DiFVM：面向非结构网格可微计算流体力学的向量化图式有限体积求解器

Pan Du[a,1],YongqiLi[a,b,1], Mingqi Xu[b],Jian-Xun Wang a,b*

[a]圣母大学航空航天与机械工程系，美国印第安纳州圣母镇[b]康奈尔大学西布利机械与航空航天工程学院，美国纽约州伊萨卡

arXiv:2603.15920v1 [cs.MS] 16 Mar 2026

摘要

可微编程已成为基于梯度的逆问题以及计算流体力学中端到端物理与机器学习混合方法的结构性前提。然而，现有可微计算流体力学平台仅适用于结构化笛卡尔网格，无法处理贴体非结构离散不可或缺的几何复杂域。我们提出了**DiFVM**，这是首个能够原生运行于非结构多面体网格上的 GPU 加速、端到端可微有限体积计算流体力学求解器。其关键突破在于发现有限体积离散与图神经网络消息传递之间存在结构同构：通过将所有有限体积法算子重新表述为网格连通图上的静态散射/收集原语，DiFVM 将不规则的非结构连通关系转化为一等 GPU 数据结构。所有运算均通过 JAX/XLA 实现，支持即时编译、算子融合，以及对完整仿真流程进行自动微分。针对心血管应用，DiFVM 提供可微的 Windkessel出口边界条件；同时，它无需修改即可直接读取标准 OpenFOAM 算例目录，便于无缝融入现有工作流程。涵盖典型流动至患者特异性血流动力学的基准测试正向验证表明，DiFVM 与 OpenFOAM 结果高度一致；此外，通过根据稀疏观测推断 Windkessel 参数，验证了其端到端可微性。DiFVM 弥合了可微编程与非结构网格计算流体力学之间的关键鸿沟，使复杂工程几何上的基于梯度的逆问题和物理集成机器学习成为可能。关键词：GPU加速，图消息传递，逆问题，心血管血流动力学，物理集成机器学习，JAX

*通讯作者。电话： +1 540 3156512电子邮箱：jw2837@cornell.edu（Jian-Xun Wang ）[1]这些作者对本研究贡献相同。

1. 引言

计算流体力学（CFD）长期以来一直是科学与工程领域建模和理解复杂流动现象不可或缺的工具，应用范围涵盖空气动力学、燃烧、生物流体系统和先进制造 [1–3]。数十年的发展造就了成熟的高保真求解器，能够在工程相关尺度上模拟复杂几何中的纳维–斯托克斯方程 [4]。从原理上看，经典的计算流体力学工作流程很直接：给定计算几何、指定的边界条件、物理参数和模型系数，通过数值求解控制方程获得流场。尽管这一正向模拟范式已取得巨大成功，但其有效性依赖一个在实践中鲜少成立的假设：所有模型输入均已知、准确且足以闭合系统。在大多数实际工程和生物医学问题中，边界条件往往只能被部分观测到，必须根据间接测量结果推断；材料或本构参数可能具有空间异质性，且难以准确表征；湍流闭合则会引入模型形式不确定性，因为其中未解析的物理过程无法仅凭第一性原理推导。这些挑战自然可表述为偏微分方程约束逆问题：通过确保模拟输出与现有观测或设计目标一致，恢复未知输入、参数或闭合项。典型示例包括根据稀疏的临床压力和流量测量结果恢复血管阻力和顺应性 [5, 6], ，在形状优化中根据压力分布推断气动表面几何形状 [7, 8], ，以及根据实验或高保真参考数据识别湍流闭合项的贡献 [9, 10]。关键在于，要大规模求解此类问题，必须设置外层优化循环，反复调用正向求解器，并依据梯度信息更新未知量。通过伴随方法或自动微分（AD）进行可扩展的灵敏度计算至关重要，因为在计算流体力学应用典型的高维参数空间中，有限差分或无导数策略的计算成本会高到难以承受。

随着观测数据日益丰富，以及机器学习与计算流体力学的混合化成为科学计算中的主流范式，对梯度访问的需求也愈发迫切 [11]。大规模实验数据库、高分辨率传感器阵列和医学成像的兴起，为数据增强型计算流体力学开辟了新的机遇：在这类工作流程中

测量会主动影响或修正模拟，而不只是用于验证模拟。数据同化方法将模拟输出与观测相融合，从而得到比任一信息源单独提供的结果都更准确的状态估计 [12, 13]，但这需要梯度将信息从测量空间反向传递，穿过控制方程并进入模型状态 [14, 15]。机器学习与计算流体力学的混合化是朝这一方向迈出的又一步，也更为大胆：机器学习组件不再只是将数据同化到现有模拟状态中，而是经过训练，学习控制方程本身所包含的未知或未解析物理过程。早期的混合方法采用松耦合架构：利用高保真模拟数据离线训练机器学习模型，随后将其作为闭合模型嵌入传统计算流体力学求解器中 [16, 17]，尤其是在湍流建模领域，数据驱动的亚网格尺度闭合模型和雷诺应力修正已得到广泛探索 [18–22]。尽管这些模型在某些工况下有效，但离线训练的模型从根本上说并不符合计算流体力学的一致性要求。这是因为训练过程与求解器彼此脱节，而且无法保证这些已学习的闭合模型在前向模拟循环中稳定或准确 [23]。这具体表现为 后验性能退化，即模型在针对参考数据进行先验评估时表现良好，但集成到实际运行的求解器后却出现明显误差或数值不稳定性 [24–27]其根本原因在于反馈回路被切断：梯度无法通过求解器传播，机器学习模型因而无法学会修正部署过程中累积的误差 [28]。

这一局限促使研究转向通过可微编程实现完全耦合的物理–机器学习混合框架 [29]，，在这种框架中，机器学习组件与计算流体力学算子在单一可微计算图中进行端到端联合优化。当梯度穿过求解器，经过通量计算、压力求解和时间积分步传播时，神经网络学到的表示与数值求解器动力学保持一致，而不仅仅与离线训练数据相符。这一范式推动求解器在环学习型湍流闭合 [30, 31]，、数据驱动离散化和端到端求解器修正 [32–34]„ 以及带有可学习数值组件（例如通量或粗粒度模型增强）的混合可微求解器 [35, 36]取得了显著进展。至关重要的是，可微混合建模不止于将机器学习嵌入计算流体力学求解器：它还支持偏微分方程集成神经架构，使计算流体力学求解器或其离散算子能够作为结构化层嵌入高级神经架构之中 [37]。在这种神经可微建模范式下 [38–40]，，离散化偏微分方程算子充当固定的物理——

编码层与可训练组件交错排列，例如 ConvLSTM 块、图核或残差网络，从而构成物理一致性和数据效率均优于纯黑箱替代方案的物理集成代理模型。Wang 及其合作者已在多个领域展示了这一方向的应用，包括湍流建模 [41], 流固耦合（FSI） [28], 以及多尺度热输运 [42, 43]。例如，在用于湍流和流固耦合的混合神经可微求解器中，离散化纳维–斯托克斯算子被编码为固定的卷积层，置于循环架构中，并与可训练的 ConvLSTM 块及扩散生成模块交错组合；这些组件能够学习亚网格尺度修正并生成小尺度湍流结构，由此形成的是物理集成神经网络，而非经神经网络增强的求解器 [28, 41]。在所有这些情形中，穿过物理算子的梯度路径使模型能够利用稀疏间接数据进行符合物理规律的学习。因此，求解器的可微性并非便利之选，而是该领域正逐渐趋向的物理–机器学习混合架构的结构性前提。

尽管这一要求很明确，传统计算流体力学求解器却无法满足。大多数生产级代码采用 C++ 或 Fortran 实现，以针对非规则网格连接关系的显式循环为核心组织结构，并通过 MPI [4]优化 CPU 并行计算。过去，这些系统通常通过伴随方法（例如 SU2 [44], DOLFIN-adjoint [45],ADjoint [46], STAR-CCM+ [47]）来获取梯度，但伴随方法的实现会侵入代码、针对特定求解器，而且每种新的问题配置都需要大量重新推导。更重要的是，这些实现与 JAX [48], PyTorch [49], TensorFlow [50], Julia [51], 等自动微分生态系统在架构上彼此割裂，而现代神经网络组件正是在这些生态系统中构建和训练的。这种割裂使物理算子无法与可学习模块在单个可微计算图中无缝组合，而这正是物理–机器学习混合框架所必需的。除了可微性之外，传统求解器还存在第二种结构性不兼容：它们是原生 CPU 的，依赖于对非规则连接关系进行基于循环的执行，难以映射到现代图形处理器和张量处理器的单指令多数据（SIMD）和单指令多线程（SIMT）执行模型上。即使能够获得伴随梯度，底层求解器的速度仍比原生 GPU 机器学习基础设施慢几个数量级，因此，在现代应用规模下，迭代式梯度驱动工作流程在计算上并不实际。

这些局限 ns 推动了原生 GPU 可微计算流体动力学平台的兴起，旨在实现 -

。 [34, 42, 53–57]代表性框架包括 JAX-CFD [34], ，这是一个基于 JAX 的不可压缩流求解器，采用结构化笛卡尔网格，广泛应用于机器学习与计算流体力学混合研究；PhiFlow [53], ，这是一个与后端无关的可微偏微分方程框架，可利用简化的流体模型快速实现基于梯度的优化原型；JAX-Fluids [55], ，这是一个适用于可压缩流和多相流的高阶有限体积求解器，支持多 GPU 可扩展性，但其侧重可压缩流且采用结构化笛卡尔拓扑，因此不适用于心血管、流固耦合及大多数工业应用所涉及的低马赫数不可压缩流态；WaterLily [54], ，它通过笛卡尔浸入边界法处理绕运动物体的流动，但缺少湍流闭合模型和双向结构耦合；以及 PICT [56], ，它展示了瞬态动力学中的基于梯度的反演建模，但针对的是结构化离散上的简化物理问题。Diff-FlowFSI [57], 则代表了重大进展，它是首个专为工程尺度湍流和强耦合多物理场流固耦合而设计的可微计算流体力学平台。与以往框架不同，Diff-FlowFSI 支持壁面约束湍流的亚网格尺度模型和壁面模型，并实现不可压缩纳维–斯托克斯方程与弹性结构动力学之间的强双向耦合；这一切均基于完全即时编译、向量化的 JAX 实现，在保持完整仿真流程端到端可微的同时，速度较 CPU 并行化计算流体力学求解器最高提升两个数量级。总的来说，这些平台已使 GPU 原生可微仿真成为梯度驱动工作流程的实用基础，并可应用于日益广泛的物理系统。

尽管取得了这些进展，现有的所有可微计算流体力学框架仍存在一个根本且尚未解决的局限：它们仅适用于结构化笛卡尔网格。这一限制反映了真正的算法挑战：非结构网格具有不规则的单元连接关系、数量不定的面邻居、几何非正交性以及无法合并的内存访问模式，这些特性从根本上不利于现代 GPU 硬件采用向量化 SIMD/SIMT 执行模型。然而，对于决定最重要的计算流体力学应用（例如患者特异性血管解剖结构、涡轮机械流道、飞机表面，以及具有不规则边界的一般工程几何）的几何复杂域，非结构网格不可或缺；要准确分辨边界层并实现符合物理规律的通量计算，就必须采用贴体离散。有限体积法是处理此类几何的自然选择，具有与几何无关的表述、严格的局部守恒性，并能稳健处理任意单元

拓扑结构和边界条件。因此，缺少一种 GPU 原生、端到端可微且适用于非结构网格的有限体积法求解器，并非无关紧要的缺口，而是一个根本瓶颈，限制了整个可微分计算流体力学生态系统的几何适用范围，并进而制约建立在其上的物理–机器学习混合框架。据我们所知，目前尚无现有框架能够提供一种 *GPU* 高效、完全向量化、端到端可微且原生运行于三维非结构网格上的有限体积计算流体力学求解器。

