

浙江大学

本科生毕业论文

文献综述和开题报告



姓名与学号 陈冠宇 3200102033

指导教师 倪东，徐翔

年级与专业 2020级数学与应用数学 (强基计划)

所在学院 数学科学学院

一、题目：基于对流扩散方程关于物理信息机器学习 (PIML) 算法的研究

二、指导教师对文献综述、开题报告、外文翻译的具体要求：

文献综述要求：根据阅读的国内外文献撰写文献综述报告，要求根据主体展开，文献综述内容要切题，包括：

(1) 简述经典数值算法的方法，以及其相关优势和缺陷。

(2) 简述深度学习算法在偏微分方程领域的重要进展和重要方法。掌握当前该领域的研究前沿，分析毕业论文研究内容与这些论文的差异和相关。

(3) 简述深度学习算法的创新性，存在的问题以及未能解决的问题，并提出研究展望。

(4) 要求翻译其中的一片外文文献，结构完整，语句通顺。

开题报告要求：

(1) 分析问题研究背景 (包括理论意义以及实际意义)。

(2) 可行性分析

(3) 主要研究内容和研究方法。

(4) 根据研究内容写出具体的实施计划。

(5) 明确论文最后的预期结果。

指导教师（签名）_____

年 月 日

目录

一、 文献综述 1

1 背景介绍 1

1.1 前言 1

1.2 对流扩散方程 2

1.3 经典数值方法 2

1.4 深度学习方法 3

2 国内外研究现状 3

2.1 研究方向及进展 3

2.2 存在问题 5

2.3 研究展望 6

3 参考文献 8

二、 开题报告 11

1 问题提出的背景 11

1.1 背景介绍 11

1.2 研究的意义和目的 11

2 可行性分析 12

3 主要内容 13

3.1 图神经网络 14

3.2 间断 Galerkin 方法^[17] 14

3.3 DG-GCN 架构 15

4 研究计划进度安排及预期目标 15

4.1 进度安排 15

4.2 预期目标 16

5 参考文献 17

三、 外文翻译 19

1 原文出处: 19

2	摘要	20
3	介绍	20
4	方法	23
4.1	概述	23
4.2	对非结构化数据的图神经网络	23
4.3	变分形式的物理信息损失项	25
4.4	统一正反问题	27
5	数值实验	27
5.1	Possion equation	29
5.2	Navier-Stokes 方程	30
6	讨论	32
6.1	GCN 与 CNN 的对比	32
6.2	展望	34
7	总结	34
四、	外文原文	35
	毕业论文（设计）文献综述和开题报告考核	41

一、文献综述

1 背景介绍

1.1 前言

偏微分方程 (Partial Differential Equation) 非常普遍, 在科学、工程、经济和金融领域有着广泛的应用。同时在很多实际工程或产业问题中, 问题常常被形式化为偏微分方程的形式。广泛研究的 PDE 包括物理学中的泊松方程 (Poisson's equation)、纳维-斯托克斯方程 (Navier-Stokes equation)、麦克斯韦方程 (Maxwell's equation)、薛定谔方程 (Schrödinger equation) 及工程应用领域的哈密顿-雅可比-贝尔曼方程 (Hamilton-Jacobi Bellman equation) 等。

因此, 微分方程描述的物理现象广泛存在于自然科学、民生、经济和工业等领域中, 其求解方法研究具有重大的理论和实际工程价值。然而如何精准的模拟并预测这些场的变化, 如流场, 电磁场甚至是多物理场, 是现在比较棘手的问题。因为对于偏微分方程而言, 仅有少部分方程能够由分析的方法得到其精确的解析结果, 但是对于绝大多数的偏微分方程而言, 由于其一些高度非线性项, 其解析解几乎不能显式表达出来。为此我们发展了 PDE 经典的数值解法, 例如有限差分方法 (FDM), 有限体积方法 (FVM), 有限元方法 (FEM) 等。

但是随着对精度, 模拟速度要求的提高, 传统计算方法也显现出一些弊端。而随着深度学习中自动微分技术的进步, 深度学习工具开始为解决这些问题注入新的活力。同时该交叉方向逐渐形成一个新兴的领域 “AI for Science”, 旨在通过 AI 加速数学、物理、化学等基础学科的研究。NeurIPS、ICML 等知名 AI 会议也都会定期举行 AI for Science 相关的研讨会。

1.2 对流扩散方程

对流扩散方程 (Convection Diffusion Equation) 是一类基本的偏微分方程，是扩散方程和对流（平流）方程的组合，描述了粒子、能量或其他物理量由于扩散和对流两个过程在物理系统内部转移的物理现象，在众多领域有着广泛的应用。方程通常可以写作

$$\mathcal{L}u = u_t - \nabla \cdot (A \cdot \nabla u) + b \cdot \nabla u + cu = f$$

其中 $\nabla \cdot (A \cdot \nabla u)$ 为扩散项， $b \cdot \nabla u$ 为对流项。特殊的，稳态对流扩散方程有 $-\nabla \cdot (A \cdot \nabla u) + b \cdot \nabla u + cu = f$ 。

1.3 经典数值方法

经典数值方法如 FDM, FVM, FEM 等，他们通常离散化时间域 $[0, T]$ ，用网格或三角形网络剖分空间区域 Ω 并在网格上建立简单基函数，最终将连续的微分方程转化为离散的线性方程组的形式。例如对于常微分方程 (ODEs) 而言，如 Runge-Kutta 方法。对于规则区域的微分方程而言，有限差分方法是非常有力的工具。对于不规则网格而言，FVM 和 FEM 提供了有力的算法和理论支持。虽然这些方法经过几十甚至上百年的发展，其适定性理论已经比较的完善，各种优质 (稳定，精确，唯一性) 的算法也得到了长足的发展，但我们仍需要指出一些传统方法的缺陷。

1. 经典求解算法遭遇维度灾难 (Curse of Dimensionality)。即随着求解问题维度的增加，如求解 Hamilton-Jacobi-Bellman Equation，数据点的存储以及求解速度指数级增加，可用数据变得异常稀疏，这点从系数 (刚度) 矩阵上就能看出。因此利用传统的基于网格的方法求解高维 PDE 几乎不具备可行性。

2. 难以结合实验数据，处理方程未知的情况。经典的数值算法理论上需要完整的方程表达式，包括初边值条件，而在实际的工业生产中往往不会有完整的解析表达式而往往以离散的数据的形式给出。这些数据很可能是稀疏的，残缺的，有噪声的。

3. 难以实现反问题 (Inverse Problems) 求解与设计。反问题用传统的方法非常困难。

1.4 深度学习方法

随着机器学习算法以及计算机资源的迅速发展, 可用数据的爆炸性增长, 深度神经网络 (Deep Neural Network) 已经在各个方面取得了突破性的成就, 如计算机视觉 (CV), 自然语言处理 (NLP), 强化学习 (RL) 等。借助这些算法的强大能力, 深度学习在很多现实领域取得了丰富的成果, 如数学, 工程, 药学, 材料科学, 地球科学等等。深度学习方法相较于传统计算有如下一些优势:

1. 处理大规模 (高维) 数据。他们能够从大量的数据中挖掘信息, 学习到复杂的模式和关联。这在复杂系统学习中显得尤为重要。同时, 深度学习方法能够通过随机采样的方式, 将 PDE 问题转化为机器学习问题, 避免了网格计算。因此, 在处理高维微分方程时, 基于深度学习的微分方程求解方法, 比传统方法更具有优势。
2. 非线性建模能力。神经网络能够更灵活的捕捉非线性关系。
3. 泛化能力强, 在复杂的混合问题上表现良好。尤其是一些物理量不能能够准确的已知, 并且数据具有噪声的情况。
4. 借助自动梯度算法, 避免离散化带来的误差累积, 能够有效的利用解析解的梯度信息来提高训练效率。

2 国内外研究现状

2.1 研究方向及进展

2.1.1 数据驱动 (Data Driven)

纯数据驱动的方法其实在十年前就有出现, 例如, Rudy 在 2017 年提出利用数据驱动稀疏回归方法来挖掘数据中隐藏的 PDE^[1]。

但这种方法有一些显然的缺点: 依旧遭受离散化带来截断误差的影响; 泛化能力差; 因此 Raissi 在 2018 年提出 Deep hidden physics models^[2]改进了这项工作, 利用深度神经网络作为代理模型, 大大缩减了模拟的时间, 提高了泛化能力。

同时又有一批工作希望寻求未知微分方程或其算子的函数近似结构表示, 通过学习相关数据找出背后蕴藏的微分方程模型, 进而预测系统动力学特性。其设计思想是期望采用

神经网络表示各种显示和隐式算子，学习函数到泛函的映射能力，而不是传统意义上用基函数的线性组合去逼近，进而实现在输入空间上任意取一点都能够返回相应的函数值。

代表的工作有 Lu Lu 在 2020 年的一篇工作介绍了 DeepONet^[3]，能够端到端的学习算子。Zongyi Li 在 ICLR2021 上提出了傅里叶神经算子 (Fourier Neural Operator, FNO)^[4]，借助傅里叶变换实现了无网格 (mesh-free) 的特点。这项工作在实际应用中已经有落地的成功案例，效果显著。

2.1.2 数据与机理结合 (Physics informed)

利用人工神经网络求解 PDE 早在上个实际 90 年代就有相关的工作，但那时候主要是将神经网络辅助传统数值求解算法^[5]。将数据和机理相结合的想法正式提出是由 Maziar Raissi 在 2017 年首先提出的 PINNs(Physics-informed Neural Networks)^[6]的概念。其想法主要是将 PDE 加入到损失函数中，也就是说用将物理先验信息以损失函数的形式加入到网络当中。即 $Loss = L_{MSE} + L_{PDEs}$, $L_{PDEs} = |f|^2 = |u_t - \mathcal{N}(u, u_x, u_{xx}, \dots)|^2$ ，其中的 \mathcal{N} 为偏微分算子，这种约束条件通常成为软约束 (Soft Constrain)。这项工作主要包括以下几点：

- 模拟固定时空区域内方程的数值解，预测未来步长解。
- 机理挖掘，即原始方程 (参数) 部分未知，从而预测出完整的方程 (参数)，可以算是简单的反问题。

这项工作于 2019 发表在 JCP 上，正式拉开数据与机理融合的时代。他的缺陷也比较明显：

- 对于不同的初边值条件需要重新训练，这意味着对于不同的问题实例需要重新构建和训练网络，这可能会导致较长的计算时间和工作量
- 不规则区域可能会影响收敛性，忽略内部的物理行为。

值得注意的是在 Raissi 的工作中，PDEs 通常以残差项的形式直接加入到损失函数当中。因此在随后的几年中，很多工作围绕如何提高模型学习性能，将 PDEs 作为物理先验信息更好的加入到损失项中展开了讨论：

如 Zang 在 2020 年提出将弱解形式加入 (Weak Adversarial Network)^[7], 这对于具有奇点而经典解不存在的问题有独特的优势, 同时在解决高维 PDE 上显现出很大的潜力。Weinan E 针对波动类方程提出以能量积分最小化的形式 (Deep Ritz Method)^[8]进行约束, 并将边界上的约束和作为惩罚项加入到损失函数中。还有一些工作如域分解, 网络收敛性研究等。

同时, 很多针对处理特殊的问题的 PINN 的变种开始涌现出来, 例如针对解决不规则边界的问题, 高维 PDE 问题, 多尺度问题等。

2.2 存在问题

2.2.1 物理先验信息 (Physical prior)

物理先验的信息通常会以 PDE 的形式给出, 同时还会有一些对称性, 周期性等硬约束 (Hard Constrain), 这通常会直接嵌入到网络结构当中。但将 PDE 仅仅加入到损失函数中似乎远远不够, 更多的物理先验知识的嵌入意味着模型更加的智能准确, 因此如何更好的将物理先验知识更好的嵌入到模型中是比较大的挑战。

同时在实际的应用当中, 数据的获取是十分昂贵且困难的, 因此得到的数据可能是稀疏的, 残缺的, 有噪音的。因此对于稀疏数据而言, 如何通过比较好的机理和数据融合方式使得模型能发挥最好的作用也是目前存在的问题之一。

2.2.2 高维与不规则区域

高维和不规则区域一直是经典数值算法的灾区, 因为由于计算维度的增加, 刚度矩阵开始变得稀疏, 存储量指数级增加。神经网络在高维 PDE 中展现出潜力, 但仍然存在较大的提升空间。尤其不规则区域的问题, 似乎是一个非常棘手的问题, 目前主要有以下几种方法: 1. 将区域信息以掩模的形式传入网络, 但效果较差, 因为区域信息和 PDE 信息割裂开。2. 利用图神经网络进行网格划分, 但是速度慢, 因为要根据数据三角化分割等种种原因。

