4.2 PINNs方法 | 21 |
| §4.3 数值算例 | 26 |
| §4.3.1 数值算例 1 | 27 |
| §4.3.2 数值算例 2 | 30 |
| 参考文献 | 35 |
| 攻读硕士学位期间取得的科研成果 | 41 |
| 致谢 | 42 |

第一章 绪论

§1.1 研究背景

数值计算是指以获得数值结果为目标计算, 数值计算的效率和计算结果的精确度往往依赖于合适的数值方法. 人们使用数值方法解决实际问题的历史可以追溯到距今四千年前的古代.

古罗马数学家和天文学家面对诸如天体运动、土地测量、贸易计算等实际问题时, 已经开始使用一些近似计算的方法. 他们提出了一些与数值方法相关的概念和技术, 虽然在当时没有计算机, 但是他们通过手工计算和观察, 已经能够得出一些近似解并用于实际应用中.

到十七世纪中叶, 随着数学分析的发展, 给数值方法的发展提供了大量的理论基础, 随后, Isaac Newton、Gottfried Wilhelm Leibniz 等数学家发表基本的微积分理论, 为后续数学方法提供了重要的理论基础. 到二十世纪, 随着计算机的使用和计算科学的发展, 数值计算方法的计算精度和计算速度得到了提升, 有限元法、有限体积法、间断伽辽金法开始使用在流体动力学、固体力学、电磁学等方向.

近年来, 随着机器学习的不断发展, 其应用范围逐渐扩展至求解 PDE. 具体而言, 机器学习求解 PDE 的历史可以追溯到 20 世纪 90 年代, 当时研究人员开始尝试使用神经网络来模拟和解决复杂的物理问题. 例如, Lagaris 等人^[21]首次提出了使用神经网络作为有限差分法的替代方法来解决 PDE. 然而, 这种方法在当时并未得到广泛的认可和应用. 随着时间的推移, 以及计算能力的提升和大数据的普及, 机器学习在求解 PDE 方面的应用才得到了广泛的关注和研究. 特别是深度学习方法的出现, 使得机器学习能够更有效地处理非线性问题和复杂的数据结构. 例如, Justin 和 Konstantinos^[23]提出了一种基于深度神经网络的算法, 用于求解非线性 PDE. 目前, 机器学习已经成为求解 PDE 的一种重要工具, 被广泛应用于各种科学和工程领域. 例如, Karniadakis 等人^[8]对机器学

习在流体力学、固体力学和化学反应中的应用进行了全面的综述.

§1.2 研究现状

§1.2.1 多孔介质流方程

流体在多孔介质区域流动模型 (Darcy) 方程, 可以追溯到十九世纪法国工程师 Henry Darcy 提出的描述多孔介质中流体渗透规律的 Darcy 定律. 在过去的几十年中, Darcy 方程被广泛应用于水资源管理、石油工程和气藏开发、岩石力学和地质工程等领域.

对于 Darcy 方程的理论部分, 目前已有大量的研究成果: Feng^[36] 给出了二维 Darcy 方程的弱解存在性; Chen^[37] 给出了非定常 Darcy 方程的弱解存在性. 随后, 有大量数值方法对二维、非定常 Darcy 方程进行数值求解: Hughes^[38] 通过 Mixed Discontinuous Galerkin(MDG) 方法对二维 Darcy 方程进行求解; Dawson^[40]通过改进的 Local Discontinuous Galerkin(LDG) 对二维 Darcy 方程进行求解, 并给出了全新弱形式; Bernardo Cockburn^[41] 通过 LDG 方法, 对非定常 Darcy 方程进行求解. 但是后续研究中, 模型情况可能会变得复杂, Michael^[39] 通过改变常数项系数分析系数对模型的影响; Qian Yanxia^[42] 通过 MDG 求解了不可压三维 Darcy 方程. 这些数值方法大多都是通过单纯形网格对方程进行求解, 求解时, 对资源的要求较高. 在第 2 章和 第 3 章中, 我们通过深度学习方法对两类多孔介质流方程进行求解.

§1.2.2 流体在自由流区域和多孔介质区域耦合流动方程

流体在自由流区域和多孔介质区域耦合流动 (Stokes–Darcy) 模型, 可以追溯到十九世纪的爱尔兰物理学家 George Gabriel Stokes 提出的用于描述粘性流体运动的 Stokes 方程和法国工程师 Henry Darcy 提出的描述多孔介质中流体渗透规律的 Darcy 定律. 在过去的十几年中, Stokes–Darcy 耦合模型, 受到数学界、物理学界和医学界越来越多的关注, 同时此模型被广泛应用于工业、

医学界的众多应用中, 例如地下水和地表水的相互作用、地下石油勘探与建模、血液流动模拟与血管建模、流体物料交换等问题.

针对 Stokes–Darcy 耦合问题的数值方法, 目前已有了大量的研究成果, 如: 有限元法 (FEM) ^[1]、间断Galerkin法 (DG Method)^[2]、多重网格法 ^[3]、最小二乘法 ^[7]、边界积分法 (BIM)^[9]等. 但是在实际问题中, 模型可能要考虑外在压力、物理层面的断层, 以上数值方法大多以单纯形网格进行剖分, 可能无法满足过于复杂的条件. 所以为了提高计算精度. 近些年来, 出现了很多可以对求解域采用一般形状的网格剖分、并且可以使用复杂物理现象的求解方法, 例如: 杂交高阶方法 (HHO)^[11]、虚单元方法 (VEM)^[13]、弱 Galerkin(WG) 方法 ^[14, 15]、杂交间断 Galerkin 方法 (HDG)^[19]、混合有限体积法 (MHFV)^[17, 20], 但是以上方法同样依赖剖分的网格. 因此, 我们在第 4 章中, 通过深度学习方法对耦合模型进行求解.

§1.2.3 智能求解算法

神经网络 (DNN) 在许多科学和工程领域取得了巨大的成果. 例如自然语言处理、机器人移动算法、计算机视觉、语音和图像识别. 在过去的几年中, 神经网络被计算数学界运用在数值方法中, 这是与传统方法不同数值解法. DNN 方法思想是构造一个函数去逼近由实际问题总结出的函数, 因此可以很自然的将其运用对 PDE 或者 ODE 的求解 ^[4–6]. 过去几年出现了几个成功的基于 DNN 的 PDE 求解算法, 例如深度伽辽金方法 (DGM)^[43, 44]、constrained integration ^[45]. 基于 DNN, Raissi 等人 ^[22]提出了一种基于深度学习的框架, 用于学习 PDE 的解, 这种方法被称为“物理信息神经网络”(即 PINNs). PINNs 方法的核心思想是通过提炼方程的初值、边值等方程数据, 作为约束条件引入损失函数中, 从数据中学习出方程的隐藏规律, 得到 PDE 的近似解. 因此可以处理有着复杂物理、几何规律的以及存在多个物理方程或非线性方程的模型. 此外, 给出另一种基于 DNN ^[46] 的数值方法极限学习机 (ELM), 这种方

法在一定程度上也参照了传统有限元方法, 通过构造需要求解的 PDE 的系数矩阵, 来求解目标 PDE. 本文使用 PINNs 和 ELM 对流体在自由流区域和多孔介质区域耦合流动和多孔介质流方程进行求解.

§1.2.4 本文结构

本文具体的安排如下: 在第 2 章中, 系统介绍了二维混合多孔介质流模型, 同时使用带有区域分解的极限学习方法对方程进行求解, 并给出了一些数值算例; 第 3 章中, 通过带有区域分解的极限学习方法对非定常混合形式的多孔介质流方程进行求解, 同时也给出一些数值算例; 第 4 章中, 系统介绍了非稳态 Stokes–Darcy 耦合模型, 并给出了区域分解结合 PINNs 求解耦合问题的数值格式, 同时也通过一些数值算例来验证所提数值格式的有效性.

第二章 极限学习求解混合形式的多孔介质流方程

本章主要通过极限学习 (ELM) 求解混合形式的多孔介质流 (Darcy) 方程, 首先我们对二维空间求解域 Ω 进行区域分解, 随后在每个区域剖分上建立一个“浅”层网络. 此外, 输出层只赋予权重, 且输出层的权重是我们的训练参数. 随后得到 Darcy 方程的系数矩阵. 通过系数矩阵就能得到系数矩阵的最小二乘解, 这就是输出层权重, 这样就能得到 Darcy 方程的数值格式, 最后通过数值算例验证方法的精确性.

§2.1 模型介绍

混合形式的 Darcy 方程求解域 Ω 为开的且是有界的, 并且 Ω 是 Lipschitz 连续的. 本章考虑有着 Dirichlet 边值问题的混合形式 Darcy 方程:

$$\mathbf{u} = -\mathbb{K}\nabla p, \quad \text{in } \Omega \quad (2.1a)$$

$$\nabla \cdot \mathbf{u} = \mathbf{f}, \quad \text{in } \Omega \quad (2.1b)$$

$$p = g, \quad \text{on } \partial\Omega \quad (2.1c)$$

其中 p 为流体压力, \mathbf{u} 为流体速度; g 为给出的 Dirichlet 边界条件; $\partial\Omega$ 为求解域边界; \mathbb{K} 为渗透率矩阵, 这是一个常数矩阵; ∇ 为梯度算子; $\nabla \cdot$ 为散度算子.

§2.2 极限学习的数值格式

与传统的有限元方法、有限差分法、龙格库塔方法不同的是, ELM 方法是建立人工神经网络对目标 PDE 进行逼近的机器学习算法 [33]. 与其他机器学习方法类似, ELM 方法也存在着输入空间和输出空间. 对于 ELM, 将空间自变量 (x, y) 作为输入空间. 随后定义 $\mathbf{u} := (u_1, u_2)$ 、 p 为网络的输出. 我们对每个网络进行分析, 图 2.1 是我们给出的 ELM 方法流程图.

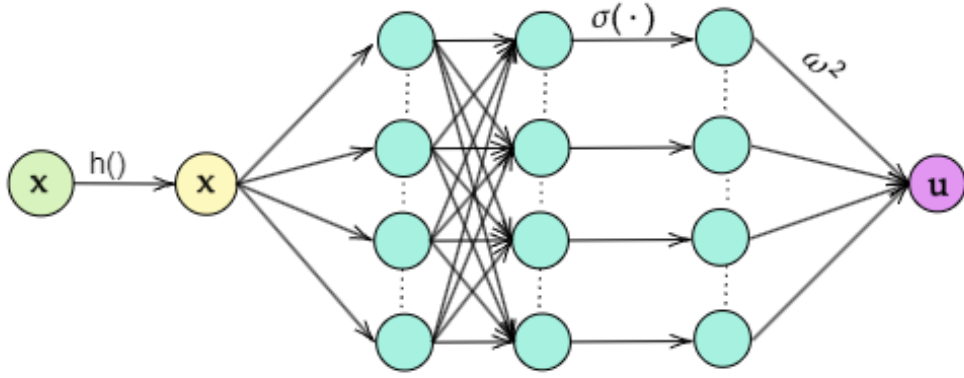


图 2.1: 极限学习流程图

首先, 在训练前, 对传入数据进行仿射变换, 把传入数据通过仿射变换函数 $h(\cdot)$, 这样能把传入数据均匀的分布在 $[-1, 1]$ 上. 随后假设神经网络的第 k 层有 i 个节点, 第 $k-1$ 层有 j 个节点, 令第 $k-1$ 层的输入为 $\phi_h(\mathbf{x})$ ($0 < m < i$), 第 k 层的输出为 $\varphi_g(\mathbf{x})$ ($0 < n < j$), 那么第 k 层的输出可以表示为:

$$\varphi_g(\mathbf{x}) = \sigma\left(\sum_{h=1}^i \phi_h(\mathbf{x})\omega_{ij} + b_j\right) \quad (2.2)$$

其中 ω_{ij} 、 b_j 为权重和偏置, $\sigma(\cdot)$ 为可以更换的激活函数. 接着, 我们约定所有隐藏层的权重和偏置被随机生成在 $[-R_m, R_m]$ ($R_m > 0$) 上. 因此, 网络输出 \mathbf{u}_h 可被表示为:

$$\mathbf{u}_h = \sum_{h=1}^i V_h(\mathbf{x})\omega_{ji}^2, \quad (2.3)$$

其中, $V_h(\mathbf{x})$ 为最后一个隐藏层的输出, ω_{ji}^2 为训练参数, i 为最后一个隐藏层的节点数. 需要注意的是, 最后一个隐藏层的输出 $V_h(\mathbf{x})$ 是已知函数, 并且在整个求解过程中是固定的. 同时 $V_h(\mathbf{x})$ 对输入 \mathbf{x} 求一阶、二阶导数可以通过自微分来实现. 由于源项方程 \mathbf{f} 是已知的. 因此, 训练参数 ω_{ji}^2 可以通过最小二乘计算进行训练以及确定.