在本研究中，我们通过提出 **DiFVM**弥合这一差距。DiFVM 是首个 GPU加速、向量化且端到端可微的有限体积计算流体力学求解器，可原生处理非结构多面体网格。其关键在于发现有限体积法离散与图神经网络（GNN）的消息传递范式之间存在结构同构：从单元中心量计算单元间通量，并将其聚合到各控制体中，恰好对应图神经网络框架通过散射/收集原语以完全向量化、GPU 原生方式执行的局部计算与邻域聚合模式 [58]。通过将所有有限体积法算子重新表述为静态的基于图的消息传递原语，非结构网格的不规则连接便从计算负担转化为一等数据结构，适于 GPU 向量化；JAX 的 XLA 编译器则提供即时编译、算子融合，以及贯穿完整流程的自动微分。DiFVM 在这一基于图的框架中表述所有算子，以求解不可压缩纳维–斯托克斯方程，并通过 `jax.lax.scan` 展开时间积分。它支持任意非结构多面体网格，包括纯四面体、六面体、棱柱体及任意混合拓扑结构，并可直接兼容 OpenFOAM 算例目录：现有 OpenFOAM 算例无需修改即可交由 DiFVM 处理，从而能够无缝融入成熟的模拟工作流程。针对心血管应用，此外还实现了三元 Windkessel （ RCR ）出口边界条件，以支持患者特异性血流动力学模拟，并根据稀疏的临床测量进行端到端参数推断。本研究的主要贡献如下：

- **首个面向非结构网格的 GPU 原生矢量化可微有限体积法求解器**。DiFVM 支持完全非结构化的多面体网格，能够处理任意单元类型，包括四面体、六面体、棱柱体、棱锥体及其混合组合，并支持非正交修正和复杂边界条件。所有现有的可微计算流体力学框架都不具备这些功能。

- **基于图的有限体积算子矢量化**。对流通量、扩散通量和 PISO 压力–速度耦合被重新表述为静态消息传递图原语，并编译为融合的 XLA GPU 内核；在大规模瞬态基准测试中，其性能显著优于采用中央处理器并行化的 OpenFOAM。

- **可直接替代使用，并兼容 OpenFOAM**。 DiFVM 无需修改即可直接读取 OpenFOAM 算例目录，因此能够立即融入现有工作流程，并便于在相同数值设置下进行定量比较。

- **适用于心血管应用的可微分 Windkessel 边界条件**。DiFVM 在可微分流程中实现了三元 Windkessel（RCR）出口边界条件，支持在解剖结构逼真的血管几何结构上开展患者特异性血流动力学模拟。

- **在多个基准测试中进行了全面的正向验证**。 通过三维泊松问题、被动标量对流、顶盖驱动方腔流、弯管流、圆柱体涡脱落以及患者特异性多分支主动脉流，展示了其准确性，并与 OpenFOAM 进行了定量比较 [59] 以及 SimVascular [60]。

- **通过逆问题验证了端到端自动微分**。 自动微分流程通过以下任务得到验证：基于梯度反演方腔流的顶盖边界条件，以及从脉动血流中的稀疏压力和流量观测中同时推断六个 Windkessel 参数。

本文其余部分安排如下。第2 节介绍 DiFVM 方法，包括有限体积离散、求解器格式和面向GPU的向量化策略。第3 节展示基准测试验证、效率比较和反问题演示。第4 节讨论当前的局限性与未来方向。第5 节总结全文。

2. 方法

2.1. 概述

DiFVM 是一款专为可微分计算而设计的有限体积计算流体力学求解器，旨在解决非结构网格上缺乏 GPU 原生、端到端可微分模拟的根本问题。

如图1所示，DiFVM 围绕四项紧密集成的能力构建： *(i)* 原生支持任意非结构多面体网格； *(ii)* 将所有有限体积法算子重新表述为基于图的消息传递原语，实现全面的 GPU 向量化； *(iii)* 实现贯穿完整仿真流程的端到端自动微分，涵盖从边界条件和物理参数到每一次通量计算、线性求解和时间积分；以及 *(iv)* 即插即用地兼容 OpenFOAM，并为心血管应用提供可微分的温塞尔边界条件。所有数值运算均在 JAX/XLA 生态系统中原生实现，支持即时（JIT）编译、算子融合和 GPU加速执行，其速度比 CPU 并行化计算流体力学求解器快一个至两个数量级。

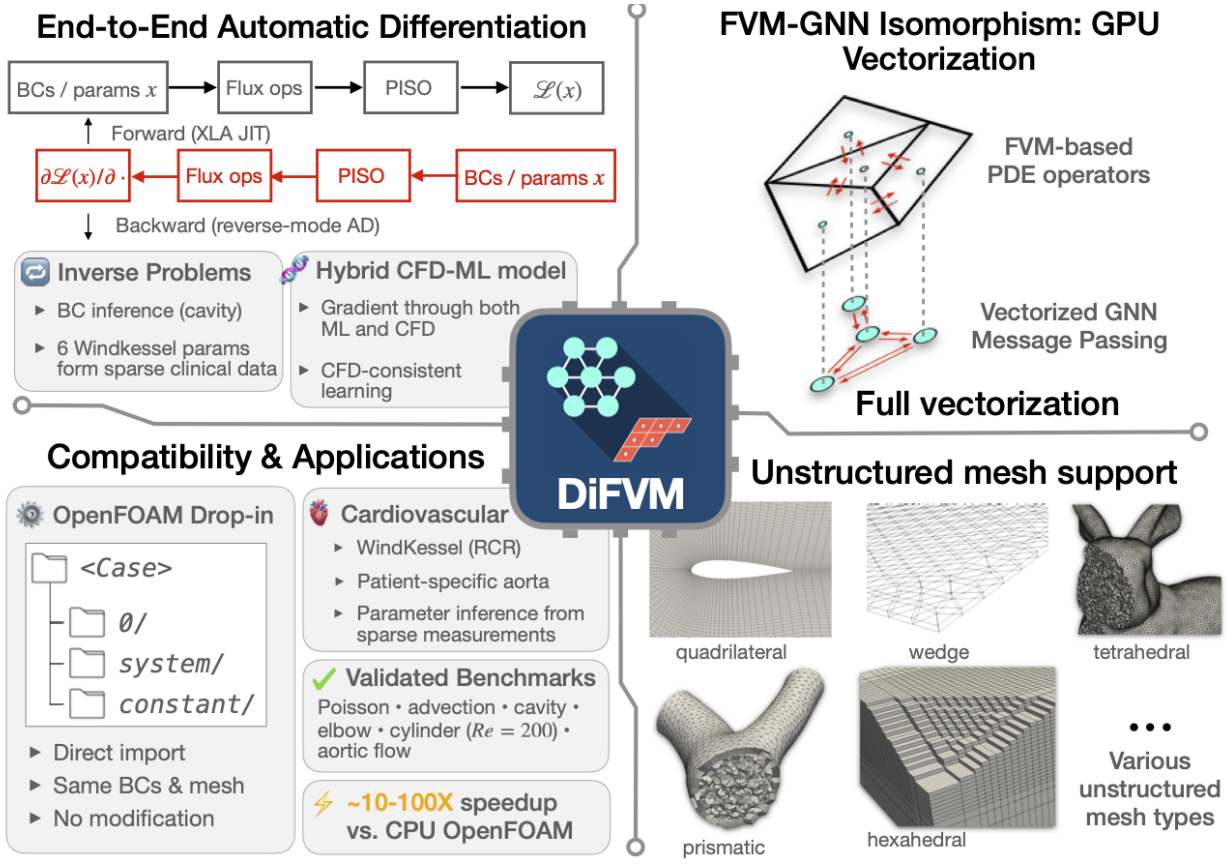

图1：DiFVM 概览：端到端可微分的基于图的有限体积法求解器，支持 GPU 向量化、原生非结构网格和与 OpenFOAM 即插即用兼容，可用于反问题和混合机器学习-计算流体力学建模。

其核心设计洞见在于，有限体积法离散与图神经网络的消息传递（MP）范式之间存在结构同构 [58]：根据单元中心的物理量计算单元间通

量，并在每个控制体上对其进行聚合，这恰好就是图神经网络框架通过散射/收集原语以完全向量化、GPU 原生的方式执行的 MP（消息构造与邻域聚合）模式。关键在于，这种映射精确无损：它是对经典有限体积法算子的重新表述，而非近似。通过将所有有限体积法算子表示为网格连通图上的静态基于图的 MP 原语，我们可以显式表示非结构网格邻接关系，并通过批量化的收集—计算—散射内核计算通量，从而尽管连通关系不规则，仍能实现高效的 GPU 向量化。JAX 的加速线性代数（XLA）编译器可在整个求解器流程中提供即时编译与算子融合；时间积分则通过 jax.lax.scan 展开，以消除 Python 端开销并实现内存高效的时间反向传播。

2.2. 控制方程

DiFVM 在有界区域 $\Omega \subset \mathbb{R}^3$ 及其边界 $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_W$上求解牛顿流体的非定常不可压缩纳维–斯托克斯方程：

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho}\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \mathbf{f}, \tag{2}$$

其中 $\mathbf{u}(\mathbf{x}\ t) \in \mathbb{R}^3$ 表示速度场，$p(\mathbf{x}\ t)$ 表示压力场，ν 表示运动黏度，$\mathbf{f}$ 表示体力，而 $\otimes$ 表示外积。连续性方程（1）施加无散约束，动量方程（2）则平衡惯性、压力梯度、黏性扩散和体力。该方程组通过初始条件和边界条件闭合：

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{3}$$

$$\mathbf{u} = \mathbf{u}_\Gamma(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_D \quad \text{(Dirichlet: inlet, wall, or moving surface)}, \tag{4}$$

$$-p\mathbf{n} + \nu(\nabla \mathbf{u})\mathbf{n} = \mathbf{h}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_N \quad \text{(Neumann: outlet or traction)}, \tag{5}$$

$$\mathbf{u} = \mathbf{0}, \quad \mathbf{x} \in \Gamma_W \quad \text{(no-slip wall)}, \tag{6}$$

其中$\mathbf{n}$ 表示 $\partial\Omega$上的外向单位法向量。对于心血管应用，诺伊曼出口条件 $\mathbf{h}$ 由第 2.6.3 节详述的温德凯塞尔模型给出，该模型通过集总参数血管阻力–顺应性回路，将出口压力与瞬时流量动态耦合起来。

2.3. 非结构网格上的有限体积离散

2.3.1. 网格数据结构与同位单元中心布局

计算域 Ω 被离散为一组互不重叠的多面体控制体（单元） $\{\Omega_i\}$, $i = 1,\ldots, N_c$，其中 N_c为单元总数。每个单元 Ω_i 由一组面 $\mathcal{F}(i) = \{f_1, f_2, \ldots\}$围成，其中每个内部面 f恰好由两个单元共享（所属单元 i 和相邻单元 j），而每个边界面关联一个单元，并指定相应的边界条件。对于每个面 f，有向面积矢量 $\mathbf{S}f$ 从所属单元指向外侧，其大小等于面面积 $A_f = |\mathbf{S}_f|$，方向与外向单位法向量 $\hat{\mathbf{n}}f$一致。单元 i的质心记为 $\mathbf{x}_i$，单元体积为 V_i，面质心为 $\mathbf{x}f$。

DiFVM 采用同位单元中心布局：速度 $\mathbf{u}_i$ 和压力p_i 均存储在单元质心处。这种布局能够自然适配任意多面体单元，并简化第 2.4 节所述的基于图的向量化策略。为防止同位网格上出现虚假的压力-速度解耦，在构造连续性方程所需的面速度时，会应用 Rhie–Chow 动量插值 [61]，详见第 2.4.2 节。初始化时会提取完整的网格拓扑（即单元质心、面面积向量、所属/相邻单元索引以及边界区域分配），并将其存储为静态、不可变数组。由于整个模拟过程中网格几何保持固定，因此生成的连通图构成静态数据结构，可预先计算一次、加载到 GPU 内存中，并在所有时间步中重复使用而无需修改。