2.3 研究展望

在深度学习和 PDE 相结合的领域未来的研究方向大概可以分为两个方面：

- 针对具体的不同问题方法本身的改进，例如针对方程多尺度现象，不规则边界如何处理等。
- 在不同的应用场景下实际的应用，例如多尺度耦合问题，多物理场问题等。

下面分别介绍：

2.3.1 多尺度问题 (Multiscale Problem)

计算精度是神经网络作为代理模型一直以来需要关注的问题，而多尺度问题所关注的便是：

1. 如何能够更加精确地同时捕捉到宏观的状态和微观的纹理细节，例如处理问题中存在的奇点，边界层，振荡现象等。因为虽然物理信息神经网络 (PINN) 在将物理模型与有间隙和噪声的观测数据集成方面表现出了非凡的前景，但是在要逼近的目标函数表现出高频或多尺度特征的情况下，它们仍然举步维艰。

2. 如何建模多尺度耦合的过程。由于一些过程在不同的尺度下本构方程不同，微观过程为宏观过程提供支撑，二者以某种方式耦合在一起。以地球系统为例，其动力学特性受到物理、化学和生物等过程相互作用影响，这些过程发生在跨越多个数量级的时空尺度，如何将不同尺度的方程耦合在一起是一项具有挑战性的工作。

2.3.2 多物理场 (Multi-physics problem)

目前多数利用深度学习方法来解 PDE 的方法都集中在一次解决单个 PDE，或者是一类 PDE，然而在实际应用的一些场景中往往是由多个物理场叠加而成。因此在模拟时候给定的不是单个 PDE，而是多个 PDEs 耦合在一起是值的考虑的问题。

例如，考虑一个导体内部的电流密度分布会导致热量的产生，这个热量会影响导体内部的温度分布，进而影响电阻率，从而影响电流密度分布。因此，电磁场和热传导是耦合在一起的，需要同时考虑它们的影响。

另一个例子是磁流体力学 (Magnetohydrodynamics), 即磁场和流场的耦合, 十分的复杂。这在等离子体物理、地球物理学以及一些工程应用中都有重要的应用。

2.3.3 与传统数值方法结合

可以看到目前利用神经网络的方法, 和经典的数值算法割裂有点大。然而对于深度学习算法而言, 他天然的和经典算法相融。例如, 卷积神经网络中的卷积核可以换为有限差分的刚度系数; 图神经网络中的三角分割天然的可以和有限元方法相契合。而深度学习算法借助自动梯度算法避免了差分的阶段误差, 因此如何更好的将两者相辅相成是目前存在的挑战之一。

3 参考文献

- [1] RUDY S H, BRUNTON S L, PROCTOR J L, et al. Data-driven discovery of partial differential equations[J/OL]. *Science Advances*, 2017, 3(4): e1602614. eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.1602614>. <https://www.science.org/doi/abs/10.1126/sciadv.1602614>. DOI: 10.1126/sciadv.1602614.
- [2] RAISSI M. Deep hidden physics models: deep learning of nonlinear partial differential equations[J]. *J. Mach. Learn. Res.*, 2018, 19(1): 932-955.
- [3] LU L, JIN P, PANG G, et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators[J]. *Nature Machine Intelligence*, 2021, 3(3): 218-229.
- [4] LI Z, KOVACHKI N, AZIZZADENESHELI K, et al. Fourier Neural Operator for Parametric Partial Differential Equations[J]., 2020.
- [5] DISSANAYAKE M W M G, PHAN-THIEN N. Neural-network-based approximations for solving partial differential equations[J/OL]. *Communications in Numerical Methods in Engineering*, 1994, 10(3): 195-201. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1640100303>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1640100303>. DOI: <https://doi.org/10.1002/cnm.1640100303>.
- [6] RAISSI M, PERDIKARIS P, KARNIADAKIS G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J/OL]. *Journal of Computational Physics*, 2019, 378: 686-707. <https://www.sciencedirect.com/science/article/pii/S0021999118307125>. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [7] ZANG Y, BAO G, YE X, et al. Weak adversarial networks for high-dimensional partial differential equations[J/OL]. *Journal of Computational Physics*, 2020, 411: 109409. <https://www.sciencedirect.com/science/article/pii/S0021999120301832>. DOI: <https://doi.org/10.1016/j.jcp.2020.109409>.
- [8] EE W, YU B. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems[J]. *Communications in Mathematics and Statistics*, 2017, 6. DOI: 10.1007/s40304-018-0127-z.
- [9] LONG Z, LU Y, MA X, et al. PDE-Net: Learning PDEs from Data[C/OL]//DY J, KRAUSE A. *Proceedings of Machine Learning Research: Proceedings of the 35th International Conference on Machine Learning*: vol. 80. PMLR, 2018: 3208-3216. <https://proceedings.mlr.press/v80/long18a.html>.
- [10] WANDEL N, WEINMANN M, KLEIN R. Learning Incompressible Fluid Dynamics from Scratch - Towards Fast, Differentiable Fluid Models that Generalize[C]//Ninth International Conference on Learning Representations. 2021.
- [11] IAKOVLEV V, HEINONEN M, LÄHDESMÄKI H. Learning continuous-time PDEs from sparse data with graph neural networks[J]., 2020.
- [12] GAO H, ZAHM M J, WANG J X. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems[J/OL]. *Computer Methods in Applied Mechanics and Engineering*, 2022, 390: 114502. <https://www.sciencedirect.com/science/article/pii/S0045782521007076>. DOI: <https://doi.org/10.1016/j.cma.2021.114502>.
- [13] GAO H, SUN L, WANG J X. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving Parameterized Steady-State PDEs on irregular domain[J/OL]. *Journal of Computational Physics*, 2020: 110079 [2021-01-04]. <https://linkinghub.elsevier.com/retrieve/pii/S0021999120308536>. DOI: 10.1016/j.jcp.2020.110079.
- [14] GULAKALA R, MARKERT B, STOFFEL M. Graph Neural Network enhanced Finite Element modelling[J/OL]. *PAMM*, 2023, 22(1): e202200306. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.202200306>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202200306>. DOI: <https://doi.org/10.1002/pamm.202200306>.
- [15] DE AVILA BELBUTE-PERES F, ECONOMON T, KOLTER Z. Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction[C/OL]//III H D, SINGH A. *Proceedings of Machine Learning Research: Proceedings of the 37th International Conference on Machine Learning*: vol. 119. PMLR, 2020: 2402-2411. <https://proceedings.mlr.press/v119/de-avila-belbute-peres20a.html>.
- [16] YAO H, GAO Y, LIU Y. FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction[J/OL]. *Computer Methods in Applied Mechanics and Engineering*, 2020, 363: 112892. <https://www.sciencedirect.com/science/article/pii/S0045782520301832>.

[//www.sciencedirect.com/science/article/pii/S0045782520300748](https://www.sciencedirect.com/science/article/pii/S0045782520300748). DOI: <https://doi.org/10.1016/j.cma.2020.112892>.

- [17] 舒其望. 计算流体力学中的间断 Galerkin 方法述评[J/OL]. 力学进展, 2013, 43(20130601): 541. <http://lxjz.cstam.org.cn/article/doi/10.6052/1000-0992-13-059>. DOI: 10.6052/1000-0992-13-059.

二、开题报告

1 问题提出的背景

1.1 背景介绍

计算流体力学 (CFD) 是一个应用非常广泛的领域, 较为基础、传统的是对于宏观尺度上的连续流体的模拟, 比如说飞机、车辆、船舶的空气动力学或水动力学。又如冷暖空气的流动, 预测洋流的变化等。这些宏观尺度的计算流体力学多出现在工业领域中, 有着较为成熟的发展。但是流体仍然是非常复杂的物理系统, 它的行为受纳维-斯托克斯方程 (Navier Stokes Function, i.e NS Function) 控制。

然而由于 NS 方程高度非线性、多维度、非恒定流动以及复杂的边界条件共同导致了通常情况下很难找到方程的解析解。因此, 经典的方法通常转向数值方法、渐进近似方法等技术来解决这些方程。例如基于网格的有限体积法, 有限元方法等, 这是 CFD 中广泛使用的数值方法, 但这些方法通常通过将计算域进行网格离散化, 最终将连续的微分方程转化为离散的线性方程组的形式, 这是一个计算成本高昂的过程。更准确和更先进的 CFD 仿真的主要瓶颈之一是运行这些模型所需的时间。而在大型超级计算基础设施上进行一次模拟需要几天到几周的时间并不少见。

同时随着问题的复杂程度升高, 计算资源指数增长, 计算速度精度要求不断提高, 传统的数值计算方法遇到了一定的挑战, 如维度灾难, 难以结合实验数据, 反问题的挑战等。近年来, 深度学习在计算机视觉和自然语言处理方面的成功应用, 促使人们探索人工智能在科学计算领域的应用, 特别是在计算流体力学 (CFD) 领域, 促使人们使用深度学习方法来逼近 PDE 解决方案, 并有望解决相关问题。

1.2 研究的意义和目的

虽然基于全连接物理信息的神经网络已经得到充分发展, 但是对于一些含有参数的 PDE, 例如未知的初边值条件, 未知的几何边界, 却少有研究, 大多数深度学习模型无法

捕获这些方程的全部复杂性，而这些问题对于 pde 求解来说也至关重要，他们严重影响模拟和计算的效率。因此卷积网络引起了科研人员的兴趣，因为卷积能够极大减少参数数量，加强了点与点之间的联系。在近些年也出现了一些基于 CNN 的 pde 代理模型。例如 Zichao Long 利用 CNN 中的卷积核代替有限差分方法^[9]；Nils Wandel 使用 U-Net 求解 Marker and Cell (MAC) grid 上的不可压缩 NS 方程^[10]，但这些方法并不专注于精确建模物理过程，相反它们的主要目标是生成逼真的动画。

但利用 CNN 作为代理模型也有其弊端，因为卷积需要结构化数据，即规范化网格上所有点的值，而针对不规则区域边界问题，我们通常会对区域进行三角化操作从而得到非结构化数据。且现实中更多重要的数据集都是用图的形式存储的，例如社交网络信息，知识图谱，蛋白质网络等。这些图网络的形式并不像图像是排列整齐的矩阵形式，而是非结构化的信息。这时图神经网络 (GNN) 便显得更为重要，最早在 21 年的一篇工作中研究了基于图神经网络用稀疏非结构化数据挖掘未知 PDE 机理^[11]。

然而在查找文献的过程中发现对于动态 NS 方程在不规则区域的模拟只有较少的工作，仅有的研究主要是针对稳态 PDE 在不规则区域的表现，例如利用 Galerkin 方法求解稳态的 NS 方程 (time-independent)^[12]。还有一种将利用坐标变换将不规则区域变成规则区域处理^[13]，但是这种方法我认为并没有真正让模型学习到不规则形状参数。当然也有将深度学习和有限元方法结合的尝试^{[14][15]}，但大多数都是利用 FEM 提供的数据，GNN 作为代理模型的应用，并没有结合 FEM 本身的优势。在 20 年的一篇工作中提出了 FEA-Net^[16]方法，考虑将有限元方法和深度神经网络结合起来，但是该方法虽然利用了有限元的思想，但是却只能限制在规则区域。

因此本课题中希望在已有的静态 NS 图神经网络代理模型进行改进，结合间断有限元方法的优势，能够以更快的速度，更好的精度实现不可压 NS 方程在不规则区域上的动态模拟。

2 可行性分析

目前有让模型学习不规则区域的工作。目前主要有以下几种方法：

1. 将区域信息以掩模的形式传入网络，将位置信息作为先验知识，例如以独热编码的形式传入。

2. 将不规则区域进行三角化 (网格化) 操作, 后用图神经网络处理。

在这里我认为第一种方法虽然将区域信息传入模型, 但是并没有让方程知识和区域信息结合起来,。而图神经网络受到有限元方法的启发, 由于偏微分方程残差的变分形式 (弱形式), 其中 Neumann 边界条件可以自然地包含在控制方程的弱形式中, 微分算子的阶可以通过部分积分有效地降低, 从而大大降低了学习的复杂性。因此我认为图神经网络在处理不规则区域 PDE 上有独特的优势。

同时目前也已经有利用图神经网络处理稳态 PDE 在规则区域上的工作, 而对于时空区域上微分方程的模拟我们主要有两种方法:

1. 利用 Method Of Line 将时间域进行离散化。
2. 利用自回归方法进行时间步长的迭代。

因此基于 GNN 的学习稳态 PDE 的方法可以自然的扩展到求解时空偏微分方程。

同时我们考虑间断 Galerkin 方法 (DG), 它结合了有限元方法和有限体积方法中的数值通量, 所以它可以更好的将物理信息引入模型学习中去。因此我们相信 DG 能够很好的和图神经网络结合, 理论上可以建立更精确更快速的数值模拟器。

3 主要内容

课题主要围绕以下几个关键词进行: 图神经网络 (GNN), 间断 Galerkin 方法, 有限体积法 (FVM), 时间依赖偏微分方程 (time-dependent pde), 不规则区域;