§2.3 带有区域分解的极限学习的数值格式

为简化求解过程, 我们对求解域进行区域分解. 将求解域 $\Omega = [x_1, x_2] \times [y_1, y_2]$ 划分为 N 个不重叠的矩形子域. 令 N_x 、 N_y 为求解域 x 、 y 方向上的子域数量, 把子域的横纵坐标存成两个向量 $[x_0, x_1, \dots, x_N]$ 、 $[y_0, y_1, \dots, y_N]$. 因此, $N = N_x \times N_y$, 对任意一个子域 $\omega_{m,n}$ ($0 < m < N_x, 0 < n < N_y$), 其实际物理坐标为 $\Omega_{m,n} = [x_m, x_{m+1}] \times [y_n, y_{n+1}]$. 对于任意一个子域 $\omega_{m,n}$ 上建立两个网络, 其中一个网络用来输出流体速度 u_m , 其输入层具有两个节点, 以用来输入训练点数值, 有一个隐藏层, 有两个输出节点的输出层; 另一个网络用来输出流体压力 p , 其输入层也是具有两个节点, 同时也只有一个隐藏层, 但是只有一个输出节点的输出层. 同时, 输出层只赋予权重不赋予偏置. 并且, 所有隐藏层的权重和偏置都被随机生成在 $[-R_m, R_m]$ ($R_m > 0$). 因此在子域 $\Omega_{m,n}$ 上, 得到类似于公式 2.3 的输出:

$$u^{\Omega_{m,n}}(x, y) = \sum_{h=1}^i V_h^{u, \Omega_{m,n}}(x, y) (\omega_h^{u_1, \Omega_{m,n}}, \omega_h^{u_2, \Omega_{m,n}}) \quad (2.4a)$$

$$p^{\Omega_{m,n}}(x, y) = \sum_{h=1}^i V_h^{p, \Omega_{m,n}}(x, y) \omega_h^{p, \Omega_{m,n}}, \quad (x, y) \in \Omega_{m,n} \quad (2.4b)$$

其中 $\omega_m^{u_1, \Omega_{m,n}}$ 、 $\omega_m^{u_2, \Omega_{m,n}}$ 、 $\omega_m^{p, \Omega_{m,n}}$ 为子域 $\Omega_{m,n}$ 的训练参数, 在公式 2.4 中, $V_m^{u, \Omega_{m,n}}$ 、 $V_m^{p, \Omega_{m,n}}$ 是已知的, 网络中隐藏层的权重和偏置都是控制在 $[-R_m, R_m]$ 中的. 注意, 每个网络在隐藏层前, 仿照极限学习的数值格式加入一个仿射变换层, 将 $[x_m, x_{m+1}] \times [y_n, y_{n+1}]$ 归一化到 $[-1, 1] \times [-1, 1]$ 上.

对于子域 $\Omega_{m,n}$, 将 $[x_m, x_{m+1}]$ 、 $[y_n, y_{n+1}]$ 划分为 Q 段, 令 $(x_{p,eq}^{\Omega_{m,n}}, y_{q,eq}^{\Omega_{m,n}})$ 为 $\Omega_{m,n}$ 上的一组训练点, 其中 $0 < p, q < Q$. 这样就得到了子域 $\Omega_{m,n}$ 上的一组训练点, 训练点总数为 $Q \times Q$. 注意, 以上给出的是在子域 $\Omega_{m,n}$ 训练点服从均匀分布的情况, 此外训练点也可以服从随机分布和正交分布. 本文不对服从随机分布和正交分布进行讨论.

假设子域 Ω_a 和子域 Ω_b 有公共(内部)边界 τ_{ab} . 在内部边界上, 需要考虑源项 f_a 和 f_b 的 C^k 连续性, 即网络训练的数值和 PDE 沿该方向的 K 阶

方向导数的数值要保证数值上是相等的. 此外, 对于求解域 Ω 的 Dirichlet 边界, 流体压力网络的输出值需要满足模型的 Dirichlet 条件. 子域内部边界和 Dirichlet 边界训练点的选取以及网络的建立与在子域 Ω_{mn} 上类似.

现阶段很多科学软件, 例如 MATLAB、Python, 都可以计算方程组的最小二乘解. 在此, 我们使用 python 中的 SciPy 库或者 PyTorch 库来计算.

§2.4 数值实验

本节通过几个数值算例对带有区域分解的 ELM 方法进行验证. 此外, 带有区域分解的 ELM 方法的代码基于 PyTorch 库, 代码在 AMD Ryzen 7 5800H 上运行, 运行内存为 16G.

带有区域分解的 ELM 方法的有效性通过计算数值解与精确解的最大误差和均方误差来验证, 公式 2.5 给出了最大误差和均方误差的计算公式

$$l_{max} = \max(|u^{ELM}(x_n, y_n) - u(x_n, y_n)|) \quad (2.5a)$$

$$l_{mean} = \frac{\sum_{n=1}^N \sqrt{(u^{ELM}(x_n, y_n)^2 - u(x_n, y_n)^2)}}{N} \quad (2.5b)$$

其中, (x_n, y_n) 表示任意子域中的测试点, u 表示某一个变量的真解, u^{ELM} 表示相应变量的带有区域分解的 ELM 方法的数值解.

在数值算例中, 空间求解域为 $[0, 1] \times [0, 1]$, 每个子域剖分的测试点数量为 3000 个. 同时, 我们也会给出多组渗透率 \mathbb{K} 来验证带有区域分解的 ELM 方法的可行性. 也会给多组剖分数 N_x, N_y 、多组子域内部随机点数 $Q * Q$ 来验证改变外部参数对求解结果的影响. 图 2.2a 展示的是剖分子域 $N_x \times N_y = 2 \times 2$, 图 2.2b 展示的是剖分子域 $N_x \times N_y = 3 \times 3$.

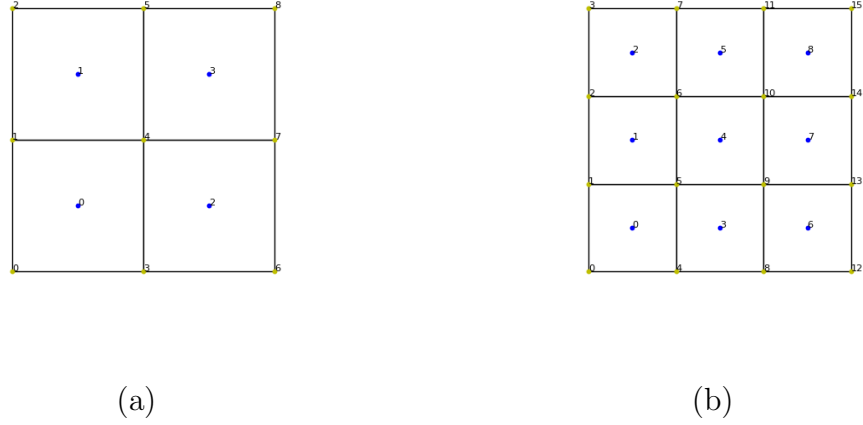


图 2.2: 算例用到的两种剖分情况

§2.4.1 数值算例 1

现给出带有区域分解的 ELM 方法求解混合形式 Darcy 方程的第一个数值算例, 考虑如下形式的 Darcy 方程:

$$\mathbf{u} = -\mathbb{K}\nabla p \quad (2.6a)$$

$$\nabla \cdot \mathbf{u} = f \quad (2.6b)$$

$$p(0, y) = g_l \quad (2.6c)$$

$$p(1, y) = g_r \quad (2.6d)$$

$$p(x, 0) = g_b \quad (2.6e)$$

$$p(x, 1) = g_t \quad (2.6f)$$

其中, g_* ($*$ = l, r, b, t) 为混合形式 Darcy 方程的 Dirichlet 边值条件. 由以上方程和边值条件, 我们给出混合形式 Darcy 方程的真解

$$p = x(1-x)y(1-y)e^{xy}$$

$$u_1 = -xy^2(1-x)(1-y)e^{xy} + xy(1-y)e^{xy} - y(1-x)(1-y)e^{xy}$$

$$u_2 = -x^2y(1-x)(1-y)e^{xy} + xy(1-x)e^{xy} - x(1-x)(1-y)e^{xy}$$

取渗透率 $\mathbb{K} = 1\mathbb{I}$ (\mathbb{I} 为单位矩阵), 每个内部边界、Dirichlet 边界训练点取 80 个, 并且每个网络只有一个隐藏层, 流体压力网络隐藏层的权重和偏置控制在 $[-R_m, R_m]$ ($R_M = 0.6$), 流体速度网络隐藏层的权重和偏置控制在 $[-R_m, R_m]$ ($R_M = 0.3$), 隐藏层长度 i 、子域剖分情况、子域内部训练点数量 $Q * Q$ 详见表 ?? 和 表 ??.

§2.4.2 数值算例 2

考虑如下形式的 Darcy 方程:

$$\mathbf{u} = -\mathbb{K}\nabla p \quad (2.8a)$$

$$\nabla \cdot \mathbf{u} = f \quad (2.8b)$$

$$p(0, y) = g_l \quad (2.8c)$$

$$p(1, y) = g_r \quad (2.8d)$$

$$p(x, 0) = g_b \quad (2.8e)$$

$$p(x, 1) = g_t \quad (2.8f)$$

其中, g_* ($*$ = l, r, b, t) 为混合形式 Darcy 方程的 Dirichlet 边值条件. 由以上方程和边值条件, 给出混合形式 Darcy 方程的真解

$$\begin{aligned} p &= -\left(\frac{3}{2}\cos\left(\pi x + \frac{2\pi}{5}\right) + 2\cos\left(2\pi x - \frac{\pi}{5}\right)\left(\frac{2\pi}{5}\cos\left(\pi y + \frac{2\pi}{5}\right) + 2\cos\left(2\pi y - \frac{\pi}{5}\right)\right)\right) \\ u_1 &= -0.5\left(1.5\pi\sin\left(\pi x + \frac{2\pi}{5}\right) + 4\pi\sin\left(2\pi x - \frac{\pi}{5}\right)\right)\left(1.5\cos\left(\pi y + \frac{2\pi}{5}\right) + 2\cos\left(2\pi y - \frac{\pi}{5}\right)\right) \\ u_2 &= -0.5\left(-1.5\pi\sin\left(\pi y + \frac{2\pi}{5}\right) - 4\pi\sin\left(2\pi y - \frac{\pi}{5}\right)\right)\left(-1.5\cos\left(\pi x + \frac{2\pi}{5}\right) - 2\cos\left(2\pi x - \frac{\pi}{5}\right)\right) \end{aligned}$$

通过数值实验我们发现, 随着隐藏层的增加, 网络误差的精度会有明显的下降, 带有区域分解的 ELM 方法的最大误差已经大于传统方法的平均误差, 带

表 2.1: 三隐藏层混合 Darcy 方程的速度数值结果

| 子域数量 | 每个子域训练点数量 | 隐藏层长度 | 最大误差 | 平均误差 | 计算时间(s) |
|------|-----------|-------------|------------|------------|---------|
| 2×2 | 10×10 | 100×150×100 | 2.1329e-04 | 3.8669e-05 | 1.70745 |
| | | 100×150×200 | 2.1320e-04 | 4.4126e-05 | 5.72108 |
| | | 100×150×300 | 7.9095e-05 | 1.6418e-05 | 17.9167 |
| 2×2 | 11×11 | 100×150×100 | 3.1624e-05 | 5.7995e-06 | 1.95038 |
| | | 100×150×200 | 5.3934e-05 | 9.1622e-06 | 7.30199 |
| | | 100×150×300 | 2.3766e-05 | 3.3172e-06 | 20.4432 |
| 2×2 | 12×12 | 100×150×100 | 2.4415e-05 | 3.9702e-06 | 1.87603 |
| | | 100×150×200 | 1.6712e-05 | 3.3533e-06 | 6.88898 |
| | | 100×150×300 | 3.0456e-06 | 6.6734e-07 | 26.0535 |

有区域分解的 ELM 方法的平均误差仅能稳定在 10^{-6} 数量级上; 计算时间有着明显的增加.