2.3.2. 对流通量和扩散通量的空间离散

对动量方程进行积分（2）于单元Ω_i 并应用散度定理，可得半离散有限体积法平衡式：

$$V_i \frac{d\mathbf{u}_i}{dt} = - \sum_{f\in\mathcal{F}(i)} \underbrace{(\mathbf{u}\otimes\mathbf{u})_f}_{\mathbf{J}_f^C} \cdot \mathbf{S}_f + \sum_{f\in\mathcal{F}(i)} \underbrace{\nu\,(\nabla\mathbf{u})_f}_{\mathbf{J}_f^D} \cdot \mathbf{S}_f - \frac{1}{\rho} \sum_{f\in\mathcal{F}(i)} p_f\,\mathbf{S}_f + V_i\mathbf{f}, \tag{7}$$

其中，对流通量$\mathbf{J}^C = \mathbf{u}\otimes\mathbf{u}$ 和扩散通量$\mathbf{J}^D = \nu\nabla\mathbf{u}$ 是面质心处的二阶张量，$\mathbf{J}_f^C \cdot \mathbf{S}f$和$\mathbf{J}_f^D \cdot \mathbf{S}f$分别表示穿过面 f的净对流动量通量和净扩散动量通量。对流面通量可简化为$\mathbf{J}_f^C \cdot \mathbf{S}f = (\mathbf{u}f \cdot \mathbf{S}f)\,\mathbf{u}f$，这也是离散实现中采用的形式。面面积向量$\mathbf{S}f$ 指向Ω_i外部，且p_f，$\mathbf{u}f$ 分别表示面插值压力和速度。对流通量和扩散通量的离散处理存在根本差异，下面将分别详细说明。

对流通量离散。对流通量张量$\mathbf{J}^C = \mathbf{u} \otimes \mathbf{u}$是一个二阶张量，但它与面面积向量的点积会对每个速度分量分别化简为一个标量输运问题。具体而言，应用恒等式（$\mathbf{a} \otimes \mathbf{b}$）$\cdot\, \mathbf{n} =$（$\mathbf{b} \cdot \mathbf{n}$）$\mathbf{a}$:

$$\mathbf{J}_f^C \cdot \mathbf{S}_f = (\mathbf{u} \otimes \mathbf{u})_f \cdot \mathbf{S}_f = \dot{m}_f \, \mathbf{u}_f, \qquad \dot{m}_f = \mathbf{u}_f \cdot \mathbf{S}_f, \tag{8}$$

其中， $\dot{m}_f$ 是穿过面 f 的面法向体积通量。对于一般的输运标量 ϕ（代表任一速度分量），因此面 f 上的对流项为$\dot{m}_f \, \phi_f$，而计算 $\mathbf{J}_f^C \cdot \mathbf{S}f$ 的问题完全归结为：根据相邻两个单元（所有者单元和邻居单元）的单元中心值 ϕ_i 和 ϕ_j 重构面值 ϕ_f。

DiFVM 支持多种面重构格式。一阶迎风格式 (U) 根据 $\dot{m}_f$ 的符号选取迎风单元值:

$$\phi_f^U = \begin{cases} \phi_i & \dot{m}_f \geq 0 \quad (\text{flow from } i \to j), \\ \phi_j & \dot{m}_f < 0 \quad (\text{flow from } j \to i), \end{cases} \tag{9}$$

该格式有界且无条件稳定，但精度仅为一阶。为了在保持数值稳定性的同时获得高阶精度，DiFVM 采用延迟修正 策略:

$$\dot{m}_f \, \phi_f^{HO} = \underbrace{\dot{m}_f \, \phi_f^U}_{\text{implicit}} + \underbrace{\dot{m}_f \big(\phi_f^{HO} - \phi_f^U\big)}_{\text{explicit correction}}, \tag{10}$$

其中，上标 U 和 HO 分别表示一阶迎风格式和高阶重构。该方法在收敛时实现高阶精度，同时保持动量矩阵的对角占优性和稀疏结构。高阶面值 ϕ_f^{HO}通过二阶迎风格式 (SOU)计算，该格式利用迎风单元的单元中心值和梯度重构 ϕ_f：

$$\phi_f^{HO} = \phi_C + (\nabla\phi)_C \cdot \mathbf{d}_{Cf}, \qquad \mathbf{d}_{Cf} = \mathbf{x}_f - \mathbf{x}_C, \tag{11}$$

其中， C 表示迎风单元（根据 $\dot{m}_f$的符号，为所有者 i 或邻居 j ），而$(\nabla\phi)_C$表示通过高斯–格林定理计算的单元中心梯度:

$$(\nabla\phi)_i = \frac{1}{V_i} \sum_{f \in \mathcal{F}(i)} \phi_f \, \mathbf{S}_f, \tag{12}$$

其中 ϕ_f 通过线性插值获得。

扩散通量离散与非正交修正。 扩散通量张量 $\mathbf{J}^D = \nu\nabla\mathbf{u}$是一个二阶张量。然而，其面通量 $\mathbf{J}_f^D \cdot \mathbf{S}f$ 可分解为针对每个速度分量的独立标量问题。将 $(\nabla\mathbf{u})_{kl} = \partial u^k/\partial x^l$ 写出，并与 $\mathbf{S}f$进行缩并：

$$\left[\mathbf{J}_f^D \cdot \mathbf{S}_f\right]^k = \nu\,(\nabla u^k)_f \cdot \mathbf{S}_f, \tag{13}$$

因此，对于每个速度分量 $\phi \equiv u^k$，问题都归结为计算标量面通量$\nu\,(\nabla\phi)f \cdot \mathbf{S}f$。从结构上看，这与对流情形类似，但此处的难点不在于面值重构，而在于准确计算面法向梯度 $(\nabla\phi)f \cdot \mathbf{S}f$。

在正交网格上（$\mathbf{S}_f \parallel \mathbf{d}ij$），面法向梯度可简化为紧凑的两点模板：

$$\nu\,(\nabla\phi)_f \cdot \mathbf{S}_f \approx \nu\,|\mathbf{S}_f|\,\frac{\phi_j - \phi_i}{|\mathbf{d}_{ij}|}, \tag{14}$$

该方法具有二阶精度。在一般非结构网格上，$\mathbf{S}f$ 与$\mathbf{d}ij$ 不平行，因此公式（14）会产生一阶截断误差。为恢复二阶精度，$\mathbf{S}f$ 被分解为一个正交分量 $\mathbf{\Delta}_f$和一个非正交修正项$\mathbf{k}f = \mathbf{S}_f - \mathbf{\Delta}_f$:

$$\nu\,(\nabla\phi)_f \cdot \mathbf{S}_f = \underbrace{\nu\,|\mathbf{\Delta}_f|\,\frac{\phi_j - \phi_i}{|\mathbf{d}_{ij}|}}_{\text{orthogonal}} + \underbrace{\nu\,(\nabla\phi)_f \cdot \mathbf{k}_f}_{\text{non-orthogonal correction}}, \tag{15}$$

其中， $\mathbf{\Delta}_f$ 的选择决定了修正方案 [62]。令 $\mathrm{d}ij = \mathrm{d}_{ij}/|\mathrm{d}_{ij}|$ 和 $\mathrm{S}f = \mathrm{S}_f/|\mathrm{S}_f|$分别为沿单元间方向和面法向的单位向量，最小修正法取 $\mathbf{\Delta}_f = (\mathrm{S}f \cdot \mathrm{d}ij)\mathrm{d}ij$，使 $|\mathrm{k}_f|$ 最小，但在高度非正交网格上的表现不佳；正交修正法取 $\mathbf{\Delta}_f = |\mathrm{S}_f|\mathrm{d}ij$，在偏斜网格上可能会高估修正量；DiFVM采用的过松弛法则取$\mathbf{\Delta}_f = \left(|\mathrm{S}_f|/|\mathrm{S}f \cdot \mathrm{d}_{ij}|\right)\mathrm{d}ij$，对于复杂工程几何中常见的中度非正交网格，该方法在精度与稳定性之间实现了最佳平衡 [62]。修正项中所需的单元中心梯度 $(\nabla\phi)_i$ 通过高斯–格林定理计算。

2.4. 面向 *GPU* 向量化的有限体积法求解器基于图的重构

2.4.1. 有限体积法与图神经网络之间的结构同构及静态图构建

在非结构网格上实现 GPU 向量化的关键，在于有限体积法离散与图神经网络消息传递（MP）范式之间存在结构同构，如图2所示。如图所示，可将网格表示为有向图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$，其中每个节点

$v_i \in \mathcal{V}$ 对应一个存储状态的单元质心（$\mathbf{u}_i, p_i$），每条有向边 $e_{ji} \in \mathcal{E}$ 对应一个内部面，携带从相邻单元 j 传递至所属单元 i的信息。边界面则视为从虚节点指向内部单元的有向边。每条边都带有静态几何属性 $\mathbf{g}^{ji} = (\mathbf{S}f, \mathbf{x}f)$，这些属性在初始化时预先计算，并在整个模拟过程中保持不变。

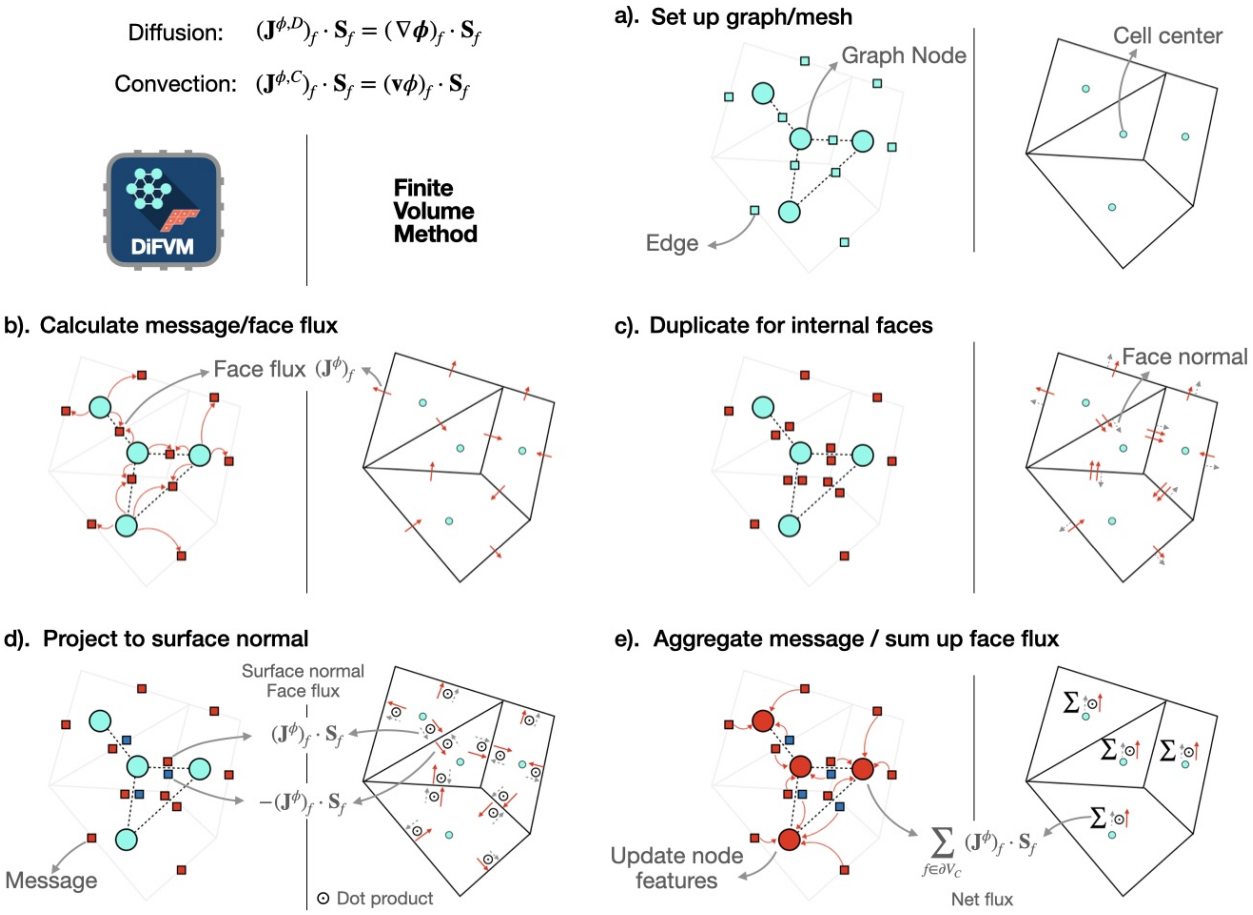

图 2：非结构网格上的图消息传递与有限体积离散之间的类比。单元中心和共享面分别映射为图节点和有向边。面通量计算和聚合分别与图神经网络的消息构建和节点更新步骤完全对应，从而能够通过散射/收集原语实现全面的 GPU 向量化。