具体的问题可以写作: 在不规则区域上, 以图神经网络作为代理模型求解二维不可压缩 Navier-Stokes 方程, 如圆柱绕流模拟。I.e.

$$\begin{cases} u_t + u \cdot \nabla u = \frac{1}{\rho}(-\nabla p + \mu \Delta u + f) \\ \nabla \cdot u = 0 \text{ in } \Omega \times [0, T] \end{cases}$$

其中 Ω 为不规则区域, 且给出区域上稀疏点的非结构化数据。

主要尝试以下几个方法:

3.1 图神经网络

图神经网络是在图对象上运行的深度学习框架，是一种数据结构，通过节点和边自然的表征一组对象及其关系。

广义上来讲，GNN 利用节点和边的结构和特征来实现以下三个任务之一：

node classification; link prediction; graph classification;

GNN 的关键思想在于：学习如何通过图中所有节点的边，将信息从当前节点传播到整个图中。一般的 GNN 框架包含三个步骤来更新节点：

- **Message computation:** 类似与给节点编码，将节点的信息转换成一条“消息”。转换方法有很多种，常见的有将节点信息传入 MLP 中编码。
- **Aggregation:** 信息聚合，将相邻节点的信息进行聚合，合适的聚合方法有求和，平均值和最大值等
- **Updating:** 给定来自相邻节点的聚合消息，使用处理器函数更新每个节点的嵌入。

它直接对典型 CFD 模拟任务中使用的非均匀网格进行操作，而不是直接在大多数先前工作使用的规则网格上运行，因此有比常用的卷积操作要好很多。

3.2 间断 Galerkin 方法^[17]

间断 Galerkin (DG) 方法属于有限元方法，它结合了有限元法 (具有弱形式、有限维解和试验函数空间) 和有限体积法 (具有数值通量、非线性限制器) 的优点, 可以应用于任何形式的网格，允许不同的网格单元采用不同种类逼近函数空间，具有局部守恒性质。和传统有限元方法的主要区别在于在 DG 方法中有限元函数空间由分片多项式函数构成，允许在单元边界不连续。

主要有以下优点：

- **适用于复杂几何体：** 间断 Galerkin 方法在处理复杂的几何体时表现良好，因为它可以自然地适应非结构化网格。
- **高阶准确性：** 与传统的有限元方法相比，间断 Galerkin 方法通常具有更高的数值精度，尤其是在处理高阶 PDE 时。

- 处理激波和间断：由于其间断性质，这种方法对于处理包含激波和间断的问题非常有效，例如对于守恒律方程组的求解。
- 稳定性：间断 Galerkin 方法在一些流体动力学和固体力学问题中表现出良好的数值稳定性，特别是在处理激波和间断时。

3.3 DG-GCN 架构

将会利用 Dealii 开源 C++ 软件包或 COMSOL 等工业软件作为数据提供，以 pytorch 为框架编写实验代码，以图卷积神经网络为模型。

由于 GCN 的形式可以推导为

$$y = \sigma(Ug_{\theta}(\Lambda)U^Tx)$$

记给定的图结构 (Graph) 为 \mathcal{G} ， L 是其对应的 Laplace 矩阵，对 L 进行矩阵分解， U 是由 L 的特征向量构建的矩阵， Λ 是特征值构成的对角矩阵， $g_{\theta}(\Lambda)$ 是卷积核， x 是输入特征， σ 是激活函数。

对于卷积核而言，已有利用 $e^{-\frac{ik\pi x}{L}}$ 作为基函数，即以傅里叶变换为基础进行设计；Chebyshev 多项式函数空间为基础设计卷积核。在有限元方法中同样定义了例如连续分段多项式基函数空间

$$\mathcal{V}_h^p = \{v_h \in \mathcal{H}^1(\Omega) : v_h|_K \in \mathcal{P}_k(K), \forall K \in \mathcal{E}_h\}$$

因此理论上同样可以定义类似的卷积核进行优化。同时对于内罚间断 Galerkin 方法，可以将加罚项加入到损失项中。还有局部间断 Galerkin 方法等，具体方法选择还有待学习和探讨。

4 研究计划进度安排及预期目标

4.1 进度安排

2月26日-3月1日：开学前阅读相关论文，完成相关领域研究现状的探索和文献综述。完成开题答辩。

3月2日-3月21日：能够使用现有的 PhyGeoNet 以及 PIGGN 基本的求解器，复现相关论文。学习间断 Galerkin 方法 (Discontinuous Galerkin) 和有限体积方法。复现基于图神经网络利用 Galerkin 方法解决静态 PDE 的方法。利用 DealII 或工业软件如 SU2 计算出基础圆柱绕流实验数据。

完成 DG 和 GNN 两种方法对动态 (time-dependent)NS 方程在不规则区域上耦合的理论分析部分和结构设计，考虑将数值通量的概念加入模型学习过程。

3月22日-3月28日：完成中期报告，开始准备撰写论文，编写耦合部分程序代码。

3月29日-4月29日：完成求解器代码并进行数值对比实验, 撰写论文初稿, 修改后提交终稿。

5月26日-5月31日：毕业论文答辩

4.2 预期目标

在不规则边界条件下，利用图神经网络作为代理模型，结合经典的数值算法 (间断 Galerkin 方法) 的优势，完成对 NS 方程模拟的工作。期望在 NS 方程反问题以及正向模拟问题上超越经典算法，例如利用更少的数据和时间等，完成对这些问题的数值实验基准分析；具体实现代码并对算例进行计算，对网络结构设计进行理论分析，期望增强模型可解释性。

5 参考文献

- [1] RUDY S H, BRUNTON S L, PROCTOR J L, et al. Data-driven discovery of partial differential equations[J/OL]. *Science Advances*, 2017, 3(4): e1602614. eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.1602614>. <https://www.science.org/doi/abs/10.1126/sciadv.1602614>. DOI: 10.1126/sciadv.1602614.
- [2] RAISSI M. Deep hidden physics models: deep learning of nonlinear partial differential equations[J]. *J. Mach. Learn. Res.*, 2018, 19(1): 932-955.
- [3] LU L, JIN P, PANG G, et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators[J]. *Nature Machine Intelligence*, 2021, 3(3): 218-229.
- [4] LI Z, KOVACHKI N, AZIZZADENESHELI K, et al. Fourier Neural Operator for Parametric Partial Differential Equations[J]., 2020.
- [5] DISSANAYAKE M W M G, PHAN-THIEN N. Neural-network-based approximations for solving partial differential equations[J/OL]. *Communications in Numerical Methods in Engineering*, 1994, 10(3): 195-201. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1640100303>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1640100303>. DOI: <https://doi.org/10.1002/cnm.1640100303>.
- [6] RAISSI M, PERDIKARIS P, KARNIADAKIS G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J/OL]. *Journal of Computational Physics*, 2019, 378: 686-707. <https://www.sciencedirect.com/science/article/pii/S0021999118307125>. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [7] ZANG Y, BAO G, YE X, et al. Weak adversarial networks for high-dimensional partial differential equations[J/OL]. *Journal of Computational Physics*, 2020, 411: 109409. <https://www.sciencedirect.com/science/article/pii/S0021999120301832>. DOI: <https://doi.org/10.1016/j.jcp.2020.109409>.
- [8] EE W, YU B. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems[J]. *Communications in Mathematics and Statistics*, 2017, 6. DOI: 10.1007/s40304-018-0127-z.
- [9] LONG Z, LU Y, MA X, et al. PDE-Net: Learning PDEs from Data[C/OL]//DY J, KRAUSE A. *Proceedings of Machine Learning Research: Proceedings of the 35th International Conference on Machine Learning*: vol. 80. PMLR, 2018: 3208-3216. <https://proceedings.mlr.press/v80/long18a.html>.
- [10] WANDEL N, WEINMANN M, KLEIN R. Learning Incompressible Fluid Dynamics from Scratch - Towards Fast, Differentiable Fluid Models that Generalize[C]//Ninth International Conference on Learning Representations. 2021.
- [11] IAKOVLEV V, HEINONEN M, LÄHDESMÄKI H. Learning continuous-time PDEs from sparse data with graph neural networks[J]., 2020.
- [12] GAO H, ZAHM M J, WANG J X. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems[J/OL]. *Computer Methods in Applied Mechanics and Engineering*, 2022, 390: 114502. <https://www.sciencedirect.com/science/article/pii/S0045782521007076>. DOI: <https://doi.org/10.1016/j.cma.2021.114502>.
- [13] GAO H, SUN L, WANG J X. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving Parameterized Steady-State PDEs on irregular domain[J/OL]. *Journal of Computational Physics*, 2020: 110079 [2021-01-04]. <https://linkinghub.elsevier.com/retrieve/pii/S0021999120308536>. DOI: 10.1016/j.jcp.2020.110079.
- [14] GULAKALA R, MARKERT B, STOFFEL M. Graph Neural Network enhanced Finite Element modelling[J/OL]. *PAMM*, 2023, 22(1): e202200306. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.202200306>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.202200306>. DOI: <https://doi.org/10.1002/pamm.202200306>.
- [15] DE AVILA BELBUTE-PERES F, ECONOMON T, KOLTER Z. Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction[C/OL]//III H D, SINGH A. *Proceedings of Machine Learning Research: Proceedings of the 37th International Conference on Machine Learning*: vol. 119. PMLR, 2020: 2402-2411. <https://proceedings.mlr.press/v119/de-avila-belbute-peres20a.html>.
- [16] YAO H, GAO Y, LIU Y. FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction[J/OL]. *Computer Methods in Applied Mechanics and Engineering*, 2020, 363: 112892. <https://www.sciencedirect.com/science/article/pii/S0045782520301832>.

[//www.sciencedirect.com/science/article/pii/S0045782520300748](http://www.sciencedirect.com/science/article/pii/S0045782520300748). DOI: <https://doi.org/10.1016/j.cma.2020.112892>.

- [17] 舒其望. 计算流体力学中的间断 Galerkin 方法述评[J/OL]. 力学进展, 2013, 43(20130601): 541. <http://lxjz.cstam.org.cn/article/doi/10.6052/1000-0992-13-059>. DOI: 10.6052/1000-0992-13-059.

三、外文翻译

1 原文出处:

题目:

Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems

作者: Han Gao, Matthew J. Zahr, Jian-Xun Wang

期刊名: Computer methods in applied mechanics and engineering

2 摘要

尽管物理知情神经网络 (PINN) 在解决正问题和逆问题方面有着巨大的前景，但是一些技术挑战成为更复杂和更现实的应用的障碍。首先，大多数现有的 PINN 都是基于全连接网络来学习连续函数，这些函数具有较差的可扩展性和硬边界强制性。其次，无穷搜索空间使网络训练的非凸优化过于复杂。第三，尽管基于卷积神经网络 (CNN) 的离散学习可以显著提高训练效率，但是 CNN 难以处理非结构化网格的不规则几何形状。针对这些问题，提出了一种基于图卷积网络 (GCN) 和偏微分方程 (PDE) 变分结构的离散 PINN 框架，用于统一求解正、逆偏微分方程 (PDE)。使用分段多项式基可以降低搜索空间的维数，便于训练和收敛。该方法不需要对经典 PINN 中的惩罚参数进行调整，可以严格施加边界条件，同化正反两种情况下的稀疏数据。GCNs 的灵活性被用于非结构化网格的不规则几何图形。该方法在线性和非线性偏微分方程控制的各种正向和反向计算力学问题上显现出优势。

3 介绍

偏微分方程 (PDE) 在工程应用中占有重要地位，因为控制自然或人为复杂系统的大多数物理问题都是由偏微分方程描述的。然而，寻找解决大多数偏微分方程是一个具有挑战性的问题，这可能涉及复杂的数值技术和可能是耗时的，特别是在参数或初始/边界条件部分已知的情况下。最近，物理信息神经网络 (PINN)[1] 作为求解正向和逆向偏微分方程的一种新范式，由于其与传统的数值方法相比具有很大的灵活性和简单性而受到越来越多的关注。PINN 的主要想法是用深度神经网络来逼近 PDE 的解，手段主要通过结合 PDE 残差和数据偏差的损失函数。这种独特的损失函数使得物理信息的训练得到提升，不仅结合了 PDE 本身还有稀疏的数据。

对于如何建立 PDE 残差项的神经微分算子，PINN 可以被分为两类方法：连续和离散。连续的 PINNs 通常采用全链接神经网络来逼近关于时空坐标 (x, t) 连续的解函数 $f(x, t)$ ，其中关于空间和时间的导数项计算基于自动微分算法。由于 Raissi 等人 [1] 关于连续 FC-PINN 解决正向和反向偏微分方程惊艳的贡献，连续 PINN 正在经历一个复兴。它的优点和有效性已经在许多领域的大量科学应用中得到证明 [3-7]。例如，在流体应用中，PINN