图 2.3a、2.3b、2.3c 是在隐藏层长度为 $[100 \times 300]$ 时混合 Darcy 方程的速度真解、速度数值解和速度误差图像; 图 2.3d、2.3e、2.3f 是在有两个隐藏层下的混合 Darcy 方程的压力真解、压力数值解和压力误差图像. 其训练点的数量为 12×12 . 通过图像发现, 带有区域分解的 ELM 方法对于混合形式 Darcy 方程有着较好的拟合效果.

§2.5 本章总结

本章给出了带有区域分解的 ELM 方法求解混合形式 Darcy 方程的数值格式, 并通过两个数值算例验证了带有区域分解的 ELM 方法的可行性. 并通过改变子域剖分数量、子域训练点数量、隐藏层长度、隐藏层宽度, 总结网络误差精度变化情况.

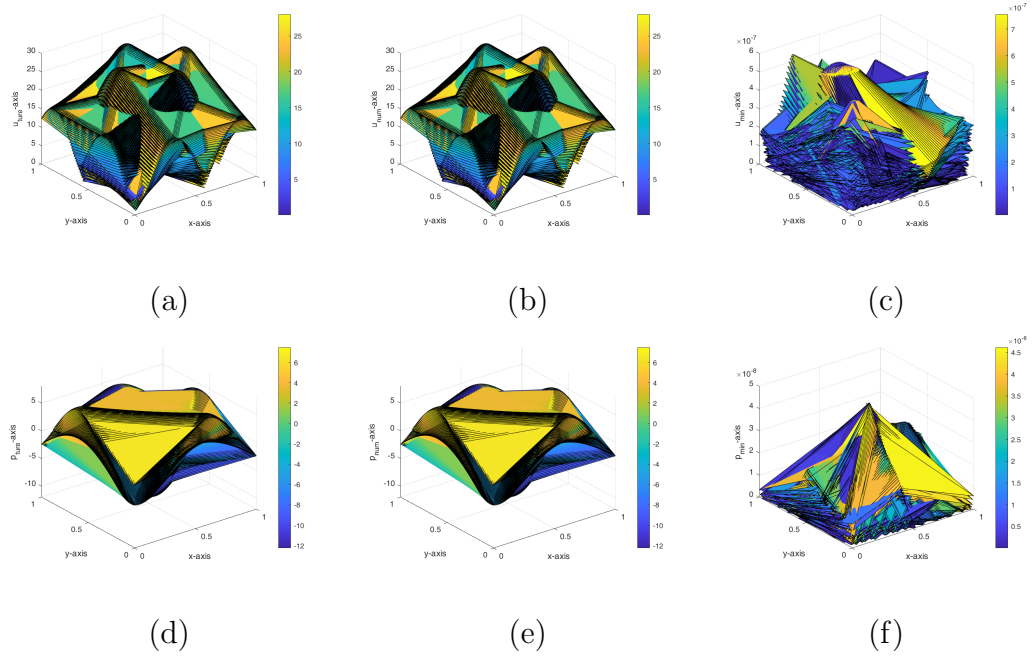


图 2.3: 隐藏层长度为 $[100 \times 300]$ 混合Darcy方程速度真解、速度数值解、速度误差; 压力真解、压力数值解、压力误差图

此外, 通过数值实验我们发现, 混合形式 Darcy 方程压力误差可以稳定在 10^{-8} 至 10^{-11} 数量级上.

第三章 极限学习求解非定常的混合多孔介质流方程

本章主要通过极限学习 (ELM) 求解非定常的混合多孔介质流 (Darcy) 方程, 与第 2 章多孔介质流方程不同的是, 本章需要考虑时间维度. 在网络设定上, 与第二章相似, 首先我们对时间求解域 $[0, T]$ 和二维空间求解域 Ω 进行区域分解, 随后在每个区域剖分上建立一个“浅”层网络. 同时, 输出层只赋予权重, 且输出层的权重是我们的训练参数. 随后得到 Darcy 方程的系数矩阵. 通过系数矩阵就能得到系数矩阵的最小二乘解, 这就是输出层权重, 这样就能得到 Darcy 方程的数值格式, 最后通过数值算例验证方法的精确性.

§3.1 模型介绍

对于非定常的混合多孔介质流方程空间求解域 Ω 为开的且是有界的, 并且 Ω 是 Lipschitz 连续的, 空间求解域为 $[0, T]$. 由此, 本章考虑非定常的 Darcy 方程:

$$\partial_t p + \nabla \cdot \mathbf{u} = \mathbf{f}, \quad \text{in } \Omega \times [0, T], \quad (3.1a)$$

$$\mathbf{u} = -\mathbb{K} \nabla p, \quad \text{in } \Omega \times [0, T], \quad (3.1b)$$

$$p = g, \quad \text{on } \partial\Omega \times [0, T], \quad (3.1c)$$

$$p(\mathbf{x}, 0) = h(\mathbf{x}), \quad \text{in } \Omega, \quad (3.1d)$$

其中 p 为流体压力, \mathbf{u} 为流体速度; g 为给出的 Dirichlet 边界条件; h 为方程的初值条件; $\partial\Omega$ 为求解域边界; \mathbb{K} 为渗透率矩阵, 这是一个常数矩阵; ∇ 为梯度算子; $\nabla \cdot$ 为散度算子.

§3.2 带有区域分解的极限学习的数值格式

为了能处理较长时间的问题, 我们在时间方向上采取了 block time marching (BTM) 格式, 即将时间区间 $[0, T]$ 分为 N_T 份 $([t_j, t_{j+1}])$ 其

中 $j = 0, 1, 2, \dots, N_T - 1$). 在任意一时间块 $[t_j, t_{j+1}]$ 中, 变量的初值条件由上一个时间块中该变量的末值给出. 为简化求解过程, 类似于本文 §2.3 节, 对空间求解域进行区域分解. 将空间求解域 $\Omega = [x_1, x_2] \times [y_1, y_2]$ 划分为 N 个不重叠的矩形子域. 令 N_x 、 N_y 为求解域 x 、 y 方向上的子域数量, 把子域的横纵坐标存成两个向量 $[x_0, x_1, \dots, x_N]$ 、 $[y_0, y_1, \dots, y_N]$, 随后, 用 $\Omega_{mnk} = [x_m, x_{m+1}] \times [y_n, y_{n+1}] \times [t_k, t_{k+1}]$ 来表示一个求解子域, 因此, 共有 $N_x \times N_y \times N_T$ 个子域.

现在使用 §2.2 节中的数值格式来对每个求解域的非定常 Darcy 方程进行求解, 每个求解子域上建立两个“浅”的前馈神经网络网络, 每个神经网络都有三个输入节点, 分别输入 (x, y, t) , 其中一个网络有两个输出节点, 分别输出流体速度的两个分量; 另一个网络有一个输出节点, 输出流体的压力. 同时, 输出层只赋予权重不赋予偏置. 并且, 所有隐藏层的权重和偏置都被随机生成在 $[-R_m, R_m]$ ($R_m > 0$). 然后, 在输入层后加一个仿射变换层, 将求解子域 Ω_{mnk} 中的输入数据归一化到 $[-1, 1] \times [-1, 1] \times [-1, 1]$, 由此, 在求解子域 Ω_{mnk} , 可以得到类似于公式 2.3 的输出:

$$u^{\Omega_{m,n,k}}(x, y, t) = \sum_{h=1}^i V_h^{u, \Omega_{m,n,k}}(x, y, t) (\omega_m^{u_1, \Omega_{m,n,k}}, \omega_m^{u_2, \Omega_{m,n,k}}) \quad (3.2a)$$

$$p^{\Omega_{m,n,k}}(x, y, t) = \sum_{h=1}^i V_h^{p, \Omega_{m,n,k}}(x, y, t) \omega_m^{p, \Omega_{m,n,k}}, \quad (x, y, t) \in \Omega_{m,n,k} \quad (3.2b)$$

其中 $\omega_h^{p, \Omega_{m,n,k}}$ 、 $\omega_h^{u_1, \Omega_{m,n,k}}$ 、 $\omega_h^{u_2, \Omega_{m,n,k}}$ 为求解子域 $\Omega_{m,n,k}$ 的训练参数, 在公式 (3.2) 中 $V_h^{u, \Omega_{m,n,k}}(x, y, t)$ 和 $V_h^{p, \Omega_{m,n,k}}(x, y, t)$ 是已知的. 我们对求解子域 $\Omega_{m,n,k}$, 将 $[x_m, x_{m+1}]$ 、 $[y_n, y_{n+1}]$ 、 $[t_k, t_{k+1}]$ 划分为 Q 段, 令 $(x_{p,eq}^{\Omega_{m,n,k}}, y_{q,eq}^{\Omega_{m,n,k}}, t_{r,eq}^{\Omega_{m,n,k}})$ 为 $\Omega_{m,n,k}$ 上的一组训练点, 并且 $0 < p, q, r < Q$, 训练点总数为 $Q \times Q \times Q$.

假设求解子域 Ω_a 与 Ω_b 有公共 (内部) 面. 在内部面上, 需要考虑源项 \mathbf{f}_a 和 \mathbf{f}_b 的 C^k 连续性, 即网络训练的数值和 PDE 沿该方向的 K 阶方向导数的数值要保证数值上是相等的. 此外, 对于求解域 Ω 的 Dirichlet 边

界, 流体压力网络的输出值需要满足模型的 Dirichlet 条件. 子域内部边界和 Dirichlet 边界训练点的选取以及网络的建立与在子域 Ω_{mnk} 上类似.

通过以上数值格式, 在任意求解子域 Ω_{mnk} 中, 取训练点 $(x_{p,eq}^{\Omega_{m,n,k}}, y_{q,eq}^{\Omega_{m,n,k}}, t_{r,eq}^{\Omega_{m,n,k}})$ 可以求解非定常 Darcy 方程 3.1, 令流体速度的系数矩阵为:

通过公式 ?? 至 ??, 可以组成一个用来计算训练参数的线性方程组. 假设通过区域分解, 共有 N_{in} 个剖分内部边界, 有 N_{bd} 个剖分外部面. 因此这个方程组共有 $N_x * N_y * N_t * Q * Q * Q * 3 + N_{in} * Q * Q * Q * 3 + N_{bd} * Q * Q * Q * 3$ 个方程, 有 $i * 3 * N$ 个需要计算的训练参数. 通过 SciPy 库, 我们计算出参数方程的最小二乘解, 按顺序反向更新到建立的网络中.