初始化时，DiFVM 根据网格构建以下静态数组，并将其上传到 GPU 内存一次，在整个模拟过程中重复使用：owner、neighbor $\in \mathbb{Z}^{|\mathcal{E}|}$（所有者/相邻单元索引）；Sf、xf $\in \mathbb{R}^{|\mathcal{E}|\times 3}$（面面积向量、面质心）；cf、vol $\in \mathbb{R}^{N_c}$（单元中心、单元体积）；以及边界片区索引数组，用于将每个边界面映射到其条件类型和参数。动态状态仅由两个节点级数组组成：U $\in \mathbb{R}^{N_c \times 3}$ 和 p $\in \mathbb{R}^{N_c}$。由于所有

几何数组都是静态的，XLA 编译器会将其视为编译时常量，并将对它们的访问融合到经过优化的 GPU 内核中。

在此图表示下，有限体积法残差（7）恰好映射为三阶段的消息传递操作（$\mathcal{M}$、 $\mathcal{A}$和 $\mathcal{U}$ 分别为消息构造、聚合和更新算子）：

1. **消息构造**。 对于每条有向边 e_{ji}，根据两个端点单元的状态和边几何属性，计算面通量消息 $\mathbf{m}_{ji}$：

$$\mathbf{m}_{ji} = \mathcal{M}(\mathbf{u}_i, \mathbf{u}_j, \mathbf{g}_{ji}) = \underbrace{-\mathbf{J}_f^C \cdot \mathbf{S}_f}_{\text{convective}} + \underbrace{\mathbf{J}_f^D \cdot \mathbf{S}_f}_{\text{diffusive}} - \underbrace{\frac{1}{\rho} p_f \, \mathbf{S}_f}_{\text{pressure}}, \tag{16}$$

其中，$\mathbf{J}_f^C$ 和$\mathbf{J}_f^D$ 分别表示在面 f处计算得到的对流通量张量和扩散通量张量。每条消息都是节点和边属性的逐点函数，不依赖全局网格拓扑，因此可以在所有边上进行高度并行的计算。

2. **通量聚合**。 对于每个节点 v_i，聚合所有入边消息 ：

$$\mathbf{R}_i = \mathcal{A}(\mathbf{m}_{ji}, \mathcal{E}) = \sum_{j \in \mathcal{N}(i)} \sigma_{ij} \, \mathbf{m}_{ji}, \tag{17}$$

其中， $\mathcal{N}(i)$表示与单元 i 共享一个面的单元集合， $\sigma_{ij} = \pm 1$ 表示外向定向约定。该聚合通过 jax.ops.segment_sum 实现：根据预先计算的所有者数组，对逐边通量数组进行索引；这相当于对所有 N_c 个节点执行一次并行散射加法。

3. **节点更新**。 使用聚合残差更新单元状态：

$$\mathbf{u}_i^{n+1} = \mathcal{U}(\mathbf{u}_i^n, \frac{\mathbf{R}_i}{V_i}) \tag{18}$$

其中 V_i 是单元体积， $\mathcal{U}$ 表示时间积分算子（例如，前向欧拉，$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta t \mathbf{R}_i / V_i$）。对于不可压缩流， $\mathcal{U}$ 必须满足无散度约束 $\nabla \cdot \mathbf{u} = 0$，这无法通过简单的显式步实现。DiFVM 将 $\mathcal{U}$ 实现为满足无散度约束的消息传递过程，详见第 2.4.2节。

非规则网格连接关系在传统 CPU 求解器中需要按单元顺序遍历，而在这里则完全编码在预计算的边索引数组中，这些数组作为批量张量运算的静态聚集/散射索引。由此，残差计算得以完全向量化，并可简化为两个 GPU 内核：一个对所有 $|\mathcal{E}|$ 边执行并行收集计算（MP），另一个对所有 N_c 节点执行并行散射加法（通量聚合），且无需在 Python 层面循环遍历网格元素。

2.4.2. 将 *PISO* 视为无散度约束的消息传递

朴素的显式节点更新（18）无法在同位网格上保证 $\nabla\cdot\mathbf{u}=0$，因为速度和压力并非独立确定。压力隐式算子分裂法（PISO）算法 [63]通过将 $\mathcal{U}$ 分解为三个图操作来解决这一问题，这些操作共同将速度投影到无散子空间。每个操作都仅通过 $\mathcal{G}$上的散射/收集原语来实现，不涉及图抽象之外的任何操作。

第 *1* 层：动量*MP*层。计算预测速度$\mathbf{u}^*$，方法是从时间 t^n推进动量残差（7），并显式处理 p^n的压力梯度：

$$\frac{\mathbf{u}_i^*-\mathbf{u}_i^n}{\Delta t}=\frac{1}{V_i}\big[\mathbf{R}_i^{\mathrm{conv}}(\mathbf{u}^n)+\mathbf{R}_i^{\mathrm{diff}}(\mathbf{u}^n)+\mathbf{R}_i^{\mathrm{pres}}(p^n)\big], \tag{19}$$

其中$\mathbf{R}^{\mathrm{conv}}$、$\mathbf{R}^{\mathrm{diff}}$、$\mathbf{R}^{\mathrm{pres}}$ 通过（16）–（17）计算。这是一个标准的MP残差层：可在所有边上轻松并行计算。由此得到的$\mathbf{u}^*$ 通常不满足连续性。

第 *2* 层：图拉普拉斯投影层。为恢复 $\nabla\cdot\mathbf{u}=0$，对方程（7）进行隐式离散化并组装后，可得到一个线性系统，其对角系数 a_i^P源自时间导数项和隐式扩散项，而其非对角算子 $H_i(\mathbf{u})$ 汇集所有邻居贡献：

$$a_i^P\,\mathbf{u}_i=H_i(\mathbf{u}^*)-\frac{1}{\rho}\sum_{f\in\mathcal{F}(i)}p_f^{n+1}\,\mathbf{S}_f. \tag{20}$$

代入离散连续性约束 $\sum_{f\in\mathcal{F}(i)}\mathbf{u}_f^{n+1}\cdot\mathbf{S}_f=0$，并对面速度应用 Rhie–Chow 插值，得到：

$$\underbrace{\sum_{f\in\mathcal{F}(i)}\frac{1}{a_f^P}(\nabla p^{n+1})_f\cdot\mathbf{S}_f}_{\text{weighted graph Laplacian on }\mathcal{G}}=\sum_{f\in\mathcal{F}(i)}\left(\frac{H(\mathbf{u}^*)}{a^P}\right)_f\cdot\mathbf{S}_f. \tag{21}$$

左侧恰好是定义在 $\mathcal{G}$ 上的加权图拉普拉斯，其边权重为 $1/a_f^P$，并通过相同的所有者/邻居数组进行散射/聚集来组装。右侧是对经过面插值的 H/a^P 场进行标准 MP 聚合。最终得到的对称正定线性系统采用共轭梯度法并结合不完全LU预条件进行迭代求解。

第 *3* 层：校正*MP*层。一旦获得 p^{n+1}，即可通过最后一次散射操作校正速度场：

$$\mathbf{u}_i^{n+1}=\frac{H_i(\mathbf{u}^*)}{a_i^P}-\frac{1}{\rho\,a_i^P}\sum_{f\in\mathcal{F}(i)}p_f^{n+1}\,\mathbf{S}_f, \tag{22}$$

并通过 Rhie–Chow 插值校正面法向通量，以确保每个面上的离散质量守恒。第 2–3 层可以重复执行（通常两次），以减小压力–速度分裂误差。这一完整的三层流程构成了 $\mathcal{U}$ 方程（18）中的内容：在 $\mathcal{G}$ 上完全通过图操作实现的无散约束更新。

2.5. 端到端自动微分

2.5.1. 通过求解器进行反向模式微分

DiFVM 的端到端梯度计算通过 JAX 的反向模式（向量－雅可比积，VJP）自动微分实现，并应用于完整的仿真流程。给定一个标量损失函数 $\mathcal{L}(\mathbf{u}(T;\,\boldsymbol{\theta}))$，它通过参数$\boldsymbol{\theta}$ 依赖于时间 T的模拟状态（参数可能包括边界条件值、物理参数、模型系数或初始条件），梯度 $\partial\mathcal{L}/\partial\boldsymbol{\theta}$ 通过 jax.vjp 对模拟函数进行计算：

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbf{v}^{\top} \frac{\partial \mathbf{u}(T;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{23}$$

其中，$\mathbf{v} = \partial\mathcal{L}/\partial\mathbf{u}(T)$ 是伴随向量（损失对最终状态的敏感度）。JAX 的自动微分引擎会通过 jax.lax.scan 循环以及每个基于图的算子向后传播该伴随向量，包括对流通量、扩散通量、压力求解、速度校正和边界条件施加，无需手动推导伴随方程。由于所有操作都表示为可微的 JAX 原语（算术运算、聚集、散射、线性求解），VJP 规则会自动组合，从而得到机器精度的精确梯度。

2.5.2. 梯度检查点与隐式微分

通过 n_{steps}步时间步进进行朴素自动微分需要存储所有中间激活值，从而产生 $\mathcal{O}(n_{\text{steps}})$ 的内存开销。对于在细网格上进行的长时间模拟，这可能会超出 GPU 内存容量。DiFVM 通过梯度检查点解决这一问题：在反向传播期间有选择地重新生成中间状态，而不是将其全部存储：

$$\text{Memory} \sim \mathcal{O}(\sqrt{n_{\text{steps}}} \cdot N_c), \qquad \text{Compute} \sim \mathcal{O}(n_{\text{steps}} \cdot N_c), \tag{24}$$

在 jax.lax.scan 循环中应用最优二项式检查点调度 [64] 。在每次 PISO 内部迭代中，还会通过压力求解额外应用隐式微分

泊松线性求解，以避免展开迭代求解器（这需要存储所有共轭梯度迭代结果）。具体而言，对于已收敛的压力求解 $\mathcal{F}(p^*,\ \mathbf{u}^*) = 0$，$p^*$ 关于$\mathbf{u}^*$ 的伴随量通过隐函数定理计算 [40]:

$$\frac{\partial p^*}{\partial \mathbf{u}^*} = -\left(\frac{\partial \mathcal{F}}{\partial p^*}\right)^{-1} \frac{\partial \mathcal{F}}{\partial \mathbf{u}^*}, \tag{25}$$

其中，伴随线性系统 $(\partial \mathcal{F}/\partial p^*)^\top \boldsymbol{\lambda} = \partial \mathcal{L}/\partial p^*$ 使用与前向传播相同的迭代求解器求解。这使得压力求解的梯度能够精确计算，且内存和计算开销相当于单次前向求解，与线性求解器的迭代次数无关。