已经被用于在没有训练标签的前向参数设置中的理想化血管流动问题的快速替代建模 [8]。此外, PINN 也已经在逆向建模设置中制定, 以从心血管问题中的可观察数据 (例如浓度数据) 中提取不可观察的信息 (例如血流速度)[9-12]。Jin 等 [13] 应用 FC-PINN 求解从层流到湍流的 Navier-Stokes 方程, 而 Mao 等 [14] 进一步证明了它们在高速流动问题上的有效性。最近, NVIDIA 开发了一个基于连续 PINN 的可扩展实现 SimNet, 并将其应用于解决具有大规模 GPU 并行化的各种多物理问题 [15]。

尽管由于其巨大的灵活性而取得了巨大的成功和快速的发展, 但是目前连续的 PINN 仍然存在一些局限性。首先, 它们受到高训练成本的影响, 因为在高维时空 (和参数) 域中的大量配置点上的点式制定需要大量的 AD 计算 [16,17]。其次, 对于连续的 PINN, 制定严格的初始/边界条件 (IC/BC) 是具有挑战性的, 这在寻找正确的单一 PDE 的解方面是有效的, 特别是当标记数据非常稀缺甚至不存在时 [8]。尽管可以引入基于距离的特定解决方案, 使用专门设计的代数表达式或低容量神经网络 [8,18], 在几个简单的 2-D 域上严格施加 IC/BC, 但是它未能显示对于真实世界应用的复杂几何形状的有效性。

为了降低训练成本, 提高学习效率, 利用卷积运算和数值离散化的离散 PINN 已经开始激发兴趣, 因为它们具有更好的效率和可伸缩性 [19,20]。具体而言, 在离散 PINN 中常用卷积神经网络 (CNN) 来直接学习整个时空解场的端到端, 并且基于数值离散代替逐点 AD 计算物理信息损失的所有导数项。例如, Zhu 等 [19] 开发了一种物理约束的卷积编码器-解码器来解决高维椭圆偏微分方程, 而 Geneva 等 [21] 进一步将该框架扩展到具有参数初始条件的动态双曲偏微分方程。Zhang 等 [22] 提出了一个物理指导的 CNN 用于地震响应建模, 也探索了类似的想法与一个递归神经网络 (RNN) 用于非线性结构的元建模 [23]。Wandel 等 [24] 最近提出了一种参数设置下基于自回归 U 网的无数据流体替代方法。在上述工作中, 计算区域是正则的, 离散化的统一网格, 其中偏微分方程残差计算的差分 (FD) 方法。这是因为基于 FD 的神经网络基本上植根于矩形区域的结构化笛卡尔网格。除了基于 FD 的 PINN 之外, 有限体积 (FV) 离散化也被用来构造基于 PDE 的损失函数来解决稳态流体问题, 然而, 由于经典卷积运算的固有局限性, 这仍然限于矩形域 [25]。为了使基于物理知识的 CNN 能够用非结构化网格解决不规则域上的参数偏微分方程, Gao 等 [20] 提出了一种基于几何自适应物理知识的 CNN, PhyGeoNet, 其将预先计算的坐标映射嵌入到经典的 CNN 结构中。尽管 PhyGeoNet 的有效性已经在简单的不规则域上得到了证明, 但

是对于一般的复杂几何形状来说，它仍然具有挑战性。

针对目前存在的问题，我们提出了一种基于广义卷积运算的非结构化网格的新的离散化 PINN 框架来处理不规则域。也就是说，卷积操作直接对非结构化网格数据进行，这些数据可以看作是离散的非欧几里德流形，即图。此外，PDE 信息图卷积网络 (GCN) 结构的构造受到有限元 (FE) 方法的启发 [26,27]，这是另一种经典的数值离散技术，对于物理知情学习具有许多优点。首先，由于偏微分方程残差的变分形式 (弱形式)，其中 Neumann 边界条件可以自然地包含在控制方程的弱形式中，微分算子的阶可以通过部分积分有效地降低，从而大大降低了学习的复杂性。此外，强形式 PINN 所需要的大量搭配点可以是被相对较少的正交点所取代，这可能潜在地减少相当大的训练成本。对于连续型 PINN，变分 (弱) 形式最近得到了发展，并且已经显示出对于强形式 PINN 的显著优越性。

在这些变分连续的 PINN 中，通常建立一个逐点全连通的神经网络，结合多项式测试函数，以 Petrov-Galerkin 的方式制定变分形式。由于深层神经网络具有黑盒特性，精确的正交法则难以建立，这导致与变分形式相关的额外误差。此外，基本边界条件不能以硬约束的方式加入。Yao 等 [34] 提出的 FEM-Net 是一个基于 FE 的离散 PINN，其中基于 FE 的卷积已被开发用于为 CNN 构建变分偏微分方程残差。然而，这种方法是在线性假设下，并且经典的 CNN 仍然受限于矩形区域。

本文提出了一种基于图卷积网络和偏微分方程变分结构的离散 PINN 框架，用于统一求解正、逆偏微分方程。具体而言，新的贡献概述如下：

(a) 将图卷积运算引入物理信息学习中，充分利用基于有限元的离散化方法对非结构化网格的不规则域进行离散化。与基于经典 CNN 的离散 PINN 不同，该方法不需要网格化，因为它可以像传统的有限元求解器那样直接处理带有单纯形/四边形单元的非结构网格。

(b) 采用一组有限维多项式基函数重建基于 Galerkin 公式输出节点解图的全场预测，从而大大缩小搜索空间，方便训练。此外，由于两个测试/试验函数都基于标准多项式，变分积分可以用高斯求积精确计算。

(c) 所建议的 PINN 设计能够准确地满足必要的边界条件，避免了在大多数 PINN 中使用 BC 软约束执行来调整惩罚系数。

(d) 提出一项新的数据同化方案，以严格执行观测数据。

4 方法

4.1 概述

考虑一个在有界区域 $\Omega \subset R^d$ 的物理系统, 由一组非线性, 稳定的参数化偏微分方程以通用的离散形式控制,

$$R(U(\mu); \mu) = 0$$

其中 $\mu \in R^{N_\mu}$ 为 PDE 的参数向量, $U: R^{N_\mu} \rightarrow R^{N_U}$ 为离散化参数相关的解, $R: R^U \times R^{N_\mu} \rightarrow R^U$ 代表离散 PDE 算子。偏微分方程组受边界条件 (BC) 的约束, 边界条件定义在区域的边界条件 $\partial\Omega$ 上。

在这项工作中, 我们针对这种偏微分方程控制系统在正向和逆向问题, 提出了一个创新的物理知情图神经伽辽金网络 (PI-GGN) 以建立一个解决方法。在正问题中, 我们的目标是得到给定已知边界条件和参数 μ 的解 U ; 对于反问题而言, 当系统的 BC 和参数 μ 部分已知时, 并且有状态的稀疏观测, 从而对问题进行求解。在所提出的框架中, 设计了一个 GCN 来学习一组非结构化网格上的状态节点解。基于连续伽辽金方法重构了物理损失函数中的偏微分方程残差项。系统的基本边界条件以硬约束的形式加入, 附加的数据可以被同化来同时求解正问题和逆问题。该方法的每个部分都将会在以下小节中详细说明。

4.2 对非结构化数据的图神经网络

由于 GCN 在处理非结构化数据方面具有极大的灵活性, 人们对将其应用于科学机器学习问题越来越有兴趣。通过经典的数据驱动, 基于图表的学习在模拟各种计算力学问题方面表现出色 [35-40]。通常, 通过定义非欧几里德空间的卷积运算, GCNs 将 CNN 类型的结构推广到图形数据。对图的节点之间的依赖关系进行建模的能力是使 GCNs 能够处理有任意边界的非结构化网格数据。从图 1 可以看出, 一个图由节点和边构成, 其中每个节点都是由自身的特征向量 f 定义, 而节点之间的关系就用边来描述。一个节点的邻接集 $N(\cdot)$ 指的是和该节点通过边相连的所有节点集合。因此, 一个具有非结构化网格和对应节点处的解的网格能够自然的被图自然的定义出来。和基于 CNN 的离散 PINN 类似, GCN 也用来解决离散的解 $U(\bar{\mu}) \approx \hat{U}(\Theta^*)$, 其中 Θ^* 是 GCN 对于 $\bar{\mu}$ 的训练参数。

注意： 由于深度神经网络的通用逼近能力，一般情况下，GCN 的输入特征向量可以由网格离散成任意空间变化的域。在这项工作中，GCN 接受一个输入图，其中每个节点与其网格的空间坐标相关联，然后将离散解域作为输出图输出，其中每个节点包含相应的节点解向量

和 CNN 类似，输出的解可以通过在输入层的多次图卷积操作得到，通过消息传递函数逐次更新节点信息。可以写成

$$\mathbf{f}_i^{(l)} = \gamma^{(l)} \left(\mathbf{f}_i^{(l-1)}, \square_{j \in \mathcal{N}(i)}^{(l)} \Psi^{(l)} \left(\mathbf{f}_i^{(l-1)}, \mathbf{f}_j^{(l-1)} \right) \right),$$

其中 i 记为第 i 个节点， l 记为第 l 层 $\gamma\Psi$ 为非线性可微函数， \square 记为可微置换不变函数 (例如，求和、平均值或最大值)。特征向量可以由 $\mathbf{f}_i^{(l)} \in \mathbb{R}^{N_f^{(l)}}$ 和 $\mathbf{f}_i^{(l-1)} \in \mathbb{R}^{N_f^{(l-1)}}$ 其中 $N_{f^{(l-1)}}$ 和 $N_{f^{(l)}}$ 分别为第 $(l-1)$ 和 l 层特征维度。

为了实现的简单性，所有的节点特性通常都被连接起来，变成一个更大的向量 \mathbf{X} 。边缘连接的信息存储在一个稀疏矩阵 \mathbf{A} 中，称为邻接矩阵。在实际应用中，边缘被用来隐含地表达邻接矩阵。在这项工作中，GCN 是基于 Chebyshev 谱图卷积算子 [41] 构建的，该算子源于谱卷积定理 [42]，其中引入了切比雪夫多项式以避免昂贵的特征分解。具体来说，Chebyshev 图卷积的消息传递函数可以写成：

$$\mathbf{X}^l = \text{ReLU} \left(\sum_{k=1}^K \mathbf{Z}^{(l-1,k)} \cdot \Theta^{(l-1,k)} + \mathbf{b}^{l-1} \right)$$

其中 $\Theta^{(l-1,k)}$ 为第 $(l-1)$ 层第 k 个基底的训练参数， $\mathbf{b}^{(l-1)}$ 为偏置值向量，第 k 个基递归计算方式如下：

$$\begin{aligned} \mathbf{Z}^{(l-1,1)} &= \mathbf{X}^{(l-1)} \\ \mathbf{Z}^{(l-1,2)} &= \hat{\mathbf{L}} \cdot \mathbf{X}^{(l-1)}, \\ \mathbf{Z}^{(l-1,k)} &= 2\hat{\mathbf{L}} \cdot \mathbf{Z}^{(l-1,k-1)} - \mathbf{Z}^{(l-1,k-2)}, \\ \hat{\mathbf{L}} &= \mathbf{L} - \mathbf{I} \\ \mathbf{L} &= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{\frac{1}{2}} \end{aligned}$$

在这里 K 设为 10.