§3.3 数值实验

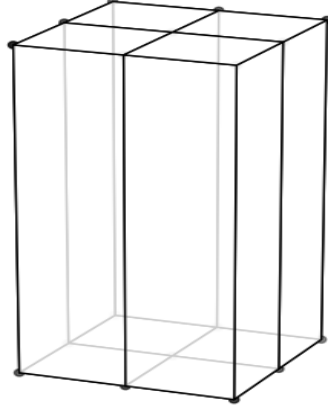
本节通过几个数值算例对带有区域分解的 ELM 方法进行验证. 此外, 数值实验代码基于 PyTorch 库, 在 Intel Core(TM) i9-13900k 上运行, 运行内存为 32G. 类似于公式 2.5, 我们给出三维形式下的最大误差公式和均方误差公式:

$$l_{max} = \max(|u^{ELM}(x_n, y_n, t_n) - u(x_n, y_n, t_n)|) \quad (3.3a)$$

$$l_{mean} = \frac{\sum_{n=1}^N \sqrt{|u^{ELM}(x_n, y_n, t_n)^2 - u(x_n, y_n, t_n)^2|}}{N} \quad (3.3b)$$

其中, (x_n, y_n, t_n) 表示任意子域中的测试点, u 表示为某一个变量的真解, u^{ELM} 表示相应变量的带有区域分解的 ELM 方法的数值解.

在数值算例中, 我们考虑处理长时间模型, 定义时空求解域 (x, y, t) 为 $[0, 0.5] \times [0, 0.5] \times [0, 5]$, 每个时间块的步长为 0.25, 每个子域的测试点数量为 3000 个. 图 3.1 展示的是单个时间块内的剖分情况.

图 3.1: $N_x \times N_y \times N_t = 2 \times 2 \times 1$

§3.3.1 数值算例 1

传统的 LDG 方法 [35] 将非定常 darcy 方程的误差控制在 10^{-5} 数量级. 通过表 ?? 和 ??, 我们发现, 带有区域分解的 ELM 方法可以将误差控制在 10^{-9} 数量级上; 并且, 只有一个隐藏层的网络效果最好, 随着隐藏层宽度的减小, 网络误差精度有较为明显的提升. 与传统方法不同的是, 带有区域分解的 ELM 方法受 BTM 格式时间步长的影响很小, 但是空间求解域的取值对网络误差精度有着较大的影响.

§3.3.2 数值算例 2

现给出带有区域分解的 ELM 方法求解非定常混合形式 Darcy 方程第二个数值算例, 考虑如下形式的非定常混合 Darcy 方程:

$$\mathbf{u} = -\mathbb{K}\nabla p \quad (3.4a)$$

$$\partial_t p + \nabla \cdot \mathbf{u} = \mathbf{f} \quad (3.4b)$$

$$p(0, y, t) = g_l \quad (3.4c)$$

$$p(1, y, t) = g_r \quad (3.4d)$$

$$p(x, 0, t) = g_b \quad (3.4e)$$

$$p(x, 1, t) = g_t \quad (3.4f)$$

$$p(x, y, 0) = h \quad (3.4g)$$

其中, $g_*(*) = l, r, b, t$ 为非定常混合 Darcy 方程的 Dirichlet 边值条件; h 为非定常混合 Darcy 方程的初值条件, 由以上初边值条件, 我们给出非定常 Darcy 方程真解:

$$p = (x(1-x)(y-1) + \frac{y^3}{3} - y^2 + y - 0.5) \cos(t)$$

$$u_1 = -(-x(y-1) + (1-x)(y-1)) \cos(t)$$

$$u_2 = -(x(1-x) + y^2 - 2y + 1) \cos(t)$$

取渗透率 $\mathbb{K} = 0.5$, 并且考虑较单一隐藏层, 流体压力网络的隐藏层的权重和偏置控制在 $[-R_m, R_m](R_m = 0.2)$, 流体速度网络网络隐藏层的权重和偏置控制在 $[-R_m, R_m](R_m = 0.23)$, 空间维度进行 2×2 剖分, 每个训练子域选取的训练点数量为 $15 * 15 * 15$ 个, 隐藏层宽度、长度 i , 时间维度及剖分情况见表 3.1 和表 3.2.

表 3.1: 压力结果

| 时间维度 | 时间步长 (S) | 隐藏层长度 | 最大误差 | 平均误差 |
|--------|----------|-------|------------|------------|
| [0, 2] | 1.0 | 300 | 4.0605e-07 | 6.2789e-08 |
| | 0.2 | 300 | 2.0755e-07 | 3.2873e-08 |
| | 0.01 | 300 | 2.0755e-07 | 3.2873e-08 |
| [0, 3] | 1.0 | 300 | 4.0374e-06 | 6.3979e-07 |
| | 0.2 | 300 | 3.4293e-06 | 6.0362e-07 |
| | 0.01 | 300 | 2.9916e-06 | 5.4087e-07 |

表 3.2: 速度结果

| 时间维度 | 时间步长 (S) | 隐藏层长度 | 最大误差 | 平均误差 |
|--------|----------|-------|------------|------------|
| [0, 2] | 1.0 | 200 | 7.8276e-07 | 3.6007e-07 |
| | 0.2 | 200 | 7.7312e-07 | 1.3529e-07 |
| | 0.01 | 200 | 7.8276e-07 | 1.4875e-07 |
| [0, 3] | 1.0 | 300 | 3.0810e-05 | 6.2481e-06 |
| | 0.2 | 300 | 4.2739e-05 | 7.2481e-06 |
| | 0.01 | 300 | 3.0810e-05 | 3.7295e-06 |

通过表 3.1 和表 3.2 表示, 对于较为复杂的数值算例, 压力网络误差精度可以控制在 10^{-8} , 速度网络误差精度可以控制在 10^{-7} .

§3.4 本章总结

本章给出了带有区域分解的 ELM 方法求解非定常的混合形式 Darcy 方程的数值格式, 并通过两个数值算例验证了带有区域分解的 ELM 方法的可行性. 并通过改变时间维度、时间块误差、隐藏层长度, 总结网络误差精度变化情况.

通过数值实验我们发现, 非定常混合形式 Darcy 方程的误差可以控制在 10^{-7} 至 10^{-9} 数量级上.

第四章 物理信息神经网络 (PINNs) 处理 Stokes-Darcy 方程

§4.1 模型介绍

首先, 令模型求解域 Ω 都是开的且有界的, 其中 Ω_S 是求解 Stokes 方程的自由流区域, Ω_D 是求解 Darcy 方程的多孔介质区域. 同时, 假设 Ω_S 和 Ω_D 都是 Lipschitz 连续的. 定义自由流和多孔介质流交界面 $\Gamma = \partial\Omega_S \cap \partial\Omega_D$, 此外, 令 Γ_S 、 Γ_D 分别对应 Stokes 方程和 Darcy 方程的所在区域的边界, 同时记 \mathbf{n}_S 为交界面处 Stokes 方程区域的单位外法向量, \mathbf{n}_D 为交界面处 Darcy 方程区域的单位外法向量.

§4.1.1 Stokes 方程

对于自由流区域 Ω_S , 以及 $(\mathbf{x}, t) \in \Omega_S \times [0, T]$, 考虑如下的 Stokes 方程

$$\partial_t \mathbf{u}_S - \nabla \cdot \mathbb{T}(\mathbf{u}_S, p_S) = \mathbf{f}_S, \quad \text{in } \Omega_S \times [0, T], \quad (4.1a)$$

$$\nabla \cdot \mathbf{u}_S = 0, \quad \text{in } \Omega_S \times [0, T], \quad (4.1b)$$

$$\mathbf{u}|_{\Gamma_S} = \mathbf{g}_S, \quad \text{on } \Gamma_S \times [0, T], \quad (4.1c)$$

$$\mathbf{u}_S(\mathbf{x}, 0) = \mathbf{h}_S(\mathbf{x}), \quad \text{in } \Omega_S, \quad (4.1d)$$

其中, $\mathbb{T}(\mathbf{u}_S, p_S) = -p_S \mathbb{I} + 2\nu \nabla_s \mathbf{u}_S$, ν 为粘度系数, ∇_s 为对称梯度算子, $\nabla \cdot$ 为散度算子, \mathbf{u}_S 为流体速度, p_S 为压力, \mathbf{g}_S 为给定的边界函数, \mathbf{f}_S 为外力项, \mathbf{h}_S 为给定的初值函数.

§4.1.2 Darcy 方程

对于多孔介质区域, 考虑如下的 Darcy 方程

$$\partial_t p_D + \nabla \cdot \mathbf{u}_D = \mathbf{f}_D, \quad \text{in } \Omega_D \times [0, T], \quad (4.2a)$$

$$\mathbf{u}_D = -\mathbb{k} \nabla p_D, \quad \text{in } \Omega_D \times [0, T], \quad (4.2b)$$

$$p_D = g_D, \quad \text{on } \Gamma_D \times [0, T], \quad (4.2c)$$

$$p_D(\mathbf{x}, 0) = h_D(\mathbf{x}), \quad \text{in } \Omega_D, \quad (4.2d)$$

其中 p_D 为流体压力, \mathbf{u}_D 为流体速度, \mathbb{k} 为渗透系数, g_D 为给定的边界函数, h_D 为给定的初值函数, \mathbf{f}_D 为包括重力加速度在内的其他物体力。

§4.1.3 交界面条件

$$\mathbf{u}_S \cdot \mathbf{n}_S = -\mathbf{u}_D \cdot \mathbf{n}_D \quad \text{on } \Gamma \times [0, T], \quad (4.3a)$$

$$\mathbf{n}_S \cdot \mathbb{T}(\mathbf{u}_S, p_S) \cdot \mathbf{n}_S = -p_D \quad \text{on } \Gamma \times [0, T], \quad (4.3b)$$

$$\boldsymbol{\tau}_j \cdot \mathbb{T}(\mathbf{u}_S, p_S) \cdot \mathbf{n}_S = \frac{\alpha}{\sqrt{\mathbb{k}}} \mathbf{u}_S \cdot \boldsymbol{\tau}_j \quad \text{on } \Gamma \times [0, T], \quad j = 1, \dots, d-1, \quad (4.3c)$$

其中 $(\boldsymbol{\tau}_j)_{j=1, \dots, d-1}$ 是 Γ 上切向空间的正交基, (4.3a) 是质量守恒条件, (4.3b) 是交界面上的动量平衡条件, (4.3c) 是 Beavers-Joseph-Saffman^[28, 29] 条件, α 为 Beavers-Joseph-Saffman 系数, 是由多孔介质流决定的。

§4.1.4 数值格式

先取 $q = -\mathbf{u}_S \cdot \mathbf{n}_S$ 定义为 Γ 上的迭代变量, 同时给定初始迭代值 $q^{(0)} = 0$ 。那么, 对于 $k = 1, 2, 3, \dots$ 直至收敛, 执行以下步骤:

- 对于 Darcy 方程部分, 有

$$\partial_t p_D^{(k)} + \nabla \cdot \mathbf{u}_D^{(k)} = f_D^{(k)}, \quad \text{in } \Omega_D \times [0, T], \quad (4.4a)$$

$$\nabla p_D^{(k)} + \mathbb{k}^{-1} \mathbf{u}_D^{(k)} = 0, \quad \text{in } \Omega_D \times [0, T], \quad (4.4b)$$

$$p_D^{(k)} = g_D, \quad \text{on } \Gamma_D \times [0, T], \quad (4.4c)$$

$$\mathbf{u}_D^{(k)} \cdot \mathbf{n}_D = q^{(k-1)}, \quad \text{on } \Gamma \times [0, T], \quad (4.4d)$$

$$p_D^{(k)} = h_D, \quad \text{in } \Omega_D \times 0. \quad (4.4e)$$

- 随后在 Stokes 方程部分, 为了简化符号, 记 $\mathbb{T}(\mathbf{u}_S, p_S)^{(k)} := \mathbb{T}(\mathbf{u}_S^{(k)}, p_S^{(k)})$, 且利用公式 (4.4) 可以得到 $p_D^{(k)}$, 将其带入 (4.3a)–(4.3c) 中对 Stokes 方程进行求解:

$$\partial_t \mathbf{u}_S^{(k)} - \nabla \cdot \mathbb{T}(\mathbf{u}_S, p_S)^{(k)} = \mathbf{f}_S^{(k)}, \quad \text{in } \Omega_S \times [0, T], \quad (4.5a)$$

$$\nabla \cdot \mathbf{u}_S^{(k)} = 0, \quad \text{in } \Omega_S \times [0, T], \quad (4.5b)$$

$$\mathbf{u}_S^{(k)} = \mathbf{g}_D, \quad \text{on } \Gamma_S \times [0, T], \quad (4.5c)$$

$$\boldsymbol{\tau}_j \cdot \mathbb{T}(\mathbf{u}_S, p_S)^{(k)} \cdot \mathbf{n}_S = \frac{\alpha}{\sqrt{\mathbb{k}}} \mathbf{u}_S^{(k)} \cdot \boldsymbol{\tau}_j, \quad \text{on } \Gamma \times [0, T], \quad j = 1, \dots, d-1, \quad (4.5d)$$

$$\mathbf{n}_S \cdot \mathbb{T}(\mathbf{u}_S, p_S)^{(k)} \cdot \mathbf{n}_S = -p_D^{(k)}, \quad \text{on } \Gamma \times [0, T], \quad (4.5e)$$

$$\mathbf{u}_S^{(k)} = \mathbf{h}_S, \quad \text{in } \Omega_S \times 0. \quad (4.5f)$$