2.6. 支持的边界条件、数值格式和接口

2.6.1. 数值格式和线性求解器

DiFVM 支持一套全面的数值格式，用于时间积分、对流离散和扩散通量处理，汇总见表1。时间积分方面，可采用一阶显式和后向欧拉格式，以及二阶克兰克–尼科尔森格式；为保证细网格上的稳定性，建议使用隐式格式。对流格式从一阶迎风格式到高阶格式（SOU、QUICK）不等，均在方程（10）的延迟修正框架内实现，以保持动量矩阵的对角占优。扩散通量处理支持第 2.3.2 节所述的非正交修正策略：不修正（仅适用于正交网格）、最小修正、正交修正，以及默认采用的过松弛修正。压力–速度耦合仅通过 PISO算法处理。压力泊松线性系统和动量线性系统采用共轭梯度法（CG）、BiCGStab 或 GMRES 求解，且均使用不完全LU预条件；对于对称压力系统，默认采用共轭梯度法；对于非对称动量系统，默认采用 BiCGStab。

2.6.2. 标准边界条件

DiFVM 支持狄利克雷速度边界条件（定值入口、无滑移壁面）、诺伊曼压力边界条件（零梯度出口、自由滑移壁面）、用于二维模拟的空边界区，以及时变入口流型。所有边界条件均在与内部通量计算相同的散射框架中，通过对边界边界区执行图操作来实现，并且可通过第2.5节所述的向量–雅可比积流水线，对其参数求导。

表1：DiFVM 支持的数值模型、离散格式和功能。

类别	支持的选项
Mesh	非结构多面体、任意单元类型
时间积分	前向欧拉（一阶）、后向欧拉（一阶隐式）、克兰克–尼科尔森（二阶）
压力-速度	PISO
对流	迎风格式（1 阶）、中心格式（2 阶）、二阶迎风格式（延迟修正）、QUICK格式（延迟修正）
扩散通量	无修正、最小修正、正交修正、过松弛（默认）
线性求解器	CG、BiCGStab、GMRES（均采用不完全LU预条件）
边界条件	狄利克雷、诺伊曼、空（2D）、时变入口流、温德凯塞尔RCR
接口	OpenFOAM 即插即用（兼容 blockMesh/snappyHexMesh）、VTK非结构网格

2.6.3. 心血管应用的温德凯塞尔*RCR*出口条件

对于心血管模拟，DiFVM 在出口边界实现了三元件温德凯塞尔（RCR）模型，用于表示计算域中未明确解析的下游血管系统的集总血流动力学阻力和顺应性。三个元件分别表示近端动脉阻力 R_p、下游动脉系统的弹性储能能力 C，以及小动脉和毛细血管的外周血管阻力 R_d。通过边界面 Γ_o 的出口流量根据离散面通量计算：

$$Q_o(t) = \sum_{f \in \Gamma_o} \dot{m}_f, \qquad \dot{m}_f = \mathbf{u}_f \cdot \mathbf{S}_f, \tag{26}$$

采用与内部通量汇总相同的散射操作。设 $p_o(t)$ 为面积平均出口压力。RCR 模型引入内部电容压力状态 $p_c(t)$，其由下式控制：

$$C\frac{dp_c}{dt} + \frac{1}{R_d}p_c = Q_o(t), \tag{27a}$$

$$p_o(t) = p_c(t) + R_p Q_o(t). \tag{27b}$$

其中，方程（27a）规定了顺应性元件与远端阻力之间的流量分配，而方程（27b）加入了近端阻力导致的压降。由于方程（27a）是一个具有常系数的线性常微分方程，假设 Q_o 在 $[t^n, t^{n+1}]$ 内分段恒定，可得到精确更新：

$$p_c^{n+1} = p_c^n \exp\left(-\frac{\Delta t}{R_d C}\right) + R_d Q_o^{n+1}\left(1 - \exp\left(-\frac{\Delta t}{R_d C}\right)\right), \tag{28}$$

并将 $p_o^{n+1} = p_c^{n+1} + R_p Q_o^{n+1}$作为 PISO 压力求解中的出口压力边界条件。DiFVM 还支持对 Windkessel 常微分方程采用显式前向欧拉和隐式后向欧拉时间

离散格式，可通过求解器配置进行选择。对于多个出口边界片 $\{\Gamma_{o,j}\}_{j=1}^{N_o}$，每个出口都会分配一个独立的 RCR 模型（$R_{p,\ j}$，$C_j$，$R_{d,\ j}$），并行更新。

关键在于，所有参数 (R_p、C、R_d) 都作为 JAX 计算图中的叶节点公开。由于方程（28）完全由可微分的 JAX 操作实现，且 Q_o^{n+1}是速度状态的可微分函数，因此梯度 $\partial\mathcal{L}/\partial(R_p$、$C$、$R_d)$ 会通过第 2.5节所述的同一向量-雅可比积过程自动计算，从而能够根据稀疏的临床观测，同时对所有出口处的 Windkessel 参数进行端到端推断。

2.6.4. OpenFOAM 和 *VTK* 接口

DiFVM 提供两个层级的外部格式支持。对于 OpenFOAM，DiFVM 在算例层面可作为即插即用的替代方案：它无需任何修改即可直接读取标准 OpenFOAM 算例目录，从 `constant/polyMesh/`(`points`、`faces`、`owner`、`neighbour`、`boundary`) 中获取网格拓扑，从 `0/U` 和 `0/p`中获取初始条件和边界条件，从 `constant/transportProperties`中获取物理参数，并从 `controlDict`,`fvSchemes`, `fvSolution`中获取数值设置。因此，用户可以完全在 OpenFOAM 生态系统内构建和配置模拟，使用 `blockMesh`、`snappyHexMesh`或任何可导出 OpenFOAM 格式的第三方网格生成器，然后将生成的算例目录直接传递给 DiFVM，而无需重新指定网格、边界条件或物理/数值参数。这确保跨求解器验证严格在完全相同的问题设置下进行，消除了网格离散和边界条件指定在精度比较中作为混淆变量的影响。对于 OpenFOAM 生态系统之外的一般非结构网格，DiFVM 还支持导入和导出 VTK 非结构网格格式，从而实现与第三方网格生成器以及前后处理流程的互操作。我们还为 SimVascular 网格提供了接口，使 SimVascular 的心血管网格划分工作流能够与 DiFVM 求解器无缝集成。在所有情况下，导入的网格拓扑都会转换为相同的静态图数组（见第 2.4.1节），求解器核心不会因此进行任何特定于格式的修改。

3. 数值结果

本节通过一系列分为两部分的基准问题，评估 DiFVM 的准确性、稳健性和可微性。第一部分（第 3.1 节）验证正向

模拟的准确性，涵盖六个几何和物理复杂度逐步增加的测试案例，从三维泊松问题和二维典型流动，到非定常圆柱绕流涡脱落以及三维脉动患者特异性主动脉血流动力学。在所有案例中，DiFVM 均在完全相同的网格、边界条件和数值设置下，与 OpenFOAM 进行定量比较，并根据解析解或高保真参考数据评估误差。第二部分（第 3.2 节）通过两个复杂度逐步增加的反问题展示端到端可微性：从方腔流中的稀疏内部速度测量值恢复单个标量边界参数，以及根据脉动分叉几何中的入口压力和出口流量观测，同时推断六个 Windkessel 参数。总体而言，这些基准测试证明，DiFVM 既是适用于非结构网格的高保真 GPU加速正向求解器，也是面向偏微分方程约束逆问题和数据驱动混合计算流体力学与机器学习建模的梯度可计算平台。

3.1. 正向模拟验证

3.1.1. 三维泊松方程

我们首先使用一个具有给定源项和狄利克雷边界条件的三维泊松问题对 DiFVM 进行验证。计算域为单位立方体 $\Omega = [0,1] \times [0,1] \times [0,1]$，采用非结构化四面体网格进行离散。控制方程为

$$\nabla \cdot (\nabla \phi) = f \quad \text{in } \Omega, \tag{29}$$

在全部六个面上施加狄利克雷边界条件，

$$\phi = 10 \quad \text{on } \partial\Omega. \tag{30}$$

体源项 f 以解析形式给定为

$$f(x,y,z) = -10\pi^2 \sin(\pi x)\,(y^4 - y)\sin(2\pi z) + 24\,\sin(\pi x)\,y^2\,\sin(2\pi z), \tag{31}$$

其对应的精确解为

$$\phi_{\mathrm{GT}}(x,y,z) = 2\sin(\pi x)\,(y^4 - y)\,\sin(2\pi z) + 10. \tag{32}$$

根据构造，ϕ_{GT} 精确满足方程（29）和边界条件（30），因此可作为验证用的解析真实值。

泊松方程使用 DiFVM 求解，扩散算子采用二阶精度的有限体积离散，其中表面法向梯度通过结合过松弛非正交修正的中心差分格式计算。数值解与 ϕ_{GT} 进行比较，以评估逐点误差分布和全局误差范数。图3 展示了

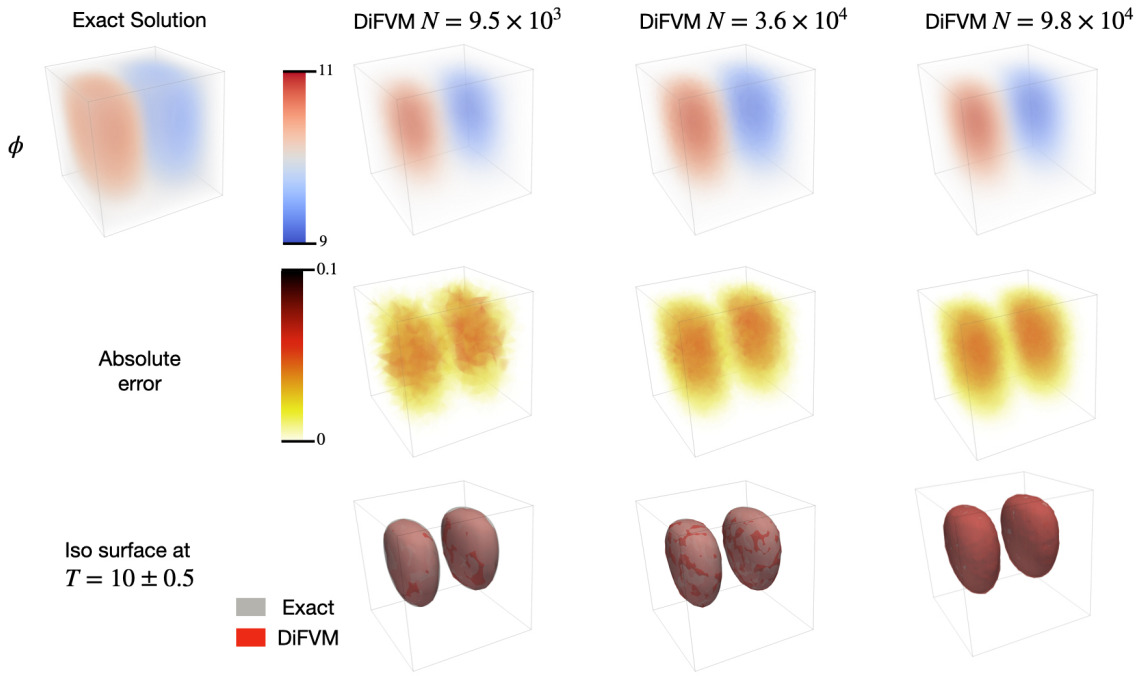

图3：三维泊松基准测试。随着网格分辨率提高（从左至右为 $N = 9.5\times10^3$、3.6×10^4、9.8×10^4 个单元），解析解与 DiFVM 结果的比较。顶行：标量场 ϕ。中行：绝对误差分布。底行：$\phi = 10 \pm 0.5$ 处的等值面（灰色：精确解；红色：DiFVM）。网格加密使误差幅值单调减小、等值面对齐效果改善，表明该方法在非结构网格上具有二阶空间精度。