4.3 变分形式的物理信息损失项

损失函数通常基于 PDE 残差项，稳定场景的 PDE 可以写成：

$$(6) \nabla \cdot F(u, \nabla u; \boldsymbol{\mu}) = S(u, \nabla u; \boldsymbol{\mu}) \text{ in } \Omega$$

其中 $u : \Omega \rightarrow \mathbb{R}^{N_c}$ 为解， $F : \mathbb{R}^{N_c} \rightarrow \mathbb{R}^{N_c \times d}$ 为流函数， $S : \mathbb{R}^{N_c} \rightarrow \mathbb{R}^{N_c}$ 为源项， $\nabla := (\partial_{x_1}, \dots, \partial_{x_d})$ 记为在区域中的微分算子。因此上式可以代表很大一部分静态偏微分方程，如泊松方程、线性弹性方程和纳维尔-斯托克斯方程等。

4.3.1 弱形式的 PDE 残差项

对于连续的 FC-PINNs 来说，计算损失项中的微分是通过自动微分在逐点处计算的，并且 FCNN 作为连续试探函数搜索无穷维解空间。因此，无限搜索空间使网络训练的非凸优化过于复杂，通常需要大量的配置点。在这项工作中，我们使用分段多项式基，以减少搜索空间的维数，并促进物理知识的训练/收敛。具体来说，守恒定律 (Eq. (6)) 采用节点连续伽辽金方法离散，用连续分段多项式基函数构造试验空间

$$\mathcal{V}_h^p = \left\{ v \in [\mathcal{H}^1(\Omega)]^{N_c} \mid v|_K \in [\mathcal{P}_p(K)]^{N_c}, \forall K \in \mathcal{E}_h \right\}$$

其中 $\mathcal{H}^1(\Omega)$ 为 Sobolev 空间， $\mathcal{P}_p(K)$ 为定义在单元 K 上次数至多为 p 的多项式空间， \mathcal{E}_h 为有限元网格。测试空间和 \mathcal{V}_h^p 相同且解 $u_h \in \mathcal{V}_h^p$ 对任意测试函数 $\omega_h \in \mathcal{V}_h^p$ 满足 PDE 的弱解形式，

$$\int_{\partial\Omega} \omega_h \cdot F(u_h, \nabla u_h; \boldsymbol{\mu}) n dS - \int_{\Omega} \nabla \omega_h : F(u_h, \nabla u_h; \boldsymbol{\mu}) dV = \int_{\Omega} \omega_h \cdot S(u_h, \nabla u_h; \boldsymbol{\mu}) dV.$$

我们引入 \mathcal{V}_h^p 的基函数 $\boldsymbol{\Phi}(x) \in \mathbb{R}^{N_U \times N_c}$ 来表示测试变量 $\omega_h(x) = \boldsymbol{\Phi}(x)^T \tilde{\boldsymbol{W}}$ ，其中 $\tilde{\boldsymbol{W}} \in \mathbb{R}^{N_U}$ 为基函数中变量系数，利用测试函数系数的任意性可以形成 Galerkin 形式的等价形式：

$$\int_{\partial\Omega} \boldsymbol{\Phi} \cdot F(u_h, \nabla u_h; \boldsymbol{\mu}) n dS - \int_{\Omega} \nabla \boldsymbol{\Phi} : F(u_h, \nabla u_h; \boldsymbol{\mu}) dV - \int_{\Omega} \boldsymbol{\Phi} \cdot S(u_h, \nabla u_h; \boldsymbol{\mu}) dV = 0.$$

我们引入 $\{(\beta_i^v, \tilde{x}_i^v)\}_{i=1}^{N_{qv}}, \{(\beta_i^s, \tilde{x}_i^s)\}_{i=1}^{N_{qs}}$ 分别作为在 $\Omega, \partial\Omega$ 上积分的权重和点, 并且定义残差为:

$$(10) \quad \begin{aligned} \mathbf{R}(\tilde{\mathbf{U}}; \boldsymbol{\mu}) = & \sum_{i=1}^{N_{qs}} \beta_i^s \Phi(\tilde{x}_i^s) \cdot F\left(\tilde{u}_h\left(\tilde{x}_i^s; \tilde{\mathbf{U}}\right), \nabla \tilde{u}_h\left(\tilde{x}_i^s; \tilde{\mathbf{U}}\right); \boldsymbol{\mu}\right) n - \\ & \sum_{i=1}^{N_{qv}} \beta_i^v \nabla \Phi(\tilde{x}_i^v) : F\left(\tilde{u}_h\left(\tilde{x}_i^v; \tilde{\mathbf{U}}\right), \nabla \tilde{u}_h\left(\tilde{x}_i^v; \tilde{\mathbf{U}}\right); \boldsymbol{\mu}\right) - \\ & \sum_{i=1}^{N_{qv}} \beta_i^v \Phi(\tilde{x}_i^v) \cdot S\left(\tilde{u}_h\left(\tilde{x}_i^v; \tilde{\mathbf{U}}\right), \nabla \tilde{u}_h\left(\tilde{x}_i^v; \tilde{\mathbf{U}}\right); \boldsymbol{\mu}\right), \end{aligned}$$

其中 $\tilde{u}_h : \Omega \times R^{N_U} \rightarrow R^{N_c}$ 为在 V_h^p 上离散状态向量的连续表示, 即:

$$\tilde{u}_h(x; \tilde{\mathbf{U}}) = \Phi(x)^T \tilde{\mathbf{U}}$$

表面和体积积分系数 (β^s 和 β^v) 存储为常数张量, 在网络训练过程中保持不变。基函数 Φ 的矩阵是在有限个求积点上得到的, 可以预先计算为常张量 ($\Phi(\tilde{x}^v), \Phi(\tilde{x}^s), \nabla \Phi(\tilde{x}^v), \nabla \Phi(\tilde{x}^s)$)。给出了偏微分方程残差 (Eq(10)) 将用于定义 GCN 的物理信息损失函数。即 GCN 学习节点解向量 \mathbf{U} 作为输出图 $U(\theta)$, 它以坐标 (X) 作为输入图。当偏微分方程参数 $\boldsymbol{\mu}$ 未知时, 它们可以作为可训练的参数, 与网络参数 θ 一起更新。流量函数和源函数 (F, S) 都是可微函数, 梯度信息可以从输出传播到输入。表 1 总结了这些注释。

4.3.2 必要的边界约束

我们通过限制无约束自由度, 即原理必要的 BC 来施加约束:

$$\mathbf{R}_u(\mathbf{U}_u(\boldsymbol{\mu}), \mathbf{U}_e; \boldsymbol{\mu}) = 0$$

其中 \mathbf{U}_e 为 BC 上已知值, $\mathbf{U}_u(\boldsymbol{\mu})$ 为无约束自由度 $\mathbf{U}(\boldsymbol{\mu})$ 对应的序号。在神经网络设置中, 我们通过将自由度划分为无约束 (未知) 和约束 (已知) 自由度 $\hat{\mathbf{U}}(\boldsymbol{\Theta}) = (\hat{\mathbf{U}}_u(\boldsymbol{\Theta})^T, \hat{\mathbf{U}}_c^T)^T$ 来强制执行边界条件。并使用基本 BC (即 $\mathbf{U}_c = \mathbf{U}_e$) 的已知值来定义约束自由度; 通过最小化物理信息损失函数定义无约束自由度, 损失函数可以写为:

$$\mathcal{L}_f(\boldsymbol{\Theta}; \boldsymbol{\mu}) = \left\| \mathbf{R}_u\left(\hat{\mathbf{U}}_u(\boldsymbol{\Theta}), \mathbf{U}_e; \boldsymbol{\mu}\right) \right\|_2.$$

在该公式中, 基本边界条件将通过构造自动满足, 这与连续 FC-PINN 形成鲜明对比, 连续 FC-PINN 将 FCNN 定义为逐点解函数, 这对硬边界执行提出了挑战。

4.4 统一正反问题

GCN 可以基于 Eq13 中定义的物理信息损失函数进行训练。解决下列无监督优化：

$$\Theta^* = \arg \min_{\Theta} \mathcal{L}_f(\Theta; \bar{\mu}),$$

其中 Θ^* 表示最优网络参数, μ 表示已知的偏微分方程参数, 然后用 GCN 求解正向偏微分方程 (正向解)。然而, 在许多情况下, 一些物理参数如材料性质、入口速度和雷诺数等是不可用的, 相反我们可以获得稀疏的观测数据 (标签) U_o , 这些数据可以同化来推断未知参数 (反解)。在先前的 PINN 方法中, 解决反问题以一种软约束的方式同化数据 U_o 来求解, 其中物理损失被数据损失分量增加。也就是说, 我们可以得到下面的优化公式:

$$(\Theta^*, \mu^*) = \arg \min_{\Theta, \mu} \mathcal{L}_f(\Theta; \mu) + \lambda \underbrace{\left\| \mathcal{F}^{s2o}(\hat{U}(\Theta)) - U_o \right\|_2}_{\text{data loss: } \mathcal{L}^d},$$

其中 \mathcal{F}^{s2o} 代表从状态到可观测的映射, λ 为惩罚因子。适当地调整惩罚权重 λ 对于收敛是至关重要的, 然而, 这是具有挑战性的, 并且经常以实证的方式进行 [16]。在这里, 我们介绍了一种新的方法, 同化观测数据和推断未知参数, 而不需要超参数调整。具体来说, 观测数据是通过将 GCN 输出构造为

$$\mathcal{F}^{s2o}(\hat{U}(\Theta)) = U_o.$$

因此, 通过求解以下约束条件, 可以同时得到未知参数 μ 和边界条件 $(\hat{U})_u$ 以及偏微分方程解:

$$(\Theta^*, \mu^*) = \arg \min_{\Theta, \mu} \mathcal{L}_f(\Theta; \mu), \quad \text{subject to: } \mathcal{F}^{s2o}(\hat{U}(\Theta)) = U_o.$$

算法 :

5 数值实验

我们展示了提出的物理知情图伽辽金神经网络 (PI-GGN) 在各种计算力学问题的正向和反向设置。具体来说, 本文研究了泊松方程、线性弹性方程和具有已知或未知 BCs/参数

算法 1: 算法 1 通过 GCN 解决 PDE 正问题

Input: PDE 参数 $\bar{\mu}$, 点集坐标 X 和邻接矩阵 A

Output: 解 \hat{U}

- 1 在点集上预计算矩阵基函数 Φ 得到 $\Phi(\tilde{x}^v), \Phi(\tilde{x}^s), \nabla\Phi(\tilde{x}^v), \nabla\Phi(\tilde{x}^s)$
 - 2 得到在 (10) 中定义的残差函数 $R(\tilde{U}; \mu)$
 - 3 带入计算得到 (12)
 - 4 划分自由度 $\hat{U}(\Theta) = (\hat{U}_u(\Theta)^T, \hat{U}_c^T)^T$; 强化边界条件 $\hat{U}_c = U_e$; 形成含有物理信息的损失函数;
 - 5 求解 (14) 中的优化问题来得到 $\hat{U} = (\hat{U}_u(\Theta^*)^T, U_e^T)^T$;
-

算法 2: 算法 2 通过 GCN 解决 PDE 反问题

Input: 稀疏的观测数据 U_o , 点集坐标 X 和邻接矩阵 A

Output: 解 \hat{U} 和 PDE 中参数 μ

- 1 在点集上预计算矩阵基函数 Φ 得到 $\Phi(\tilde{x}^v), \Phi(\tilde{x}^s), \nabla\Phi(\tilde{x}^v), \nabla\Phi(\tilde{x}^s)$
 - 2 得到在 (10) 中定义的残差函数 $R(\tilde{U}; \mu)$
 - 3 带入计算得到 (12)
 - 4 划分自由度 $\hat{U}(\Theta) = (\hat{U}_u(\Theta)^T, \hat{U}_c^T)^T$; 强化边界条件 $\hat{U}_c = U_e$; 严格施加观测数据 (16), 形成含有物理信息的损失函数;
 - 5 求解 (17) 中的优化问题来得到 μ^* 和 $\hat{U} = (\hat{U}_u(\Theta)^T, \hat{U}_c^T)^T$
-

的 Navier-Stokes 方程，以证明该方法的有效性。此外，我们还比较了两种不同的同化稀疏观测资料的方法，说明了严格执行参数/场反演资料的优越性。对于所有情况，GCN 体系结构保持不变，其中隐藏图层中的节点向量的维度固定为 $[32, 64, 128, 256, 128, 64, 32]$ 。相对误差度量 e 定义为，

$$e = \frac{\|\hat{U}(\Theta^*) - U(\bar{\mu})\|_2}{\|U(\bar{\mu})\|_2},$$

其中 Θ^* 为计算参数 $\bar{\mu}$ 的训练网络参数。

5.1 Poisson equation

2DPoisson 方程

$$\nabla u + f = 0 \text{ in } \Omega$$

$$u = 0 \text{ on } \partial\Omega$$

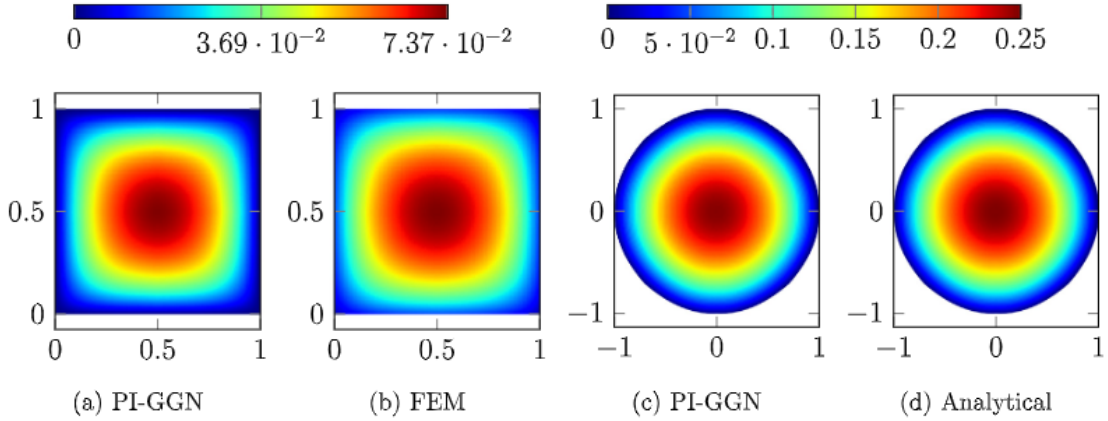


Fig. 2. PI-GGN forward solutions of the diffusion field u on the (a) square and (c) circular disks, compared against corresponding FEM or analytical solutions, where the relative prediction error of the PI-GGN is $e = 5 \times 10^{-3}$ on the square domain and $e = 5 \times 10^{-4}$ on the circular disk, respectively.