- 接着, 需要对 $q^{(k)}$ 进行更新. 取参数 $\beta \in (0, 1)$, 由此给出 $q^{(k)}$ 的更新公式为

$$q^{(k)} = (1 - \beta)q^{(k-1)} - \beta \mathbf{u}_S^{(k)} \cdot \mathbf{n}_S. \quad (4.6)$$

§4.2 PINNs方法

与传统的有限元、有限差分法等数值方法不同的是, PINNs 是用神经网络逼近所求 PDE 解的机器学习算法, 与其他机器学习算法类似, PINNs 也存在输入空间和输出空间. 对于 PINNs, 我们将 Stokes–Darcy 系统的时间自变量 t 和空间自变量 (x, y) 进行耦合后作为输入空间. 随后我们定义 $\mathbf{u} := (\mathbf{u}_S, \mathbf{u}_D)$ 、 $p := (p_S, p_D)$ 为 Stokes–Darcy 方程的解, 同时将 \mathbf{u} 、 p 作为神经网络的输出.

本文中我们采用如上一节所述的区域分解方法求解耦合问题, 为此, 利用两个神经网络分别对 Stokes 方程和 Darcy 方程进行逼近. 为了下文叙述方便, 将在下面给出神经网络表示一般函数的数学表达式 [30]:

假设 σ 为两次可微的激活函数, z 为神经元的输入数据, $z \in \mathbb{R}^n$, 此外神经网络包含 $(L + 1)$ 层, 并且每层宽度(神经元个数)为 (l_0, l_1, \dots, l_L) , 其中 l_0 为输入层宽度, l_L 为输出层宽度, 除输入层、输出层外, 其他 $(L - 1)$ 层为隐藏层, 对于 $\theta_k := (\mathbf{W}_k, \mathbf{b}_k)$ $k \in [1, L]$ 为第 k 个神经网络的权重和偏置的组合. 除输出层外, 第 $k - 1$ 层神经元和第 k 层神经元可以通过映射 $\mathcal{A}_k^\theta: \mathbb{R}^{l_{k-1}} \rightarrow \mathbb{R}^{l_k}$ 连接: $\mathcal{A}_k^\theta: z \mapsto \mathbf{W}_k z + \mathbf{b}_k$. 因此每层神经元的输出 $f_k^\theta: \mathbb{R}^{l_{k-1}} \rightarrow \mathbb{R}^{l_k}$ 可以被表示为:

$$f_k^\theta = \begin{cases} \mathcal{A}_L^\theta(z) & k = L, \\ (\sigma \circ \mathcal{A}_k^\theta)(z) & 1 \leq k \leq L - 1, \end{cases} \quad (4.7)$$

于是, 对于任意函数 u , 经过人工神经网络逼近的结果 u_θ 可以被表示为,

$$u_\theta(z) = (f_L^\theta \circ f_{L-1}^\theta \circ \dots \circ f_1^\theta)(z). \quad (4.8)$$

这里, 我们对于 Stokes 和 Darcy 方程分别采用神经网络来逼近. 其输出记为 $\mathbf{u}_{\theta S} = (u_{\theta S}^1, u_{\theta S}^2)$, $p_{\theta S}$, $\mathbf{u}_{\theta D} = (u_{\theta D}^1, u_{\theta D}^2)$, $p_{\theta D}$, 并满足迭代公式 (4.4)–(4.6).

区域分解求解 Stokes–Darcy 耦合问题的流程, 如图 4.1 所示.

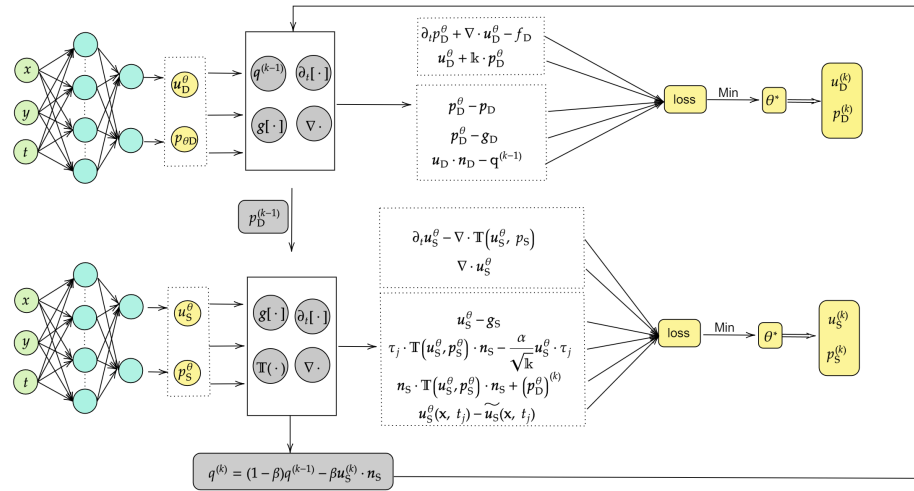


图 4.1: PINNs 方法求解 Stokes–Darcy 流程图

为了能处理较长时间的模拟, 方程在时间方向上采取了 block time marching (BTM) 格式, 即将时间区间 $[0, T]$ 分为 N_T 份 $([t_j, t_{j+1}])$ 其中 $j = 0, 1, 2, \dots, N_T - 1$. 在任意一时间块 $[t_j, t_{j+1}]$ 中, 变量的初值条件由上一个时

间块中该变量的末值给出. 下面分别考虑区域分解迭代格式 (4.4)(4.5) 中的第 k 次迭代时的残差方程.

首先给出 Darcy 区域 (4.4) 相应的残差方程:

$$R_{i1}^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t) = \partial_t p_D^\theta + \nabla \cdot \mathbf{u}_D^\theta - f_D, \quad \forall(\mathbf{x}, t) \in \Omega_D \times [t_j, t_{j+1}], \quad (4.9a)$$

$$R_{i2}^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t) = \mathbf{u}_D^\theta + \mathbb{K} \nabla p_D^\theta, \quad \forall(\mathbf{x}, t) \in \Omega_D \times [t_j, t_{j+1}], \quad (4.9b)$$

$$R_b^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t) = p_D^\theta - g_D, \quad \forall(\mathbf{x}, t) \in \Gamma_D \times [t_j, t_{j+1}], \quad (4.9c)$$

$$R_\Gamma^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t) = \mathbf{u}_D^\theta \cdot \mathbf{n}_D - q^{(k-1)}, \quad \forall(\mathbf{x}, t) \in \Gamma \times [t_j, t_{j+1}], \quad (4.9d)$$

$$R_t^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t_j) = p_D^\theta(\mathbf{x}, t_j) - \tilde{p}_D^\theta(\mathbf{x}, t_j), \quad \forall(\mathbf{x}, t) \in \Omega_D \times t_j, \quad (4.9e)$$

其中 $\tilde{p}_D^\theta(\mathbf{x}, t_j)$ 表示在上一个时间块 $[t_{j-1}, t_j]$ 中, p_D^θ 在 $t = t_j$ 处的取值.

接着, 给出 Stokes 区域 (4.5) 相应的残差方程:

$$R_{i1}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t) = \partial_t \mathbf{u}_S^\theta - \nabla \cdot \mathbb{T}(\mathbf{u}_S^\theta, p_S^\theta) - \mathbf{f}_S, \quad \forall(\mathbf{x}, t) \in \Omega_S \times [t_j, t_{j+1}], \quad (4.10a)$$

$$R_{i2}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t) = \nabla \cdot \mathbf{u}_S^\theta, \quad \forall(\mathbf{x}, t) \in \Omega_S \times [t_j, t_{j+1}], \quad (4.10b)$$

$$R_b^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t) = \mathbf{u}_S^\theta - \mathbf{g}_S, \quad \forall(\mathbf{x}, t) \in \Gamma_S \times [t_j, t_{j+1}], \quad (4.10c)$$

$$R_{\Gamma 1}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t) = \boldsymbol{\tau}_j \cdot \mathbb{T}(\mathbf{u}_S^\theta, p_S^\theta) \cdot \mathbf{n}_S - \frac{\alpha}{\sqrt{\mathbb{K}}} \mathbf{u}_S^\theta \cdot \boldsymbol{\tau}_j, \quad \forall(\mathbf{x}, t) \in \Gamma \times [t_j, t_{j+1}], \quad (4.10d)$$

$$R_{\Gamma 2}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t) = \mathbf{n}_S \cdot \mathbb{T}(\mathbf{u}_S^\theta, p_S^\theta) \cdot \mathbf{n}_S + (p_D^\theta)^{(k)}, \quad \forall(\mathbf{x}, t) \in \Gamma \times [t_j, t_{j+1}], \quad (4.10e)$$

$$R_t^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t_j) = \mathbf{u}_S^\theta(\mathbf{x}, t_j) - \tilde{\mathbf{u}}_S^\theta(\mathbf{x}, t_j), \quad \forall(\mathbf{x}, t) \in \Omega_S \times t_j, \quad (4.10f)$$

其中 $(\boldsymbol{\tau}_j)_{j=1, \dots, d-1}$ 是 Γ 上切向空间的正交基, $\tilde{\mathbf{u}}_S^\theta(\mathbf{x}, t_j)$ 表示在上一个时间块 $[t_{j-1}, t_j]$ 中, \mathbf{u}_S^θ 在 $t = t_j$ 处的取值.

由此定义迭代格式中 Darcy 方程和 Stokes 方程在时间块 $[t_j, t_{j+1}]$ 中的泛化误差:

$$\begin{aligned}
l_D = & \int_{\Omega_D \times [t_j, t_{j+1}]} (R_{i1}^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t))^2 + (R_{i2}^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t))^2 d\mathbf{x}dt \\
& + \int_{\Gamma_D \times [t_j, t_{j+1}]} (R_b^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t))^2 dsdt + \int_{\Gamma \times [t_j, t_{j+1}]} (R_\Gamma^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t))^2 dsdt \\
& + \int_{\Omega_D} (R_t^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}, t_j))^2 d\mathbf{x}, \tag{4.11a}
\end{aligned}$$

$$\begin{aligned}
l_S = & \int_{\Omega_S \times [t_j, t_{j+1}]} (R_{i1}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t))^2 + (R_{i2}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t))^2 d\mathbf{x}dt \\
& + \int_{\Gamma_S \times [t_j, t_{j+1}]} (R_b^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t))^2 dsdt + \int_{\Gamma \times [t_j, t_{j+1}]} (R_{\Gamma 1}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t))^2 dsdt \\
& + \int_{\Gamma \times [t_j, t_{j+1}]} (R_{\Gamma 2}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t))^2 dsdt + \int_{\Omega_S} (R_t^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}, t_j))^2 d\mathbf{x}. \tag{4.11b}
\end{aligned}$$

注意上面 (4.11) 给出了连续情况下的泛化误差, 而为了能够进行训练, 需要给出方程在空间和时间维度离散后的训练误差, 我们利用数值积分来代替连续意义下的积分, 所用到的积分点取为随机点 (也即后文中的配置点或训练点).