在不断提高网格分辨率（$N = 9.5 \times 10^3$, 3.6×10^4, 9.8×10^4 个单元）时，精确解与 DiFVM 结果的定性对比。随着网格加密，DiFVM 解逐渐收敛于解析解，在标量场分布（上排）方面的一致性提高，绝对误差大小（中排）减小，且在 $\phi = 10 \pm 0.5$ 处的等值面（下排）对齐效果更佳。随着网格加密，绝对误差单调减小，证实了二阶空间离散预期的收敛特性。残余误差主要集中在强曲率和大梯度区域，尤其是沿着两个椭圆形核心结构相关极值附近

z方向。这种现象符合截断误差在二阶导数幅值较大的区域被放大的规律，而随着网格分辨率提高，这种放大效应会减弱。结果验证了 DiFVM 中的扩散离散，并展示了非结构网格上的网格收敛性。

3.1.2. 二维被动标量平流

我们考虑一个被动标量的二维稳态对流–扩散问题。计算域为单位正方形 $\Omega = [0,1]\times[0,1]$，并采用非结构三角形网格进行离散，如图 4(a) 所示。入口边界处设置阶跃分布，其中 $\phi = 1$ 位于入口下部线段（∂V_1），而 $\phi = 0$ 位于入口上部线段（∂V_2）。出口（∂V_3）施加齐次诺伊曼边界条件。使用 DiFVM 获得的数值解与采用相同数值设置、通过 OpenFOAM 的 `scalarTransportFoam` 求解器计算得到的参考解进行验证。被动标量 ϕ 的控制方程为

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) - \nabla \cdot (\Gamma \nabla \phi) = f, \tag{33}$$

其中 $\mathbf{u} = (2,\ 1,\ 0)$ 是恒定的斜向速度场，$\Gamma = 0.001$ 是扩散系数。这些参数对应的佩克莱数为 $\mathrm{Pe} \approx 2236$，表明输运状态由对流强烈主导。DiFVM 和 OpenFOAM 均采用后向欧拉时间离散、高斯线性梯度与插值格式，并对扩散算子使用过松弛修正的面法向梯度。对于对流项，分别采用一阶迎风格式和通过延迟修正实现的 SOU格式计算结果。

图 4(b) 对比了由 DiFVM 和 OpenFOAM 计算得到的稳态标量场 ϕ 及其对应的绝对误差分布。DiFVM 准确捕捉了入口阶跃分布的斜向平流与扩散，与 OpenFOAM 的参考解高度吻合。整个计算域内的绝对误差均低于 10^{-5}，证实了这两种求解器在数值上的等效性。图 4(c) 和 (d) 分别展示了沿竖直线$x = 0.3$ 和水平线 $y = 0.3$提取的标量剖面。采用迎风格式时，DiFVM 和 OpenFOAM 得到的剖面几乎完全一致，误差幅值低于 1×10^{-5}。观察到的残差差异归因于浮点初始化差异所引起的轻微数值扰动，这些扰动随后因线性求解器后端的差异而被放大。相比之下，SOU格式产生了更高的峰值和更陡峭的阶跃过渡，表明其数值扩散更小，且对陡峭梯度的分辨能力更强。总体而言，这项基准测试

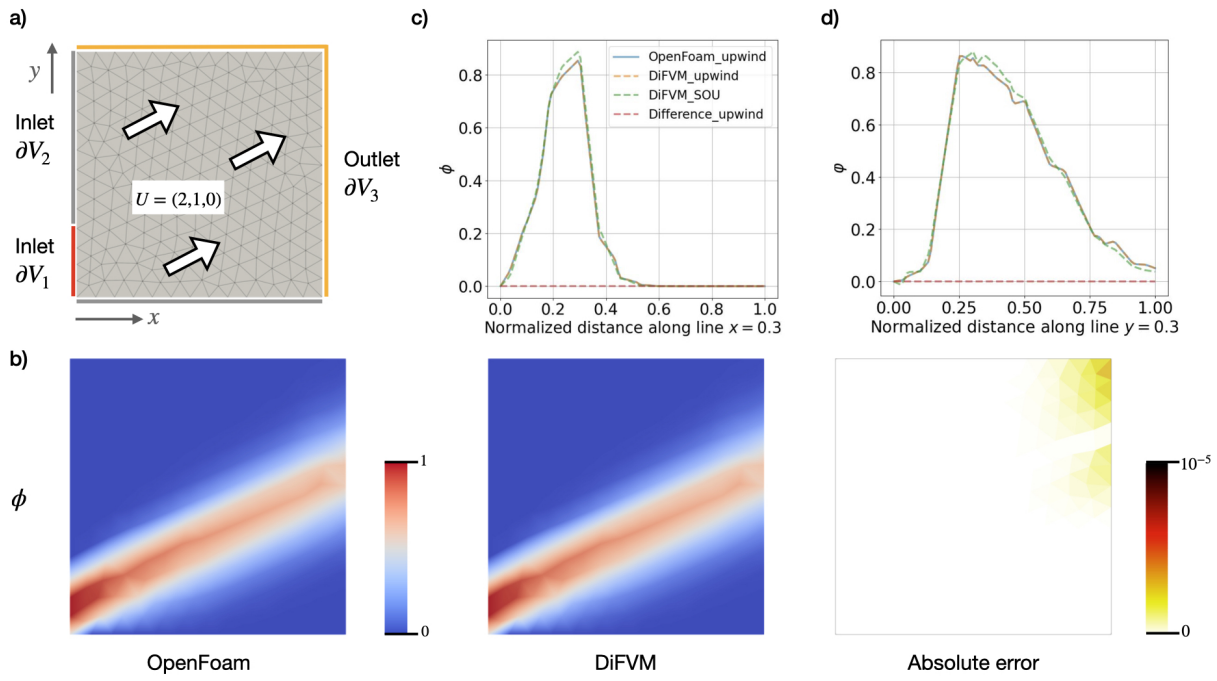

图 4：阶跃平流基准测试：(a) 计算域和边界条件；(b) OpenFOAM 和 DiFVM 计算得到的标量场 ϕ及其绝对误差分布；(c–d) 分别沿 $x = 0.3$ 和$y = 0.3$的标量剖面，对比迎风格式和 SOU格式。

验证了 DiFVM 在高佩克莱数工况下的对流离散能力 .

3.1.3. 二维弯管流

接下来，我们考虑流经 $90°$ 弯头的二维不可压缩流。这是一个经典基准案例，常用于评估曲线几何中压力–速度耦合、对流离散和扩散精度的稳健性。计算域和边界配置如图5(a)所示。计算域采用非结构三角形网格离散，壁面上施加无滑移边界条件（∂V_3）。在下方水平支路（∂V_1）和竖直支路（∂V_2）分别设置两个速度入口，入口速度均匀，分别为$\mathbf{u}_1 = (1,\ 0,\ 0)$和$\mathbf{u}_2 = (0,\ 3,\ 0)$。出口（$\partial V_4$）处对压力施加齐次诺伊曼边界条件。控制方程为不可压缩纳维–斯托克斯方程（1）–（2），采用后向欧拉时间离散求解。DiFVM 和 OpenFOAM 使用完全相同的数值设置，包括高斯线性梯度和插值格式、对流项的迎风离散，以及用于扩散的过松弛修正表面法向梯度，

并采用带有 Rhie–Chow 通量修正的 PISO 压力–速度耦合。速度和压力线性系统分别使用预条件共轭梯度法（PCG）和预条件双共轭梯度稳定法（PBiCGStab）求解器求解。

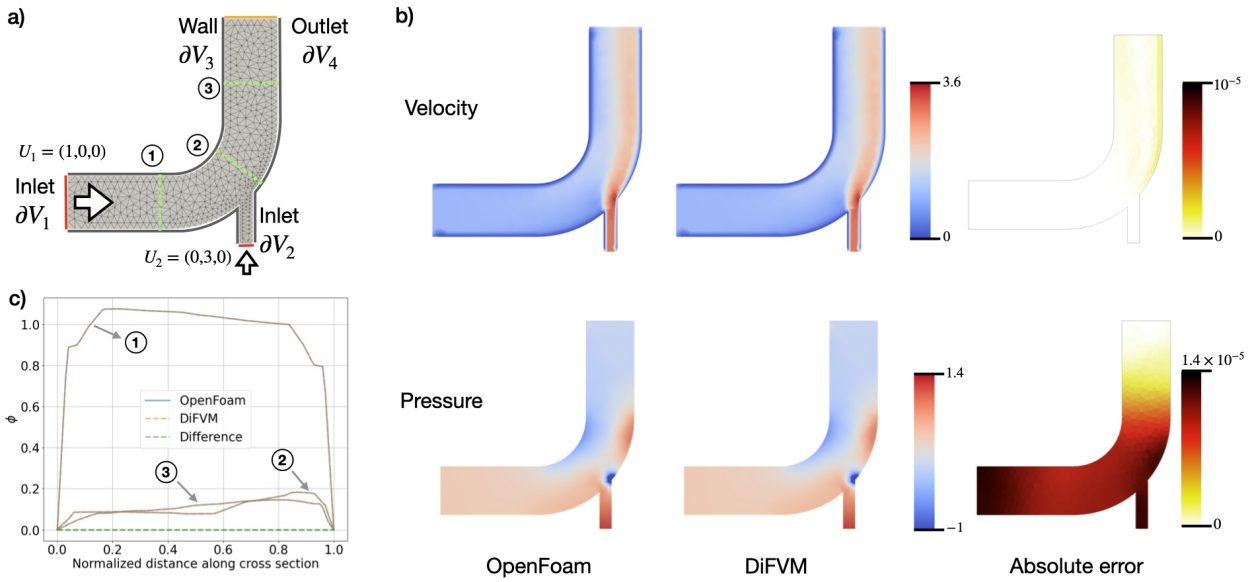

图5：二维弯管几何中稳态不可压缩流的基准验证。(a) 计算域和边界条件，包括指定的入口速度和出口压力条件。(b) 使用 OpenFOAM 和 DiFVM 得到的稳态速度大小（上排）和压力（下排）场的对比，以及相应的绝对误差分布。(c) 沿(a)中标示的代表性横截面提取的速度剖面，显示 DiFVM 与 OpenFOAM 的结果高度吻合。

图5（b）对比了使用 DiFVM 和 OpenFOAM 得到的稳态速度大小和压力场，以及相应的绝对误差分布。DiFVM 准确再现了流体流经弯头弯曲段时速度的特征性加速与重新分布，以及弯道外壁沿线的压力积聚。整个计算域内的绝对误差始终接近机器精度。为进一步量化两者的一致性，图5（c）展示了沿三个代表性横截面提取的速度剖面。在所有横截面上，DiFVM 的预测结果均与 OpenFOAM 参考解高度吻合，既捕捉到了核心流动行为，也捕捉到了近壁速度梯度。由于 DiFVM 和 OpenFOAM 遵循相同的数学公式和 PISO 压力-速度耦合过程，剩余差异主要源于计算初期浮点精度的差异，随后又受到线性求解器及底层代数运算差异的放大。这些影响仍可忽略不计，误差始终低于 1×10^{-5}。总体而言，该基准测试验证了

DiFVM 在不规则几何上的压力–速度耦合具有稳健性和准确性。

3.1.4. 二维顶盖驱动方腔流

接下来，我们考虑经典的二维顶盖驱动方腔流。这是一项典型的基准测试，其特征是会形成流动涡旋，并存在强烈的压力–速度耦合。计算区域为单位正方形$\Omega = [0,1] \times [0,1]$，并采用非结构三角形网格进行离散，如图6(a)所示。流动由顶部边界处的运动顶盖驱动，该处施加均匀的水平速度$\mathbf{u} = (2,0)$，其余壁面均满足无滑移边界条件。采用后向欧拉时间离散和PISO算法进行压力–速度耦合，以求解控制方程——不可压缩纳维–斯托克斯方程（1）–（2）。