5.1.1 扩散场的正解

我们首先考虑正向问题，在单位平方域上给出源项 $f(f=1)$ (图 2a 和 2b)。采用四个四边形单元对区域进行离散，并给出了求解和区域变换的三阶多项式基。结果表明，该图的总节点数为 49 个，远低于典型的点式 FC-PINN 的配置点总数。PI-GGN 预测的轮廓与有限元参考轮廓非常吻合，相对误差为 $e = 0.5\%$ ，但在边界附近有误差。

在图 2c 中，相同的偏微分方程在一个单位圆形区域上求解，这里存在解析解 (图 2d) $u(x, y) = \frac{1-x^2-y^2}{4}$ 。

在 PI-GGN 中，元素个数保持不变，而多项式基的阶被设为两个，因此用 25 个节点构造图。结果表明，PI-GGN 正演解与解析解基本一致，相对预测误差仅为 0.05%。这个简单的测试案例表明，基于图的离散 PINN 可以很容易地处理非结构化网格的非矩形域，这对之前的标准的基于 FD 的 CNN 架构构成了挑战，因为他们需要光栅化或坐标变换等特殊处理 [20]，使实现和收敛复杂化。

5.1.2 源项的反问题求解

其实 PI-GGN 的真正强大的作用在于通过同化附加状态观测值同时求解正问题和逆问题。例如，当源项不给定时，PI-GGN 能够同化稀疏数据求解扩散场，同时统一推导未知源项。这里我们假设常数源项 $f=2$ 是未知的，观察 u 只能在图 3a 所示的一个点观察到。我们用两种方法同化数据并解决反问题：一种是在 (15) 中加入一个数据损失作为惩罚项来同化数据，其中超参数选取为 $\lambda=1000$ ，另一种是严格基于方程 (17) 的同化。从图 3 中我们可以看到他不仅收敛到了真实的源项而且能够正确的模拟出正解，且误差小于 1%。

5.2 Navier-Stokes 方程

在最后一个测试样例中，我们研究静态不可压缩 NS 方程的正反问题，由于他高度非线性的困难，显得非常具有挑战性。定常 NS 方程模拟了具有常密度的粘性流体流动，其表达式为

$$\begin{aligned} (v \cdot \nabla)v - \nu \Delta v + \nabla p &= 0, \quad \nabla \cdot v = 0 \quad \text{in } \Omega, \\ v &= v^D \quad \text{on } \partial\Omega^D, \\ \nu(n \cdot \nabla)v - pn &= 0 \quad \text{on } \partial\Omega^N, \end{aligned}$$

其中 $v: \Omega \rightarrow \mathbf{R}^d$ 为速度向量， $p: \Omega \rightarrow \mathbf{R}$ 为压力， ν 为流体的粘度， \mathbf{n} 为边界上向外的单位法向量。解向量记为 $u = [v_1, v_2, p]$ 。粘度 ν 设为 0.01。为了稳定性考虑，我们采取了一种混合有限元方法。我们构造了一个单独的子网，用于预测每个解变量 v_1 、 v_2 和 p 。

5.2.1 正向模拟解

首先，我们对一个经典的流动问题——空腔流动进行了测试，该问题定义在一个方形区域上。盖子放在顶部边缘并向右移动 ($v_1 = 1, v_2 = 0$)。其余三个边设置为无滑动边界 ($v_1 = v_2 = 0$)。该区域由 100 个四边形单元离散。速度场和压力场的配置点数分别为 441 和 121。

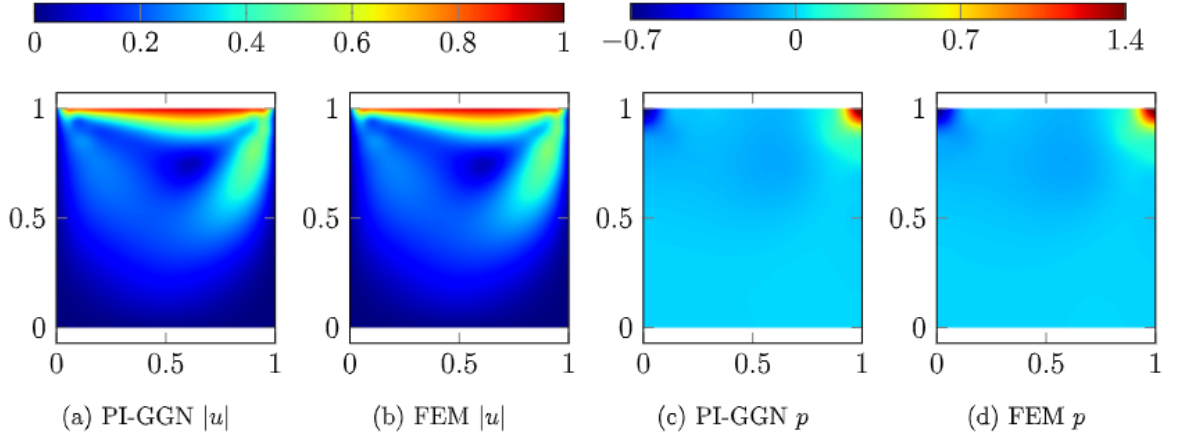


Fig. 9. PI-GGN forward solutions of the velocity magnitude and pressure fields, compared against corresponding FEM reference, where the relative errors are 8.7×10^{-3} for the velocity prediction and 1.95×10^{-2} for the pressure prediction.

如图 9 所示，PI-GGN 的速度和压力正解的轮廓与相应的有限元参考值非常吻合。相对预测误差小于 1%。值得注意的是，超过 10000 个搭配点被用来实现基于 AD 的 FC-PINN 的同等精度水平 [16,17]。

当然我们还测试了 PI-GGN 在理想狭窄中求解流体流动，其中入口速度设置为底部 ($y = 0$) $v^D = [0, 1]$ ，并且在顶部的出口处 ($y = 1$) 规定了无牵引边界条件。使用与盖驱动空腔问题相同的有限元设置。类似地，速度场和压力场都可以精确求解，PI-GGN 预测与 FEM 参考结果一致 (见图 10)

5.2.2 未知进口速度场和未观测压力场的反问题

最后，我们考虑一个由 NS 方程组的反问题。特别是当入口速度场未知时，将通过稀疏速度观测数据来推断，如图 11b 所示。真正的进气道具有如图 11e 所示的抛物线形状。剖面的函数形式在求解反问题时没有预先定义。也就是说，反转的尺寸等于进气道的自由

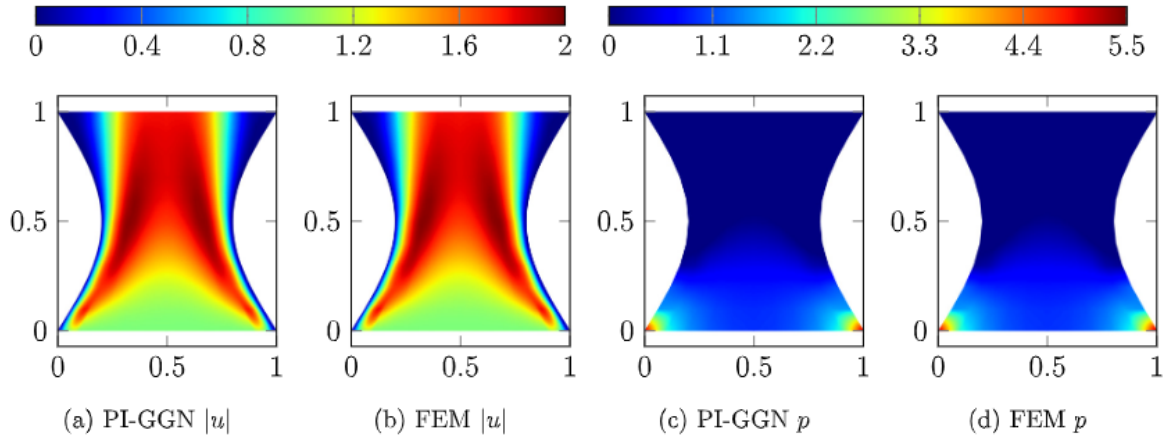


Fig. 10. PI-GGN forward solutions of the velocity magnitude and pressure fields, compared against corresponding FEM reference, where the relative errors are 4.4×10^{-3} for the velocity prediction and 1.8×10^{-2} for the pressure prediction.

度，大于 20。通过对稀疏位置的速度观测数据进行同化处理，该方法能够准确地推断出未知的进气道速度分布，并能很好地恢复整个进气道的速度场和压力场。

然而，我们发现基于惩罚的数据驱动方法推断的进气道并不十分准确，偏离了基本真理。尽管使用了与以前相同的惩罚系数，但推理效果明显恶化。提出的数据严格同化方法可以避免超参数调整，具有较好的鲁棒性。在观测数据有限的情况下，推断整个区域(空间场)的空间相关变量将使问题变得更加不适定。但这可以通过将空间场投影到径向函数或 Karhunen-Lo-eve 模式等基底上来解决，从而降低维数，减少问题的不适定性。

6 讨论

6.1 GCN 与 CNN 的对比

众所周知，经典的卷积神经网络最初是为图像识别和处理而开发的，其中卷积滤波器是为图像或类似图像的数据设计的，即具有均匀网格的矩形域。因此，CNN 不直接适用于本文所研究的案例。为了能够使用经典的 CNN，需要“光栅化”来预处理不规则域的非均匀网格数据 [45-47]。然而，光栅化也会引入一些问题，使得 PDE 约束的学习更加困难。

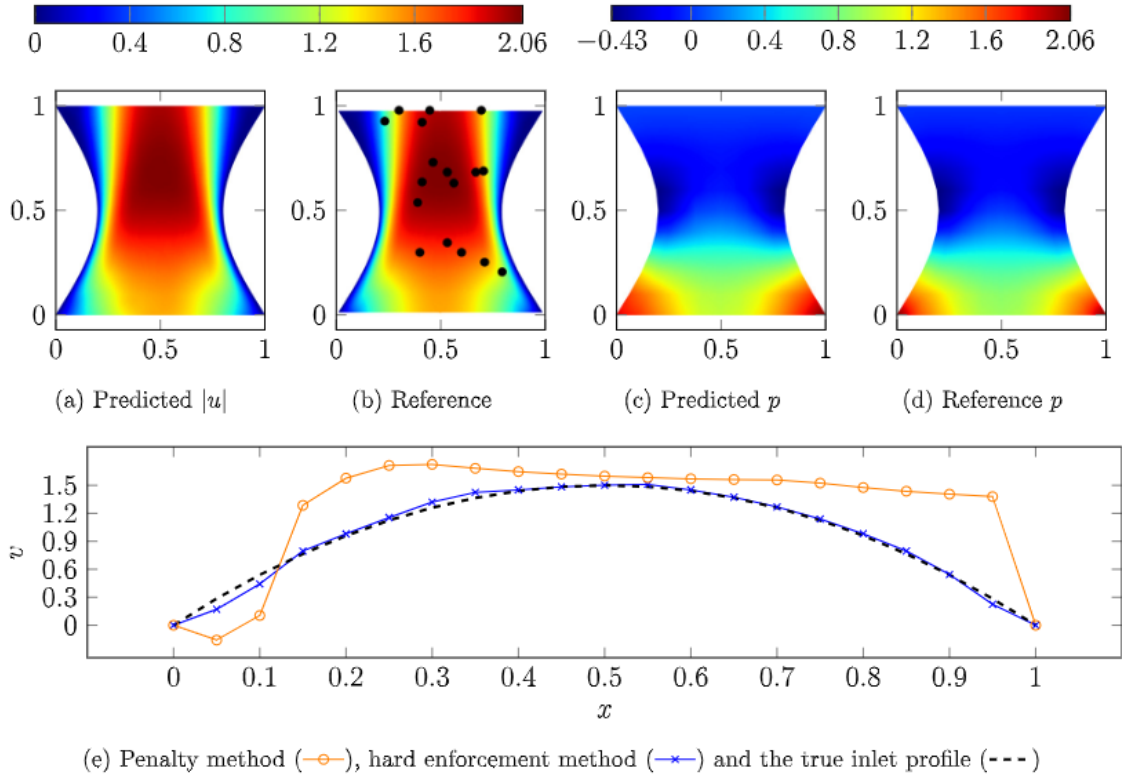


Fig. 11. PI-GGN inverse solutions of the inlet velocity field by assimilating observed velocity data at 19 randomly selected points. The relative error of the Inferred inlet profile is $e = 0.4$ by the soft penalty method while $e = 0.04$ by hard enforcement approach.