现在给出 Darcy 方程在时间块 $[t_j, t_{j+1}]$ 中的空间和时间维度下的训练点:

- $\mathcal{D}_i = \{z_n\}$, 其中 $1 \leq n \leq N_i^D$, 且 $\{z_n\} = (\mathbf{x}, t) \in \Omega_D \times [t_j, t_{j+1}]$, 为内部训练点;
- $\mathcal{D}_b = \{z_n\}$, 其中 $1 \leq n \leq N_b^D$, 且 $\{z_n\} = (\mathbf{x}, t) \in \Gamma_D \times [t_j, t_{j+1}]$, 为空间边界训练点;
- $\mathcal{D}_\Gamma = \{z_n\}$, 其中 $1 \leq n \leq N_\Gamma^D$, 且 $\{z_n\} = (\mathbf{x}, t) \in \Gamma \times [t_j, t_{j+1}]$, 为交界面训练点;
- $\mathcal{D}_t = \{\mathbf{x}_n\}$, 其中 $1 \leq n \leq N_t^D$, 且 $\{\mathbf{x}_n\} = \mathbf{x} \in \Omega_D$, 为时间边界训练点.

由此, 通过对方程 (4.11a) 中的积分进行离散化, 我们可以得到 Darcy 方程的损失函数的具体形式:

$$\mathcal{E}_D(\theta, \mathcal{D})^2 = \mathcal{E}_D^i(\theta, \mathcal{D}_i)^2 + \mathcal{E}_D^t(\theta, \mathcal{D}_t)^2 + \mathcal{E}_D^b(\theta, \mathcal{D}_b)^2 + \mathcal{E}_D^\Gamma(\theta, \mathcal{D}_\Gamma)^2, \quad (4.12)$$

其中:

$$\mathcal{E}_D^i(\theta, \mathcal{D}_i)^2 = \sum_{n=1}^{N_i^D} \omega_i^n |R_{i1}^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}_i^n, t_i^n)|^2 + \omega_i^n |R_{i2}^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}_i^n, t_i^n)|^2, \quad (4.13a)$$

$$\mathcal{E}_D^t(\theta, \mathcal{D}_t)^2 = \sum_{n=1}^{N_t^D} \omega_t^n |R_t^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}_t^n, t_t^n)|^2, \quad (4.13b)$$

$$\mathcal{E}_D^b(\theta, \mathcal{D}_b)^2 = \sum_{n=1}^{N_b^D} \omega_b^n |R_b^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}_b^n, t_b^n)|^2, \quad (4.13c)$$

$$\mathcal{E}_D^\Gamma(\theta, \mathcal{D}_\Gamma)^2 = \sum_{n=1}^{N_\Gamma^D} \omega_\Gamma^n |R_\Gamma^D[\mathbf{u}_D^\theta, p_D^\theta](\mathbf{x}_\Gamma^n, t_j)|^2, \quad (4.13d)$$

这里空间和时间维度中的积分节点也就是前面定义的训练点集合 \mathcal{D}_\star , 以及 ω_\star^n 为损失函数的权重系数 (\star 表示 i 、 t 、 b 或 Γ).

接着给出 Stokes 方程在时间块 $[t_j, t_{j+1}]$ 中的空间和时间维度下的训练点:

- $\mathcal{S}_i = \{z_n\}$, 其中 $1 \leq n \leq N_i^S$, 且 $\{z_n\} = (\mathbf{x}, t) \in \Omega_S \times [t_j, t_{j+1}]$, 为内部训练点;
- $\mathcal{S}_b = \{z_n\}$, 其中 $1 \leq n \leq N_b^S$, 且 $\{z_n\} = (\mathbf{x}, t) \in \Gamma_S \times [t_j, t_{j+1}]$, 为空间边界训练点;
- $\mathcal{S}_\Gamma = \{z_n\}$, 其中 $1 \leq n \leq N_{\Gamma 1}^S$, 且 $\{z_n\} = (\mathbf{x}, t) \in \Gamma \times [t_j, t_{j+1}]$, 为交界面训练点;
- $\mathcal{S}_t = \{z_n\}$, 其中 $1 \leq n \leq N_t^S$, 且 $\{z_n\} = \mathbf{x} \in \Omega_S$, 为时间边界训练点.

与 Darcy 方程类似, 通过对方程 (4.11b) 中的积分进行离散化, 可以得到 Stokes 方程的损失函数的具体形式:

$$\mathcal{E}_S(\theta, \mathcal{S})^2 = \mathcal{E}_S^i(\theta, \mathcal{S}_i)^2 + \mathcal{E}_S^b(\theta, \mathcal{S}_b)^2 + \mathcal{E}_S^\Gamma(\theta, \mathcal{S}_\Gamma)^2 + \mathcal{E}_S^t(\theta, \mathcal{S}_t)^2, \quad (4.14)$$

其中:

$$\mathcal{E}_S^i(\theta, \mathcal{S}_i)^2 = \sum_{n=1}^{N_i^S} \omega_{i1}^n |R_i^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}_i^n, t_i^n)|^2 + \omega_i^n |R_{i2}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}_i^n, t_i^n)|^2, \quad (4.15a)$$

$$\mathcal{E}_S^b(\theta, \mathcal{S}_b)^2 = \sum_{n=1}^{N_b^S} \omega_b^n |R_b^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}_b^n, t_b^n)|^2, \quad (4.15b)$$

$$\mathcal{E}_S^t(\theta, \mathcal{S}_t)^2 = \sum_{n=1}^{N_t^S} \omega_t^n |R_t^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}_t^n, t_t^n)|^2, \quad (4.15c)$$

$$\mathcal{E}_S^\Gamma(\theta, \mathcal{S}_\Gamma)^2 = \sum_{n=1}^{N_\Gamma^S} \omega_{\Gamma1}^n |R_{\Gamma1}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}_\Gamma^n, t_\Gamma^n)|^2 + \omega_{\Gamma2}^n |R_{\Gamma2}^S[\mathbf{u}_S^\theta, p_S^\theta](\mathbf{x}_\Gamma^n, t_\Gamma^n)|^2, \quad (4.15d)$$

这里空间和时间维度中的积分节点也就是前面定义的训练点集合 \mathcal{S}_\star , 以及 ω_\star^n 为损失函数的权重系数 (\star 表示 i 、 t 、 b 或 Γ). 在机器学习中, 常用的激活函数有: Sigmoid 函数、双曲正切函数、ReLU 函数等可微的函数 [25–27]. 由于 PINNs 模型在求解 PDE 的过程中可能出现求高阶导的情况, 所以我们尽量选择无穷次可微的激活函数. 考虑到内存使用和模型复杂度, 我们选择双曲正切函数作为网络激活函数. 同时, 为了更好的优化网络, 我们使用 LBFGS 算法进行网络优化.

§4.3 数值算例

本节通过两个数值算例对训练的网络进行验证. 此外, PINNs 方法的代码基于 PyTorch 库. 代码在 Intel(R) Xeon(R) Gold 6240 上运行. 使用 GeForce RTX 2080 对模型进行加速.

PINNs 方法的有效性可以通过计算方程数值解与精确解的相对误差来验证. 利用 PINNs 求解得到神经网络参数 θ 后, 再利用公式 (4.16) 来计算方程数值解和精确解的 l^2 相对误差:

$$l^2 = \frac{\sqrt{\sum_{n=1}^N |u(x_n, y_n, t_n) - u^\theta(x_n, y_n, t_n)|^2}}{\sqrt{\sum_{n=1}^N |u(x_n, y_n, t_n)|^2}}, \quad (4.16)$$

其中, (x_n, y_n, t_n) 表示求解区域 Ω_D (或 Ω_S) 中的一组均匀网格剖分. u 表示某个变量的真解, u_θ 表示相应的 PINNs 数值解.

在数值算例中, 时间维度的求解域为 $t \in [0, 2]$, 并均分为五部分, 每部分初始时间记为 t_j ; 空间维度的求解域为 $(x, y) \in [0, 1] \times [0, 2]$. 在训练网络时, 先用 Adma 优化算法对网络进行 100 次迭代训练, 然后使用 LBFGS 优化算法对网络进行 10000 次迭代训练. Adma 优化算法学习率为 0.001, LBFGS 优化算法学习率为 1.

本文通过 10 次 D-N 迭代对 Stokes-Darcy 方程的耦合面进行数据交换, 此外, 神经网络的全连接层使用残差网络 (ResNet) 对 Stokes-Darcy 方程进行训练, 在 Darcy 方程区域, 使用了 3 个 ResNet 块进行训练, 每个 ResNet 块有 3 个隐藏层, 每个隐藏层的长度是 30; 在 Stokes 方程区域, 使用了 2 个 ResNet 块进行训练, 每个 ResNet 块有 2 个隐藏层, 每个隐藏层的长度是 30.

Darcy 方程部分的损失函数的权重系数 $(\omega_i^n, \omega_b^n, \omega_t^n, \omega_\Gamma^n) = (0.03, 0.97, 0.03, 0.97)$; Stokes 方程部分的损失函数的权重系数 $(\omega_i^n, \omega_b^n, \omega_t^n, \omega_{\Gamma_1}^n, \omega_{\Gamma_2}^n) = (0.04, 0.96, 0.04, 0.96, 0.96)$

同时, Darcy 方程的空间维度 $\Omega_D = [0, 1] \times [0, 1]$; Stokes 方程的空间维度 $\Omega_D = [0, 1] \times [1, 2]$; 本文在空间和时间维度上随机取 1500 个训练点进行网络的训练; 随机取 200 测试点进行误差计算和画图.

§4.3.1 数值算例 1

现给出 PINNs 方法求解 Stokes-Darcy 方程的第一个数值算例, 考虑如下形式的 Stokes 方程:

$$\partial_t \mathbf{u}_S - \nabla \cdot \mathbb{T}(\mathbf{u}_S, p_S) = \mathbf{f}_S^1(x, y, t), \quad (4.17a)$$

$$\nabla \cdot \mathbf{u}_S = 0, \quad (4.17b)$$

$$\mathbf{u}_S(0, y, t) = \mathbf{g}_{S1}^1, \quad (4.17c)$$

$$\mathbf{u}_S(1, y, t) = \mathbf{g}_{S2}^1, \quad (4.17d)$$

$$\mathbf{u}_S(x, 2, t) = \mathbf{g}_{S3}^1, \quad (4.17e)$$

$$\mathbf{u}_S(x, y, t_j) = \mathbf{h}_S^1. \quad (4.17f)$$

和 Darcy 方程:

$$\partial_t p_D + \nabla \cdot \mathbf{u}_D = \mathbf{f}_D^1(x, y, t), \quad (4.18a)$$

$$\mathbf{u}_D + \mathbb{k} \nabla p_D = 0, \quad (4.18b)$$

$$p_D(0, y, t) = g_{D1}^1, \quad (4.18c)$$

$$p_D(1, y, t) = g_{D2}^1, \quad (4.18d)$$

$$p_D(x, 0, t) = g_{D3}^1, \quad (4.18e)$$

$$p_D(x, y, t_j) = h_D^1. \quad (4.18f)$$

其中, $\mathbf{g}_{S\star}^1$ 、 $g_{D\star}^1$ 为方程初值条件 ($\star = 1, 2, 3$), \mathbf{h}_S^1 、 h_D^1 为方程边值条件, \mathbf{f}_S^1 、 \mathbf{f}_D^1 为源项方程. 同时公式 (4.17) (4.18) 满足交界面条件公式 (4.3), 为方便计算, 对于公式 (4.17)、(4.18)、(4.3) 中粘滞系数 ν 、渗透系数 \mathbb{k} 、Beavers-Joseph-Saffman 系数 α 取为 1, 由以上初值条件、边值条件、源项方程、系数, 引出算例 1 真解:

$$\mathbf{u} = (-\sin(\pi y) \cos(\pi x) \cos(t), \sin(\pi x) \cos(\pi y) \cos(t)), \quad (4.19a)$$

$$p = \sin(\pi x) \cos(t), \quad (4.19b)$$

$$\phi = y \sin(\pi x) \cos(t). \quad (4.19c)$$

为了更直观展现训练效果, 时间维度上取 $t = 0.4, 0.8, 1.2, 1.6, 2.0$, 表 4.1 是算例 1 Stokes 方程 200 个训练点, 在对应时间取值、空间维度取值上的损失函数以及速度 l^2 误差、压力 l^2 误差的部分结果; 表 4.2 是算例 1 Darcy 方程 200 个训练点, 在对应时间取值、空间维度取值上的损失函数以及速度 l^2 误差、压力 l^2 误差的部分结果. 由此可见, PINNs 方法对于拟合 Stokes 方程得到的速度数值解与真解的 l^2 误差可以控制在 10^{-3} 数量级上, 对于 Darcy 方程得到的速度数值解与真解的 l^2 误差可以控制在 10^{-2} 数量级上, 对于拟