图6(b)对比了DiFVM得到的稳态速度大小和压力场，与在相同数值设置下使用OpenFOAM计算得到的参考解。DiFVM准确捕捉了运动顶盖诱导的主回流涡，以及相应的压力分布，包括涡中心附近的低压核心和方腔角部附近的压力升高。绝对误差场证实，DiFVM与OpenFOAM之间的差异很小。图6(c)和(d)分别展示了沿空腔垂直中心线和水平中心线提取的速度剖面。对于两个速度分量，DiFVM均与OpenFOAM的结果高度吻合，并准确再现了主涡结构对应的预期极值、拐点和符号变化。这项基准测试验证了DiFVM在涡主导流动中的准确性。

3.1.5. 二维圆柱绕流

接下来，我们考虑经典的二维不可压缩流绕圆柱流动问题，其雷诺数为 $Re = 200$。该基准测试呈现非定常涡脱落现象，可对非定常纳维–斯托克斯求解器进行严格验证。计算域由一个直径为 $D = 0.2$ 的圆柱体和一个矩形通道组成。入口处施加均匀来流速度$U_\infty = 1$，圆柱表面施加无滑移边界条件，流出边界处采用零梯度出口条件。采用后向欧拉时间积分求解非定常不可压缩纳维–斯托克斯方程（1）–（2），并使用 PISO 算法进行压力–速度耦合。对流项采用带延迟修正的迎风格式离散，扩散项则采用超松弛修正表面法向梯度格式离散

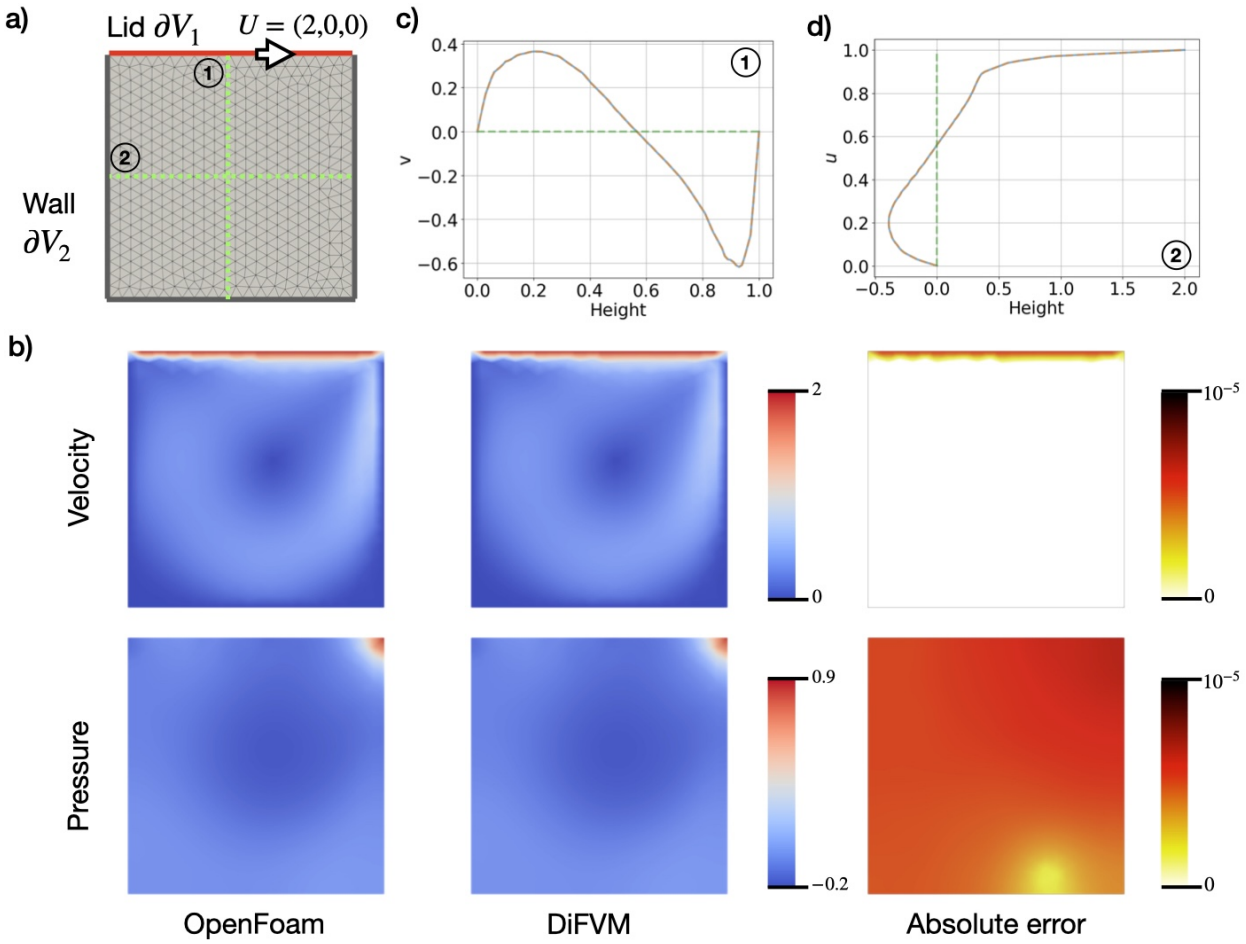

图 6：二维方腔驱动流的基准测试验证。(a) 计算域、非结构网格和边界条件，包括运动顶盖。(b) 使用 OpenFOAM 和 DiFVM 得到的稳态速度大小与压力场对比，以及相应的绝对误差分布。(c) 空腔垂直中心线上的速度剖面。(d) 水平中心线上的速度剖面。DiFVM 与 OpenFOAM 在场分布和中心线剖面方面均表现出极佳的一致性。

以考虑网格非正交性。时间步长为 $\Delta t = 10^{-3}$，模拟推进 20，000 个时间步，以达到统计周期状态。

图 7和8 比较了在一个脱落周期内多个时间点上 DiFVM 与 Open- FOAM 的瞬时速度和压力场。结果清晰地捕捉到了圆柱尾流中周期性涡脱落的开始、空间结构和时间演化，与 $Re = 200$处的经典观测结果一致。整个模拟过程中，速度的绝对误差保持在 10^{-5} 以下，压力的绝对误差保持在 2.5×10^{-4} 以下。这些结果表明，DiFVM 能够准

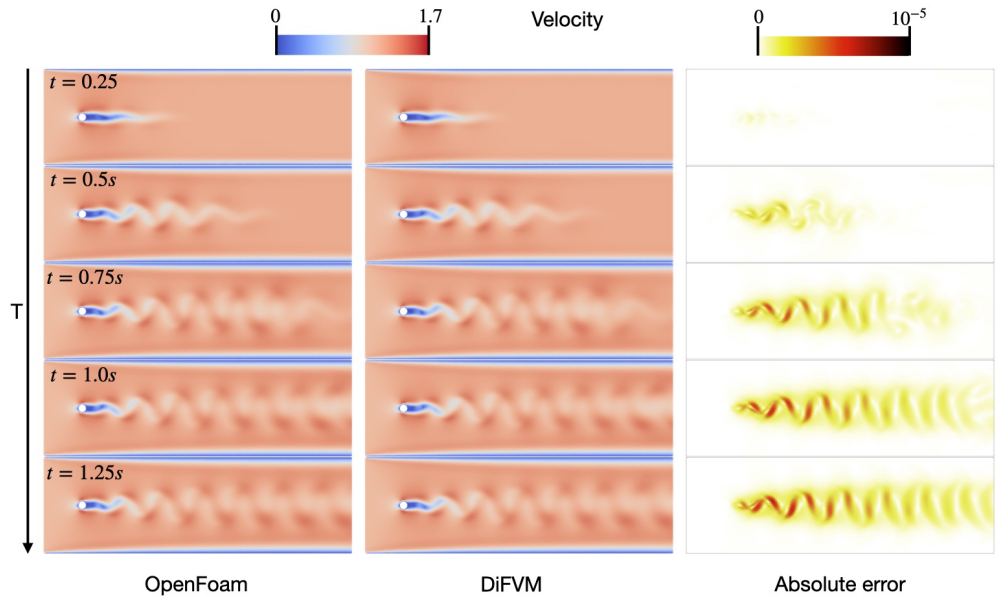

图7： $Re = 200$时的二维圆柱绕流：在一个脱落周期内的五个连续时间点，对比 OpenFOAM（左）和 DiFVM（中）的瞬时速度大小场。

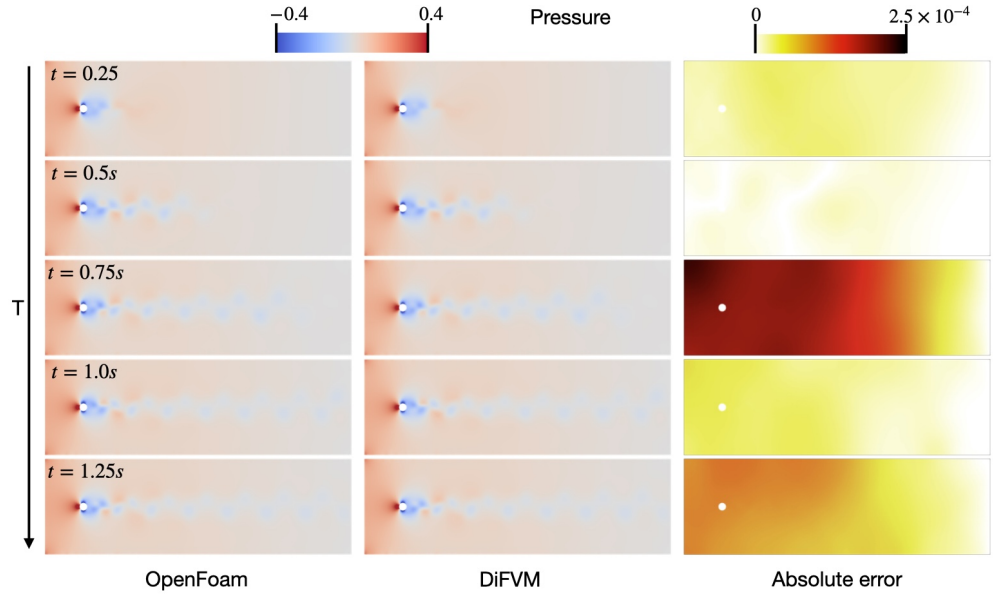

图8：二维圆柱绕流，时间为 $Re = 200$：对比 OpenFOAM（左）和 DiFVM（中）在与图7相同时间点的瞬时压力场。

确解析对流主导的瞬态流动中的非定常尾流动力学和压力–速度耦合。

3.1.6. 三维患者特异性主动脉血流

我们考虑一个取自血管模型库（VMR）的三维患者特异性多分支主动脉几何模型 [65], ，以展示 DiFVM 处理复杂三维血管域的能力。计算网格由主动脉弓及其主要分支的非结构四面体离散构成，包括左右颈总动脉和左右锁骨下动脉，如图9（a）所示。入口处设定抛物线速度分布，以表示充分发展的层流流入，

$$\mathbf{u}(\mathbf{x}) = U_{\max}\left(1 - \frac{r^2}{R^2}\right)\mathbf{n}, \tag{34}$$

其中 r 表示距入口中心线的径向距离， R 表示入口半径，而$\mathbf{n}$ 表示外向单位法向量。血管壁上施加无滑移边界条件。多个出口边界采用零梯度（无牵引）条件，以便各分支自然分配流量。采用后向欧拉时间积分和 PISO算法求解不可压缩纳维–斯托克斯方程。在相同数值格式和边界条件设置下，将 DiFVM 的解与 OpenFOAM 进行验证。

图 9（b）比较了使用 DiFVM 和 OpenFOAM 获得的稳态速度和压力场，以及相应的绝对误差分布。DiFVM 准确再现了主动脉弓中的复杂三维流动模式，包括由曲率和分支引起的速度重新分布，以及沿血管壁产生的相关压力梯度。绝对误差保持较小且局限于局部区域（$\leq 3 \times 10^{-5}$ 对于速度，$\leq 2 \times 10^{-3}$ 对于压力），表明两个求解器之间具有一致的数值处理方式。该基准测试表明，DiFVM 能够稳健地处理真实的患者特异性血管几何结构，并准确解析三维血流动力学。