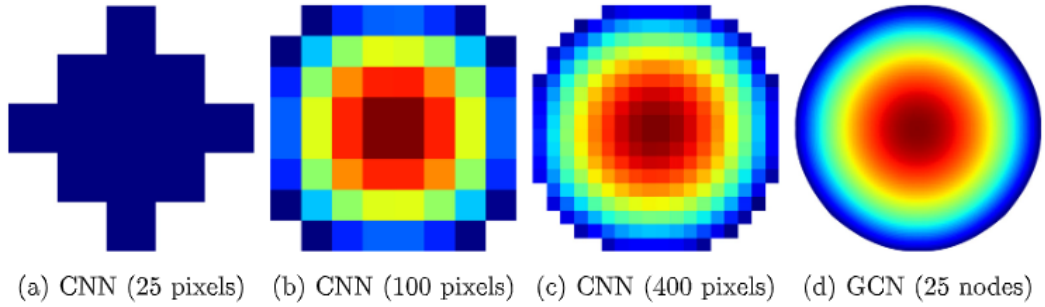


Fig. 12. The solution contours from CNN and GCN-based framework.

我们以单位圆盘为例，说明了所提出的基于 GCN 框架相对于 CNN 框架的优越性。图 12 比较了不同分辨率下 CNN 和 GCN 的解轮廓。即使是一个简单的圆形，对于 CNN 来说都需要大量的点 (见图 12c)。在非结构化网格中使用相同数量的节点 (25) 时，基于像素的

方法根本不起作用。由于缺乏平滑的分辨场和严格的边界条件，基于 CNN 的方法效果不理想。即使分辨率 $\times 8$ (400 像素)，CNN 的解也不如 GCN (大约低两个数量级) 精确。此外，与基于有限体积 (FV) 的迭代求解器相比，GCN 由于所需的迭代次数少而显示出优势。图 13 进一步显示了所提出的基于 GCN 的框架在训练效率和收敛速度方面的优点。

6.2 展望

这项工作最重要的贡献在于算法的发展，允许直接使用 GCN 解决正向和反向偏微分方程的非结构化网格。建议的框架是通用的，可以与各种最先进的通用网络控制器 [48-52] 结合使用 (在 pytorch 上的 geometric 上有很多例子)。虽然目前的工作主要集中在稳态偏微分方程，但随着进一步的发展，所提出的基于 GCN 的偏微分方程学习方法也可以扩展到求解时空偏微分方程。例如，物理信息损失函数中的时间导数可以基于正向欧拉方案来表示，并且可以构造基于图卷积的递归网络或变压器来捕获时间依赖性 [38]。

此外，该框架还可以适用于超分辨率 (SR) 任务，其中输入是一个有噪声的低分辨率场，输出是基于 PDE 信息的超分辨率训练。由于大多数现有的 SR 工作仍然严重依赖于 CNN，所以提出的框架有可能扩大 SR 在非结构化网格的不规则几何中的应用 [53,54]。

此外，我们亦可透过变分推断，将「一般通用网络」制定为一个贝叶斯网路，以处理数据噪音及量化不确定性 [55]。

7 总结

本文提出了一种新的离散 PINN 框架，用于统一求解由偏微分方程控制的正问题和逆问题。基于图卷积网络 (GCNs) 和物理信息损失函数的 Galerkin 变分公式，提出的 PINN 能够自然地处理非结构化网格的不规则域，由于多项式搜索空间减小，训练效率提高。由于边界条件和稀疏观测数据的强制执行，该方法不需要调整惩罚参数，具有较好的鲁棒性。通过对几个线性和非线性偏微分方程正反问题的数值计算，证明了该方法的有效性。此外，作者认为，这项工作有助于促进科学深度学习和经典数值技术的健康结合，而不是孤立彼此。

四、外文原文



Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 390 (2022) 114502

**Computer methods
in applied
mechanics and
engineering**

www.elsevier.com/locate/cma

Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems

Han Gao, Matthew J. Zahr, Jian-Xun Wang*

Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, United States of America

Received 16 July 2021; received in revised form 3 November 2021; accepted 18 December 2021

Available online 11 January 2022

Abstract

Despite the great promise of the physics-informed neural networks (PINNs) in solving forward and inverse problems, several technical challenges are present as roadblocks for more complex and realistic applications. First, most existing PINNs are based on point-wise formulation with fully-connected networks to learn continuous functions, which suffer from poor scalability and hard boundary enforcement. Second, the infinite search space over-complicates the non-convex optimization for network training. Third, although the convolutional neural network (CNN)-based discrete learning can significantly improve training efficiency, CNNs struggle to handle irregular geometries with unstructured meshes. To properly address these challenges, we present a novel discrete PINN framework based on graph convolutional network (GCN) and variational structure of PDE to solve forward and inverse partial differential equations (PDEs) in a unified manner. The use of a piecewise polynomial basis can reduce the dimension of search space and facilitate training and convergence. Without the need of tuning penalty parameters in classic PINNs, the proposed method can strictly impose boundary conditions and assimilate sparse data in both forward and inverse settings. The flexibility of GCNs is leveraged for irregular geometries with unstructured meshes. The effectiveness and merit of the proposed method are demonstrated over a variety of forward and inverse computational mechanics problems governed by both linear and nonlinear PDEs.

© 2021 Elsevier B.V. All rights reserved.

Keywords: Partial differential equations; Inverse problem; Physics-informed machine learning; Graph convolutional neural networks; Mechanics

1. Introduction

Partial differential equations (PDEs) play an important role in engineering applications since most of the physics governing natural or man-made complex systems are described by PDEs. However, finding solutions to most PDEs is a challenging problem, which may involve sophisticated numerical techniques and can be time-consuming, particularly for scenarios where parameters or initial/boundary conditions are partially known. Most recently, physics-informed neural networks (PINNs) [1], as a new paradigm for solving both forward and inverse PDEs, have attracted increasing attention due to its great flexibility and simplicity compared to classic numerical methods. The general idea of PINNs is to approximate the PDE solutions with deep neural networks, whose loss

* Corresponding author.

E-mail address: jwang33@nd.edu (J.-X. Wang).

<https://doi.org/10.1016/j.cma.2021.114502>

0045-7825/© 2021 Elsevier B.V. All rights reserved.

functions are formulated as a combination of PDE residuals and data mismatch. This unique loss formulation enables physics-informed training that leverages the information from both physics equations and sparse observation data.

Based on how to construct differential operators of PDE residuals using neural networks, PINNs can be classified into two categories: continuous and discrete. The continuous PINNs usually employ fully-connected (FC) neural networks to approximate the continuous solution function $f(\mathbf{x}, t)$ with respect to spatiotemporal coordinates (\mathbf{x}, t) in a point-wise manner, where the spatial and temporal derivative terms are computed using automatic differentiation (AD) techniques [2]. The continuous PINNs are undergoing a renaissance since recent impressive contributions made by Raissi et al. [1] on development of the continuous FC-PINN for solving forward and inverse PDEs. Its merit and effectiveness has been demonstrated over a plethora of scientific applications in many areas [3–7]. For instance, in fluid applications, PINNs have been used for fast surrogate modeling of idealized vascular flow problems in a forward parametric setting without training labels [8]. Moreover, PINNs have also been formulated in an inverse modeling setting to extract unobservable information (e.g., blood flow velocity) from observable data (e.g., concentration data) in cardiovascular problems [9–12]. Jin et al. [13] applied FC-PINNs to solve Navier–Stokes equations, ranging from laminar to turbulent regimes, while Mao et al. [14] further showed their effectiveness on high-speed flow problems. Recently, NVIDIA developed a scalable implementation SimNet based on continuous PINNs and applied it to solve various multiphysics problems with massive GPU parallelization [15].

Despite the enormous success and rapid developments thanks to their great flexibility, the current continuous PINNs still have some limitations. First, they suffer from high training cost since the point-wise formulation requires huge amount of AD computations on vast collocation points in a high-dimensional spatiotemporal (and parameter) domain [16,17]. Second, it is challenging to formulate a strict enforcement of initial/boundary conditions (IC/BCs) for continuous PINNs, which has been demonstrated to be effective in finding correct unique PDE solutions, especially when labeled data is very scarce or absent [8]. Although a distance-based particular solution can be introduced to strictly impose IC/BCs on a few simple 2-D domains using either specifically designed algebraic expressions or low-capacity neural networks [8,18], it fails to show the effectiveness on complex geometries for real-world applications. To reduce training costs and enable efficient learning, discrete PINNs that leverage convolution operations and numerical discretizations have begun to spur interests due to their better efficiency and scalability [19,20]. Specifically, convolutional neural networks (CNN) are often used in discrete PINN to directly learn the entire spatiotemporal solution fields end to end and all the derivative terms of the physics-informed loss are calculated based on numerical discretization instead of point-wise AD. For instance, Zhu et al. [19] developed a physics-constrained convolutional an encoder–decoder to solve high-dimensional elliptic PDEs, and Geneva et al. [21] further extended this framework to dynamic hyperbolic PDEs with parametric initial conditions. Zhang et al. [22] presented a physics-guided CNN for seismic response modeling and also explored the similar idea with a Recurrent Neural Network (RNN) for metamodeling of nonlinear structures [23]. Wandel et al. [24] recently proposed a data-free fluid surrogate based on an autoregressive U-net in a parametric setting. In aforementioned works, the computational domains are regular and discretized by uniform grids, where PDE residuals are calculated by finite difference (FD) methods. This is because the FD-based CNNs are fundamentally rooted in structured Cartesian grids of rectangular domains. Besides FD-based PINN, finite volume (FV) discretization has also been utilized to construct the PDE-based loss function to solve steady fluid problems, which, however, is still restricted to rectangular domains due to intrinsic limitations of classic convolution operations [25]. To enable physics-informed CNNs to solve parametric PDEs on irregular domains with unstructured grids, Gao et al. [20] proposed a geometry-adaptive physics-informed CNN, PhyGeoNet, which embeds a pre-computed coordinate mapping into the classic CNN structure. Although the effectiveness of the PhyGeoNet has been demonstrated on simple irregular domains, it remains challenging for general complex geometries at large.

Motivated by existing challenges, we propose a novel discrete PINN framework to handle irregular domains with unstructured grids based on generalized convolution operations. Namely, the convolution operations are directly performed on unstructured mesh data, which can be seen as discrete non-euclidean manifolds, i.e., Graph. Moreover, the construction of PDE-informed graph convolutional network (GCN) structure is inspired by finite element (FE) method [26,27], which is another classic numerical discretization technique that possesses many advantages for physics-informed learning. First, thanks to a variational formulation (weak form) of PDE residuals, where Neumann boundary conditions can be naturally incorporated in the weak formulation of governing equations, the order of differential operators can be effectively reduced by integration by part and thus the learning complexity can be largely mitigated. Moreover, the massive amount of collocations points required by strong-form PINNs can be

replaced by a relatively small amount of quadrature points, which could potentially reduce considerable training cost. The variational (weak) formulation has been recently developed for continuous PINNs and notable superiority has been shown over strong-form PINNs [28–33]. In these variational continuous PINNs, a point-wise fully-connected neural network is usually built as the trial basis, combined with polynomial test functions, to formulate the variational forms in Petrov–Galerkin fashion. Due to the black-box nature of the deep neural networks, accurate quadrature rules are difficult to construction, which leads to additional error associated with variational crimes. Moreover, the essential BCs cannot be imposed in a hard manner due to the point-wise formulation. The FEM-Net proposed by Yao et al. [34] is a FE-based discrete PINN, where a FE-based convolution has been developed to build variational PDE residuals for CNNs. However, this method is under a linear assumption and still limited to rectangular domains due to the classic CNN backbone.

In this work, we proposed an innovative discrete PINN framework based on graph convolutional network and variational structure of PDE to solve forward and inverse PDEs in a unified manner. Specifically, the novel contributions are summarized as follows:

- (a) We introduce the graph convolution operation into physics-informed learning to fully leverage the power of FE-based discretization for irregular domains with unstructured meshes. Unlike the state-of-art discrete PINNs based on classic CNNs, the proposed approach does not need rasterization as it can directly handle unstructured mesh with simplex/quadrilateral elements as traditional FE solver does.
- (b) A set of finite-dimensional polynomial basis functions are used to reconstruct the full-field predictions based on the output nodal solution graph in a Galerkin formulation, and thus, the search space can be significantly reduced to facilitate training. Moreover, since both test/trial functions are based on standard polynomials, the variational integrals can be computed accurately using Gaussian quadrature.
- (c) The proposed PINN is designed to exactly satisfy essential boundary conditions, avoiding penalty coefficient tuning in most PINNs with a soft BC enforcement.
- (d) A new data assimilation scheme is proposed to strictly enforce observation data.