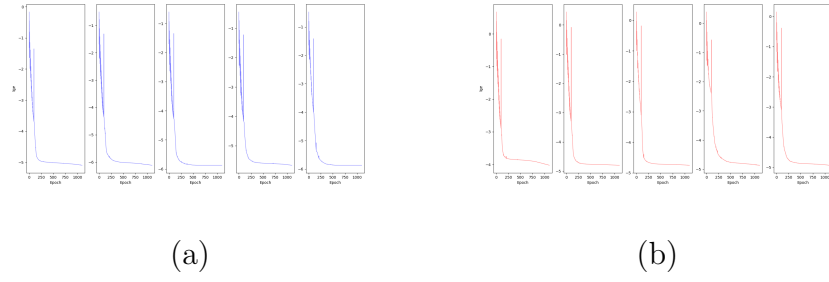


图 4.2: Stokes - Darcy 损失函数变化图像

合 Stokes-Darcy 方程得到的压力数值解与真解的 l^2 误差可以控制在 10^{-2} 数量级上.

图 4.2a 为 Darcy 方程损失函数变化图像, 图 4.2b 为 Stokes 方程损失函数变化图像. 由图可见, 当优化算法从 Adma 转化为 LBFGS 算法时, 损失函数有一个较为明显的增大点, 但随后就会迅速下降. 最终, LBFGS 算法优化的网络可以将损失函数的数值控制在 10^{-6} 数量级上.

表 4.1: Stokes 方程训练结果

| 时间维度取值 | Stokes方程损失函数 | Stokes方程速度 l^2 误差 | Stokes方程压力 l^2 误差 |
|--------|--------------|---------------------|---------------------|
| 0.4 | 1.1095e-04 | 5.6757e-03 | 2.4802e-01 |
| 0.8 | 3.9804e-05 | 4.3212e-03 | 2.0534e-01 |
| 1.2 | 1.5793e-04 | 7.4952e-03 | 3.4395e-01 |
| 1.6 | 9.9594e-05 | 5.4319e-03 | 9.0452e-02 |
| 2.0 | 7.8425e-06 | 1.4505e-03 | 4.7717e-02 |

图 4.3 给出了算例 1 在测试集上速度的精确解、PINNs 方法拟合解、逐点绝对误差的分布图. 可见 PINNs 方法对于耦合方程的速度部分有着较好的拟合效果.

图 4.4 给出了算例 1 在测试集上压力的精确解、PINNs 方法拟合解、逐点绝对误差的分布图. 可见 PINNs 方法对于耦合方程的压力部分有着较好的

表 4.2: Darcy 方程的训练结果

| 时间维度取值 | Darcy 方程损失函数 | Darcy 方程速度 l^2 误差 | Darcy 方程压力 l^2 误差 |
|--------|--------------|---------------------|---------------------|
| 0.4 | 1.2827e-03 | 1.8087e-02 | 7.1830e-01 |
| 0.8 | 4.5342e-05 | 8.2549e-02 | 4.1335e-01 |
| 1.2 | 1.7501e-03 | 2.5128e-02 | 9.8229e-02 |
| 1.6 | 5.3585e-04 | 9.5346e-02 | 3.9903e-02 |
| 2.0 | 3.0915e-05 | 1.9919e-02 | 1.0242e-02 |

拟合效果.

§4.3.2 数值算例 2

我们给出 PINNs 方法求解 Stokes-Darcy 方程的第二个数值算例, 考虑如下形式的 Stokes 方程:

$$\partial_t \mathbf{u}_S - \nabla \cdot \mathbb{T}(\mathbf{u}_S, p_S) = \mathbf{f}_S^2(x, y, t), \quad (4.20a)$$

$$\nabla \cdot \mathbf{u}_S = 0, \quad (4.20b)$$

$$\mathbf{u}_S(0, y, t) = \mathbf{g}_{S1}^2, \quad (4.20c)$$

$$\mathbf{u}_S(1, y, t) = \mathbf{g}_{S2}^2, \quad (4.20d)$$

$$\mathbf{u}_S(x, 2, t) = \mathbf{g}_{S3}^2, \quad (4.20e)$$

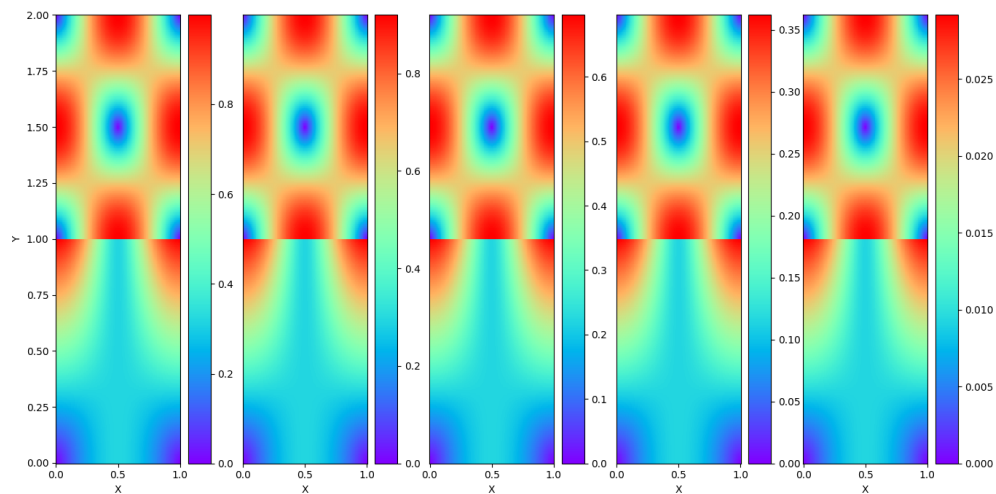
$$\mathbf{u}_S(x, y, t_j) = \mathbf{h}_S^2. \quad (4.20f)$$

和 Darcy 方程:

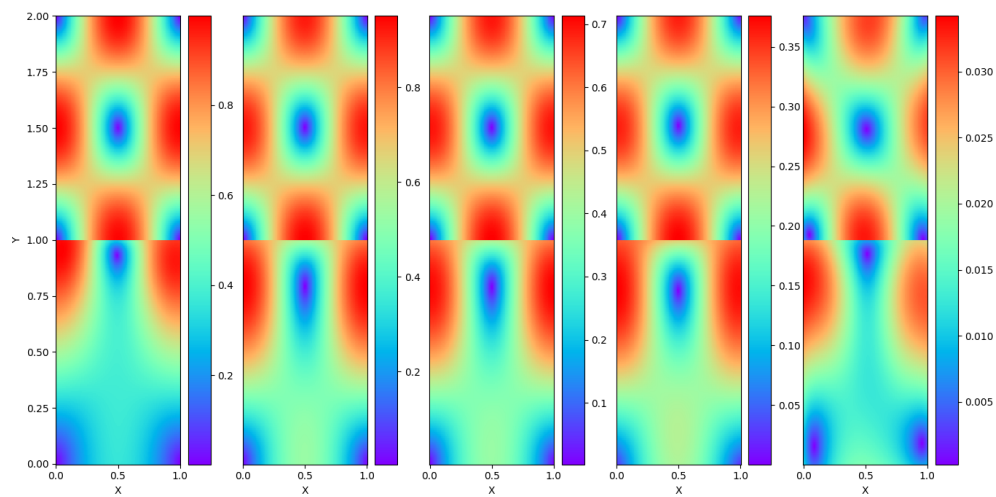
$$\partial_t p_D + \nabla \cdot \mathbf{u}_D = \mathbf{f}_D^2, \quad (4.21a)$$

$$\mathbf{u}_D + \mathbb{k} \nabla p_D = 0, \quad (4.21b)$$

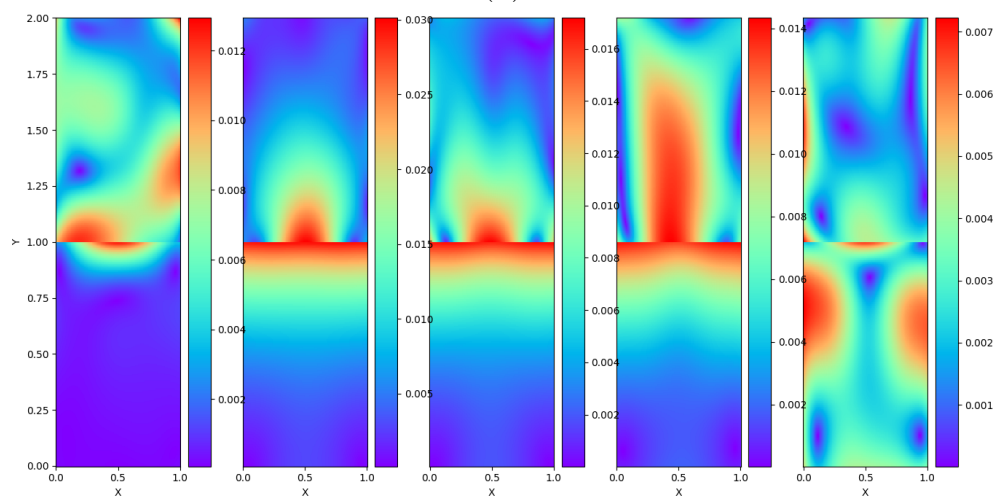
$$p_D(0, y, t) = g_{D1}^2, \quad (4.21c)$$



(a)



(b)



(c)

图 4.3: Stokes - Darcy 方程速度真解、数值解及误差图像

$$p_D(1, y, t) = g_{D2}^2, \quad (4.21d)$$

$$p_D(x, 0, t) = g_{D3}^2, \quad (4.21e)$$

$$p_D(x, y, t_j) = h_D^2. \quad (4.21f)$$

其中, $g_{S\star}^2$ 、 $g_{D\star}^2$ 为第二个数值算例的初值条件 ($\star = 1, 2, 3$), h_S^2 、 h_D^2 为算例的边值条件, f_S^2 、 f_D^2 为源项方程. 同时公式 (4.20)、(4.21) 满足交界面条件公式 (4.3), 为方便计算, 对于公式 (4.20)、(4.21)、(4.3) 中粘滞系数 ν 、渗透系数 k 、Beavers-Joseph-Saffman 系数 α 取为 1, 由以上初值条件、边值条件、源项方程、系数, 引出算例 2 真解:

$$\mathbf{u} = ((1 - 2x)(y - 1) \cos(t), (x(x - 1) + (y - 1)^2) \cos(t)), \quad (4.22a)$$

$$p = (x(1 - x)(y - 1) + \frac{y^3}{3} - y^2 + y - 0.5) \cos(t), \quad (4.22b)$$

$$\phi = (x(1 - x)(y - 1) + \frac{y^3}{3} - y^2 + y - 0.5) \cos(t). \quad (4.22c)$$

我们固定时间维度, 时间取值为 $t = 0, 0.4, 0.8, 1.2, 1.6$, 表 4.3 是算例 2 Stokes 方程 200 个训练点, 在对应时间取值、空间维度取值上的损失函数以及速度 l^2 误差、压力 l^2 误差的部分结果; 表 4.4 是算例 2 Darcy 方程 200 个训练点, 在对应时间取值、空间维度取值上的损失函数以及速度 l^2 误差、压力 l^2 误差的部分结果. 由此可见, Stokes-Darcy 方程的速度 l^2 误差可以控制在 10^{-3} 数量级上, 压力 l^2 误差可以控制在 10^{-2} 数量级上. 因此, PINNs 方法对于求解 Stokes-Darcy 方程的泛化能力较强.

表 4.3: Stokes 方程的训练结果

| 时间维度取值 | Stokes方程损失函数 | Stokes方程速度 l^2 误差 | Stokes方程压力 l^2 误差 |
|--------|--------------|---------------------|---------------------|
| 0.4 | 4.3561e-05 | 1.2168e-03 | 2.2854e-02 |
| 0.8 | 5.9524e-06 | 1.7645e-03 | 2.3557e-02 |
| 1.2 | 5.2224e-05 | 1.6245e-03 | 2.5052e-02 |
| 1.6 | 6.0023e-06 | 1.8228e-03 | 2.6052e-02 |
| 2.0 | 4.7275e-06 | 1.3525e-03 | 2.4842-e02 |

表 4.4: Darcy 方程的训练结果

| 时间维度取值 | Darcy方程损失函数 | Darcy方程速度 l^2 误差 | Darcy方程压力 l^2 误差 |
|--------|-------------|--------------------|--------------------|
| 0 | 3.5613e-06 | 5.4625e-03 | 2.7861e-02 |
| 0.4 | 3.1232e-05 | 7.0354e-03 | 3.0961e-02 |
| 0.8 | 2.9832e-06 | 6.4351e-03 | 3.1856e-02 |
| 1.2 | 1.5993e-05 | 7.7196e-03 | 3.2966e-02 |
| 1.6 | 3.7891e-06 | 5.6047e-03 | 2.9399e-02 |

参考文献

- [1] Melenk, J. M. Quasi-optimal *a priori* estimates for fluxes in mixed finite element methods and an application to the Stokes-Darcy coupling [J]. IMA J. Numer. Anal., 2014, 34, 1-27.
- [2] Wen Jing , Su Jian, He Yinnian , Chen Hongbin. A discontinuous Galerkin method for the coupled stokes and Darcy problem [J]. J. Sci. Comput., 2020, 85, 26-27.
- [3] Kumar Prashant , Luo Peiyao , Gaspar, Francisco J. , Oosterlee, Cornelis W. A multigrid multilevel Monte Carlo method for transport in the Darcy-Stokes system [J]. J. Comput. Phys., 2018, 371, 382–408.
- [4] Elbr?chter Dennis, Perekrestenko Dmytro, Grohs Philipp, B?lcskei Helmut. Deep Neural Network Approximation Theory [J]. IEEE Transactions on Information Theory, 2021, 67, 2581-2623.
- [5] Pang Guofei, Em Karniadakis George. Physics-informed learning machines for partial differential equations: Gaussian processes versus neural networks [J]. Nonlinear Syst. Complex., 2020, 32, 323–343.
- [6] De Florio Mario, Schiassi Enrico, Calabrò, Francesco, Furfaro Roberto. Physics-informed neural networks for 2nd order ODEs with sharp gradients [J]. J. Comput. Appl. Math., 2024, 436, Paper No. 115396, 8.
- [7] Ervin, V. J , Jenkins, E. W, Lee, Hyesuk, Cornelis W. Approximation of the Stokes-Darcy system by optimization [J]. J. Sci. Comput., 2014, 59, 3.
- [8] Kashinath K. , Mustafa M. , Albert A. , et al. Physics-informed machine learning: case studies for weather and climate modelling [J]. Philos. Trans. Roy. Soc. A, 2021, 379, Paper No. 20200093, 36.

-
- [9] Tlupova Svetlana . Cortez Ricardo. Boundary integral solutions of coupled Stokes and Darcy flows [J].J. Comput. Phys., 2009, 228, 158–179.
 - [10] Boubendir Yassine , Stokes-Darcy boundary integral solutions using preconditioners [J].J. Comput. Phys., 2009, 228, 8627–8641.
 - [11] Zhang Yongchao , Mei Liquan , Li, Rui , A hybrid high-order method for a coupled Stokes-Darcy problem on general meshes [J].J. Comput. Phys., 2020, 403, 109064, 23.
 - [12] Zhao Jikun , Zhang Bei, Mao, Shipeng ,Chen Shaochun, The nonconforming virtual element method for the Darcy-Stokes problem [J].Comput. Methods Appl. Mech. Engrg., 2020, 370, 113251, 23.
 - [13] Liu Xin, Li Rui, Chen Zhangxin, A virtual element method for the coupled Stokes-Darcy problem with the Beaver-Joseph-Saffman interface condition [J].Calcolo, 2019, 56, Paper No. 48, 28.
 - [14] Peng Hui ,Zhai Qilong ,Zhang Ran , Zhang Shangyou. A weak Galerkin-mixed finite element method for the Stokes-Darcy problem [J].Sci. China Math., 2021, 64, 2357–2380.
 - [15] Li Rui ,Gao Yali, Li Jian, Chen Zhangxin.A weak Galerkin finite element method for a coupled Stokes-Darcy problem on general meshes [J].J. Comput. Appl. Math., 2018, 334, 111–127.
 - [16] Botti Lorenzo, Di Pietro Daniele A. , Droniou.A Hybrid High-Order discretisation of the Brinkman problem robust in the Darcy and Stokes limits[J].Comput. Methods Appl. Mech. Engrg., 2018, 341, 278–310.
 - [17] Eymard R, Gallouët T, Herbin R.A. Discretization of heterogeneous and anisotropic diffusion problems on general nonconforming meshes SUSHI: a

-
- scheme using stabilization and hybrid interfaces [J].IMA J. Numer. Anal., 2010, 30, 1009–1043.
- [18] Tlupova, Svetlana. A domain decomposition solution of the Stokes-Darcy system in 3D based on boundary integrals [J].J. Comput. Phys., 2022, 450, Paper No. 110824, 18.
- [19] Egger Herbert , Waluga Christian. A hybrid discontinuous Galerkin method for Darcy-Stokes problems [J].Lect. Notes Comput. Sci. Eng., 2013, 91, 663–670.
- [20] Droniou Jérôme , Eymard Robert. Artificial neural networks for solving ordinary and partial differential equations [J].J. Comput. Phys., 2006, 105, 35–71.
- [21] Lagaris I.E. , Likas A. , Fotiadis D.I . Artificial neural networks for solving ordinary and partial differential equations [J].J. Comput. Phys., 1998, 9, 987-1000.
- [22] 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].J. Comput. Phys., 2019, 378, 686–707.
- [23] Sirignano Justin , Spiliopoulos Konstantinos. DGM: a deep learning algorithm for solving partial differential equations [J].J. Comput. Phys., 2018, 375, 1339–1364.
- [24] Shan Li , Zheng Haibiao, Layton William J.A decoupling method with different subdomain time steps for the nonstationary Stokes-Darcy model [J].Numer. Methods Partial Differential Equations, 2013, 29, 549–583.

-
- [25] 查文舒,李道伦, 沈路航, 张雯, 刘旭亮. 基于神经网络的偏微分方程求解方法研究综述 [J].力学学报, 2022, 54, 14.
- [26] 郑素佩,靳放, 封建湖, 林云云. 双曲型方程激波捕捉的物理信息神经网络(PINN)算法 [J].浙江大学学报:理学版, 2023, 50, 56-62.
- [27] 兰鹏, 李海潮, 叶新宇, 张升, 盛岱超. PINNs 算法及其在岩土工程中的应用研究 [J].岩土工程学报, 2021, 43, 9.
- [28] Saffman , P. G. On the Boundary Condition at the Surface of a Porous Medium [J].Stud. Appl. Math., 1971, 50, 1467-9590.
- [29] Beavers Gordon S., Joseph Daniel D. Boundary conditions at a naturally permeable wall [J].J. Fluid Mech., 1971, 50, 1467-9590.
- [30] De Ryck, Tim and Jagtap, Ameya D and Mishra, Siddhartha. Error estimates for physics-informed neural networks approximating the Navier-Stokes equations [J].IMA J. Numer. Anal., 2023, drac085.
- [31] Aycil Cesmelioglu, Sander Rhebergen. A hybridizable discontinuous Galerkin method for the coupled Navier - Stokes and Darcy problem [J].J. C. Applied Math., 2023, 114923.
- [32] Bernardi Christine, Dib Séréna , Girault Vivette, Hecht Frédéric and Murat, François, Sayah, Toni. Finite element methods for Darcy’s problem coupled with the heat equation [J].Numer. Math., 2018, 139, 315–348.
- [33] Suchuan Dong, Zongwei Li. Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations [J].Computer Methods in Applied Mechanics and Engineering,, 2021, 387, 114129.

-
- [34] Rupp Andreas , Knabner Peter, Dawson Clint. A local discontinuous Galerkin scheme for Darcy flow with internal jumps [J].Comput. Geosci., 2018, 22, 1149–1159.
- [35] Vadym Aizinger, Clint Dawson. The local discontinuous Galerkin method for three-dimensional shallow water flow [J].Comput. Geosci., 2007, 22, 734–746.
- [36] X.B. Feng. On Existence and Uniqueness Results for a Coupled System Modeling Miscible Displacement in Porous Media [J].J. A. Ana. and App., 1995, 194, 0022-247X.
- [37] Chen Zhangxin , Ewing Richard. Mathematical Analysis for Reservoir Models [J].SIAM J. M. Ana., 1999, 0, 431-453.
- [38] Hughes, Thomas J. R., Masud Arif , Wan Jing . A stabilized mixed discontinuous Galerkin method for Darcy flow [J].Comput. Methods Appl. Mech. Engrg. 2006, 195, 3347–3381.
- [39] Michael A. Anderson , Jack R. C. Parker. Sensitivity of organic contaminant transport and persistence models to Henry’s Law constants: Case of polychlorinated biphenyls [J].Water, Air, and Soil Pollution. 1990, 50, 1–18.
- [40] Dawson, Clint. The $\mathcal{P}^{\parallel} + \infty - \mathcal{S}^{\parallel}$ Local Discontinuous Galerkin Method for Elliptic Equations [J].SIAM J. Num. Ana.. 2002, 40, 2151-2170.
- [41] Cockburn Bernardo, Shu Chi-Wang. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems [J].SIAM J. Num. Ana.. 1998, 35, 2440-2463.
- [42] Cockburn Bernardo, Shu Chi-Wang. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems [J].SIAM J. Num. Ana.. 1998, 35, 2440-2463.

- [43] Antonietti P. F., Dassi F., Manzuzzi E. Shu. Machine learning based refinement strategies for polyhedral grids with applications to virtual element and polyhedral discontinuous Galerkin methods [J].J. Comput. Phys.. 2022, 469, Paper No. 111531, 22.
- [44] Yu Xinyue, Shu Chi-Wang. Multi-layer perceptron estimator for the total variation bounded constant in limiters for discontinuous Galerkin methods [J].Matematica. 2022, 1, 53–84.
- [45] Keith Rudd , Silvia Ferrari. A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks [J].Neurocomputing. 2015, 22, 277-285.
- [46] Guangbin Huang , Dianhui Wang Yuan Lan. Extreme learning machines: a survey [J].I. J. Machine Learning and Cybernetics. 2011, 2, 107-122.

攻读硕士学位期间取得的科研成果

- [1] 徐石玮, 张永超, 程变茹, 物理信息神经网络 (PINNs) 处理Stokes-Darcy方程[J]. 纯粹数学与应用数学.(已录用)
- [2] Li Fang, Yu Wang, Shiwei Xu, Zhenhua Guo, Global existence and time behaviour of strong solution to a compressible non-Newtonian fluid [J]. Non-linear Analysis: Real World Applications, 2023, 72, 1468-1218.

致谢