2. Methodology

2.1. Overview

Consider a physical system in a bounded domain ($\Omega \subset \mathbb{R}^d$) governed by a set of nonlinear, steady parameterized PDEs in the generic discretized form,

$$\mathbf{R}(\mathbf{U}(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0, \quad (1)$$

where $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ is the PDE parameter vector, $\mathbf{U} : \mathbb{R}^{N_\mu} \rightarrow \mathbb{R}^{N_U}$ is the discrete parameter-dependent state vector implicitly defined as the solution of (1), and $\mathbf{R} : \mathbb{R}^{N_U} \times \mathbb{R}^{N_\mu} \rightarrow \mathbb{R}^{N_U}$ represents the discretized PDE operator. The set of PDEs are subjected to boundary conditions (BCs), which are defined on the boundary $\partial\Omega$ of the domain. In this work, we present an innovative physics-informed graph neural Galerkin network (PI-GGN) to establish a solution approach for such PDE-governed system in both forward and inverse settings. In the forward problem, we aim to obtain the solution \mathbf{U} given known BCs and parameters $\boldsymbol{\mu}$; as for the inverse setting, the system is solved when BCs and parameters $\boldsymbol{\mu}$ are partially known, whereas sparse observations of the state are available. In the proposed framework, a GCN is devised to learn nodal solutions of the state on a set of unstructured grids. The PDE residuals in the physics-informed loss function are reconstructed based on the continuous Galerkin method. Essential BCs of the system are imposed in a hard manner and additional data can be assimilated to solve the forward and inverse problems simultaneously. Each component of the proposed method will be detailed in the following subsections.

2.2. Graph convolutional neural network for unstructured data

There has been growing interest in applying GCN for scientific machine learning problems because of its great flexibility in dealing with unstructured data. Excellent performance of the graph-based learning has been reported in modeling various computational mechanics problems through classic data-driven training [35–40]. In general, by defining convolution operations for non-Euclidean space, GCNs generalize CNN-type constructions to graph data. The capability of modeling dependencies between nodes of a graph is the key that enables GCNs to handle

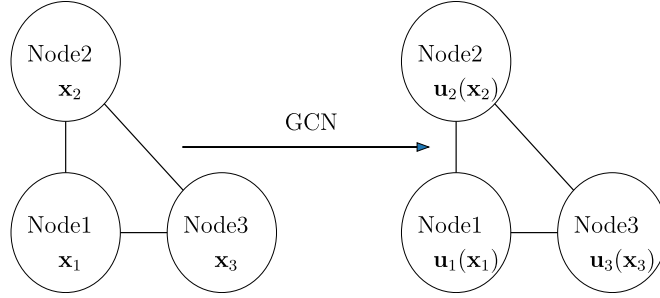


Fig. 1. An example of a GCN, where the input/output graph has 3 nodes & edges and the same adjacency matrix ($\mathcal{N}(1) = \{2, 3\}$, $\mathcal{N}(2) = \{1, 3\}$, $\mathcal{N}(3) = \{1, 2\}$). The input feature is the coordinate of each node ($f_i^{(in)} = x_i$), while the output feature is the nodal solution vector ($f_i^{(out)} = u_i(x_i)$).

unstructured mesh data with any arbitrary boundaries. As shown in Fig. 1, a graph consists of nodes and edges, where each node is defined by its feature vector \mathbf{f} and the relation with other nodes is described by edges. The neighbor $\mathcal{N}(\cdot)$ of a node refers to a set of adjacent nodes that are connected to that node via edges. Therefore, a mesh with unstructured grids and corresponding nodal PDE solutions can be naturally described as graphs. Similar to CNN-based discrete PINN [20], a GCN is built to model the discretized solution fields $\mathbf{U}(\bar{\mu}) \approx \hat{\mathbf{U}}(\Theta^*)$, where Θ^* are trained parameters of the GCN for graph convolutions for the parameter $\bar{\mu}$.

Remark. In general, the input feature vector of GCN can be any spatially varying field discretized by the mesh due to the universal approximation capacity of deep neural network. In this work, the GCN takes an input graph that each node is associated with its spatial coordinates of the mesh, and then outputs the discretized solutions fields as an out graph, where each node contains the corresponding nodal solution vector.

Similar to CNNs, the output solution graph is obtained by applying multiple graph convolution operations on the input layer, sequentially updating nodal features via a message passing function, which can be written in a generic form,

$$\mathbf{f}_i^{(l)} = \gamma^{(l)}(\mathbf{f}_i^{(l-1)}, \square_{j \in \mathcal{N}(i)}^{(l)} \Psi^{(l)}(\mathbf{f}_i^{(l-1)}, \mathbf{f}_j^{(l-1)})), \quad (2)$$

where i denotes i th node, (l) denotes l th layer, γ , Ψ are differentiable non-linear functions, and \square denotes a differentiable, permutation-invariant function (e.g., summation, mean, or maximum). The feature vectors are represented by $\mathbf{f}_i^{(l)} \in \mathbb{R}^{N_{f^{(l)}}}$ and $\mathbf{f}_i^{(l-1)} \in \mathbb{R}^{N_{f^{(l-1)}}}$, where $N_{f^{(l-1)}}$ and $N_{f^{(l)}}$ are feature dimensions in $(l-1)$ th and l th layers, respectively. For implementation simplicity, all the nodal features are usually concatenated and flattened as a larger vector \mathbf{X} . The information of edge connection is stored in a sparse matrix \mathbf{A} , known as the adjacency matrix. In practical implementation, the edges are used to implicitly formulate the adjacency matrix. In this work, the GCN is constructed based on the Chebyshev spectral graph convolution operator [41], which is derived from the spectral convolution theorem [42], where Chebyshev polynomials are introduced to avoid expensive eigen-decomposition. Specifically, the message passing function of Chebyshev graph convolution can be written as,

$$\mathbf{X}^l = \text{ReLU}\left(\sum_{k=1}^K \mathbf{Z}^{(l-1,k)} \cdot \Theta^{(l-1,k)} + \mathbf{b}^{l-1}\right), \quad (3)$$

where $\Theta^{(l-1,k)}$ are trainable parameters for the k th basis in the $(l-1)$ th layer, $\mathbf{b}^{(l-1)}$ is an additive trainable bias vector, and the basis $\mathbf{Z}^{(l-1,k)}$ is calculated recursively as follows,

$$\begin{aligned} \mathbf{Z}^{(l-1,1)} &= \mathbf{X}^{(l-1)}, \\ \mathbf{Z}^{(l-1,2)} &= \hat{\mathbf{L}} \cdot \mathbf{X}^{(l-1)}, \\ \mathbf{Z}^{(l-1,k)} &= 2\hat{\mathbf{L}} \cdot \mathbf{Z}^{(l-1,k-1)} - \mathbf{Z}^{(l-1,k-2)}, \end{aligned} \quad (4)$$

and

$$\begin{aligned}\hat{\mathbf{L}} &= \mathbf{L} - \mathbf{I} \\ \mathbf{L} &= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}\end{aligned}\quad (5)$$

where \mathbf{I} is an identity matrix and \mathbf{D} represents the degree matrix of the graph. The Rectified Linear Unit (ReLU) [43] is chosen as the nonlinear activation function and polynomial order K is set as 10 in this work.

2.3. Variational PDE-informed loss function

The loss function is built based on the PDE residuals (Eq. (1)), such that the conservation laws are utilized to inform/drive the GCN training. The generic PDE for steady-state scenarios can be re-written as,

$$\nabla \cdot \mathbf{F}(u, \nabla u; \boldsymbol{\mu}) = \mathbf{S}(u, \nabla u; \boldsymbol{\mu}) \quad \text{in } \Omega, \quad (6)$$

where $u : \Omega \rightarrow \mathbb{R}^{N_c}$ is the solution variable, $\mathbf{F} : \mathbb{R}^{N_c} \rightarrow \mathbb{R}^{N_c \times d}$ is the flux function, $\mathbf{S} : \mathbb{R}^{N_c} \rightarrow \mathbb{R}^{N_c}$ is the source term, and $\nabla := (\partial_{x_1}, \dots, \partial_{x_d})$ denotes the gradient operator defined in the physical domain. Eq. (6) can represent a wide range of static PDEs such as Poisson equation, linear elasticity equations, and Navier–Stokes equations.

2.3.1. Weak formulation of PDE residuals

For continuous FC-PINNs, the derivative terms for constructing the PDE-informed loss function are obtained by AD in point-wise manner, and the FCNN as a continuous trial function searches an infinite-dimensional solution space. Therefore, the infinite search space over-complicates the non-convex optimization for the network training and a massive amount of collocation points are usually required. In this work, we use a piecewise polynomial basis to reduce the dimension of the search space and facilitate physics-informed training/convergence. Specifically, the conservation laws (Eq. (6)) are discretized based using a nodal continuous Galerkin method and the trial space \mathcal{V}_h^p is constructed by continuous piecewise polynomial basis functions

$$\mathcal{V}_h^p = \{v \in [\mathcal{H}^1(\Omega)]^{N_c} \mid v|_K \in [\mathcal{P}_p(K)]^{N_c}, \forall K \in \mathcal{E}_h\}, \quad (7)$$

where $\mathcal{H}^1(\Omega)$ represents Sobolev spaces where weak derivatives up to order one are square integrable, $\mathcal{P}_p(K)$ is the space of polynomial functions of degree up to p defined on the element K , and \mathcal{E}_h is the finite element mesh. The test space is set to be the same as the trial space \mathcal{V}_h^p and the solution $u_h \in \mathcal{V}_h^p$ satisfies the weak formulation of the PDEs for any test function $\omega_h \in \mathcal{V}_h^p$,

$$\int_{\partial\Omega} \omega_h \cdot \mathbf{F}(u_h, \nabla u_h; \boldsymbol{\mu}) n \, dS - \int_{\Omega} \nabla \omega_h : \mathbf{F}(u_h, \nabla u_h; \boldsymbol{\mu}) \, dV = \int_{\Omega} \omega_h \cdot \mathbf{S}(u_h, \nabla u_h; \boldsymbol{\mu}) \, dV. \quad (8)$$

We introduce a basis $\boldsymbol{\Phi}(x) \in \mathbb{R}^{N_v \times N_c}$ for \mathcal{V}_h^p to express the test variables as $\omega_h(x) = \boldsymbol{\Phi}(x)^T \tilde{\mathbf{W}}$, where $\tilde{\mathbf{W}} \in \mathbb{R}^{N_v}$ are the coefficients of the test variable in the basis, which leads to an equivalent version of the Galerkin form

$$\int_{\partial\Omega} \boldsymbol{\Phi} \cdot \mathbf{F}(u_h, \nabla u_h; \boldsymbol{\mu}) n \, dS - \int_{\Omega} \nabla \boldsymbol{\Phi} : \mathbf{F}(u_h, \nabla u_h; \boldsymbol{\mu}) \, dV - \int_{\Omega} \boldsymbol{\Phi} \cdot \mathbf{S}(u_h, \nabla u_h; \boldsymbol{\mu}) \, dV = 0. \quad (9)$$

using arbitrariness of the test function coefficients. We convert this to residual form by introducing $\{(\beta_i^v, \tilde{x}_i^v)\}_{i=1}^{N_{qv}}$ and $\{(\beta_i^s, \tilde{x}_i^s)\}_{i=1}^{N_{qs}}$ as the quadrature weights and points for integrals over Ω and $\partial\Omega$, respectively, to define the residual as

$$\begin{aligned}\mathbf{R}(\tilde{\mathbf{U}}; \boldsymbol{\mu}) &= \sum_{i=1}^{N_{qs}} \beta_i^s \boldsymbol{\Phi}(\tilde{x}_i^s) \cdot \mathbf{F}(\tilde{u}_h(\tilde{x}_i^s; \tilde{\mathbf{U}}), \nabla \tilde{u}_h(\tilde{x}_i^s; \tilde{\mathbf{U}}); \boldsymbol{\mu}) n - \\ &\quad \sum_{i=1}^{N_{qv}} \beta_i^v \nabla \boldsymbol{\Phi}(\tilde{x}_i^v) : \mathbf{F}(\tilde{u}_h(\tilde{x}_i^v; \tilde{\mathbf{U}}), \nabla \tilde{u}_h(\tilde{x}_i^v; \tilde{\mathbf{U}}); \boldsymbol{\mu}) - \\ &\quad \sum_{i=1}^{N_{qv}} \beta_i^v \boldsymbol{\Phi}(\tilde{x}_i^v) \cdot \mathbf{S}(\tilde{u}_h(\tilde{x}_i^v; \tilde{\mathbf{U}}), \nabla \tilde{u}_h(\tilde{x}_i^v; \tilde{\mathbf{U}}); \boldsymbol{\mu}),\end{aligned}\quad (10)$$

毕业论文（设计）文献综述和开题报告考核

对文献综述、外文翻译和开题报告评语及成绩评定：

成绩比例	文献综述 占（10%）	开题报告 占（15%）	外文翻译 占（5%）
分值			

开题报告答辩小组负责人（签名）_____

年 月 日