3.2. 反问题建模示例

3.2.1. 方腔流中的盖板速度恢复

验证 DiFVM 的正向精度后，我们利用其端到端可微性求解顶盖驱动方腔流的反问题。如图10所示，目标是根据在流量区域内 x 个随机选定位置采集的稀疏速度测量值，推断盖板速度 U_x 的分量。 $N = 10$观测算子为

$$\mathcal{H}(\mathbf{U}) = \mathbf{H}\mathbf{U}, \tag{35}$$

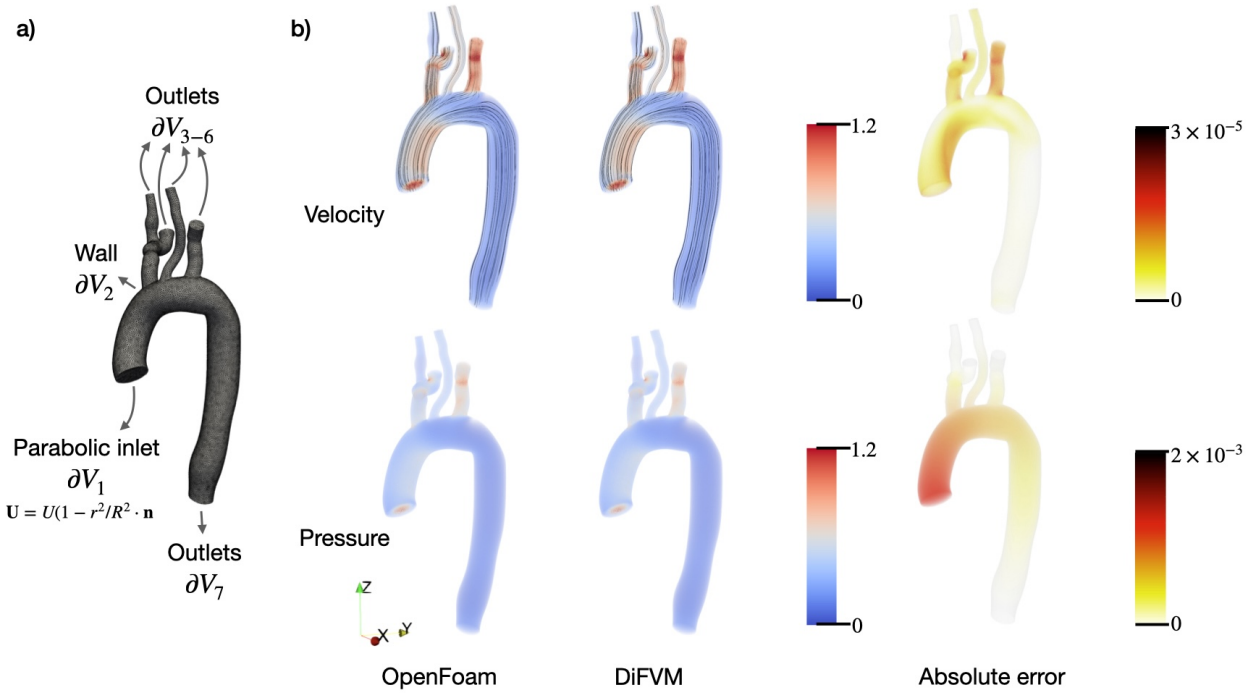

图9：三维多分支主动脉流动模拟。(a) 患者特异性主动脉几何结构及其边界条件，包括抛物线入口速度剖面、无滑移血管壁和多个出口边界（∂V_{3-7}）。(b) 使用 OpenFOAM 和 DiFVM 计算得到的速度大小（上）和压力（下）场对比，以及绝对误差分布。

其中，$\mathbf{H} \in \mathbb{R}^{N \times N_v}$ 是一个掩膜矩阵，用于从完整速度场$\mathbf{U} \in \mathbb{R}^{N_v \times 2}$中提取探测位置处的速度；$N_v$ 是网格单元总数，而$U_{m,k}$ 表示第 k个探测速度测量值。损失函数是探测位置处模拟速度与参考速度之间的均方误差（MSE）：

$$\mathcal{L} = \frac{1}{N} \sum_{k=1}^{N} \left(U_{m,k}^{\text{sim}} - U_{m,k}^{\text{data}} \right)^2, \tag{36}$$

其中，合成数据由正向模拟生成，真实值为 $U_x = 2.0\text{m/s}$。

采用 ADAM 优化器，初始学习率为 1×10^{-1}。初始猜测为 $0。5\text{m/s}$，推断得到的盖板速度在 18 个训练轮次内收敛至 $1。996\text{m/s}$ ，与真实值 $2。0\text{m/s}$ 非常接近（图 10c）。图 10d 展示了四个代表性探测位置处的初始速度和优化后速度；优化后，推断速度与参考值高度吻合。图 10e 进一步证实，逆向框架能够高保真地恢复全局盖板速度和完整流场。

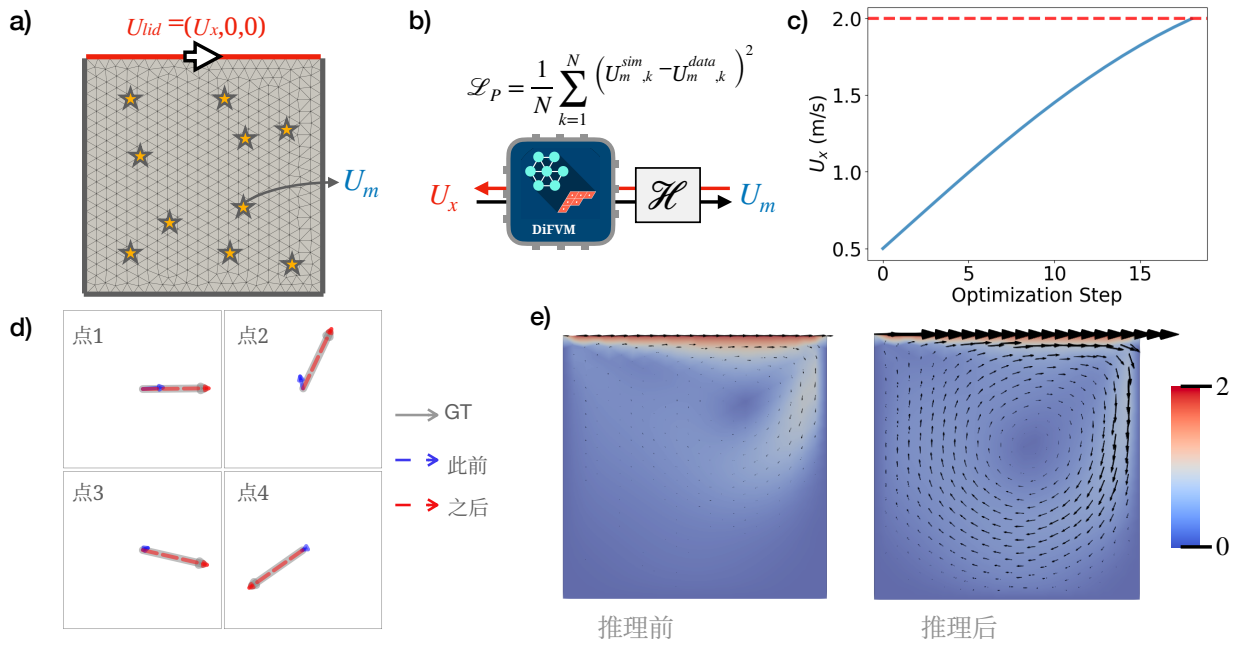

图 10：使用可微计算流体力学进行盖板速度的逆向识别。(a) 方腔流配置，其中盖板速度设为 $U_{\text{lid}} = (U_x,\ 0,\ 0)$，并在 10 个随机选取的探测位置测量内部速度 U_m。(b) 逆向框架：DiFVM 根据 U_x预测速度，并通过最小化模拟速度与测量速度之间的均方误差来优化参数。(c) 推断的盖板速度在优化迭代过程中逐渐趋近真实值。(d) 推理前（蓝色）、推理后（红色）和真实值（灰色）在四个代表性探测位置处的速度矢量。(e) 推理前后的速度大小等值线和矢量场。

3.2.2. 脉动流中的 *Windkessel* 参数推断

我们进一步提高逆问题的复杂度，考虑一种分叉几何，其两个出口均采用三元 Windkessel（RCR）边界条件（图 11）。入口处施加脉动抛物线速度分布。逆问题是根据入口压力波形 $\{R_{p1}, C_1, R_{d1}, R_{p2}, C_2, R_{d2}\}$ 以及两个出口的时间平均流量 $P(t)$ 推断六个 Windkessel 参数 $\{Q_1, Q_2\}$。传统上，这项任务通过经验估算 [66]来完成。首先，根据平均压降和相应的出口流量计算每个出口支路的总阻力：

$$R_{\text{tot},i} = \frac{\overline{P} - P_v}{\overline{Q}_i} = \frac{\frac{1}{T}\int_0^T P(t)\,dt - P_v}{\overline{Q}_i}. \tag{37}$$

随后，按照预设比例将总阻力分为近端分量和远端分量

γ,

$$R_{p,i} = \gamma R_{\mathrm{tot},i}, \tag{38}$$

$$R_{d,i} = (1-\gamma) R_{\mathrm{tot},i}, \tag{39}$$

其中， $\gamma \in [0.05, 0.3]$ 通常用于心血管建模。

根据 Windkessel 时间常数 τ 与舒张期压力波形之间的指数衰减关系估算顺应性，

$$P(t) - P_v \approx A e^{-t/\tau}, \tag{40}$$

由此得到顺应性

$$C_i = \frac{\tau}{R_{d,i}}. \tag{41}$$

在本研究中，我们采用 $\gamma = 0.15$ 进行初始估计。所得参数值列于表2，并用作后续逆向优化的初始状态。

为确保对不同量级的参数进行均衡优化，我们对定义如下的对数缩放系数 $\{\alpha_{R_{p1}}, \alpha_{C_1}, \alpha_{R_{d1}}, \alpha_{R_{p2}}, \alpha_{C_2}, \alpha_{R_{d2}}\}$ 进行优化：

$$R_{pi} = R_{pi}^{\mathrm{gt}}\, e^{\alpha_{R_{pi}}}, \quad C_i = C_i^{\mathrm{gt}}\, e^{\alpha_{C_i}}, \quad R_{di} = R_{di}^{\mathrm{gt}}\, e^{\alpha_{R_{di}}}, \tag{42}$$

其中， $i \in \{1, 2\}$ 表示出口索引，上标“gt”表示真实值。这种指数参数化可确保 Windkessel 参数严格为正，并将真实值映射至零向量，从而促进平滑优化。观测算子将模拟流场 $\{\mathbf{U}, \mathbf{P}\}$ 映射为可测量量：

$$\mathcal{H}(\mathbf{U}, \mathbf{P}) = \left(P_{\mathrm{in}}(t),\, \overline{Q}_1,\, \overline{Q}_2\right), \tag{43}$$

其中，面积平均入口压力为

$$P_{\mathrm{in}}(t) = \frac{\sum_{k \in \partial\Omega_{\mathrm{in}}} P_k(t)\, A_k}{\sum_{k \in \partial\Omega_{\mathrm{in}}} A_k}, \tag{44}$$

以及出口 i 处的时间平均出口流量为

$$\overline{Q}_i = \frac{1}{T} \int_0^T \sum_{k \in \partial\Omega_{\mathrm{out},i}} \mathbf{U}_k(t) \cdot \mathbf{n}_k\, A_k\, dt, \quad i = 1, 2, \tag{45}$$

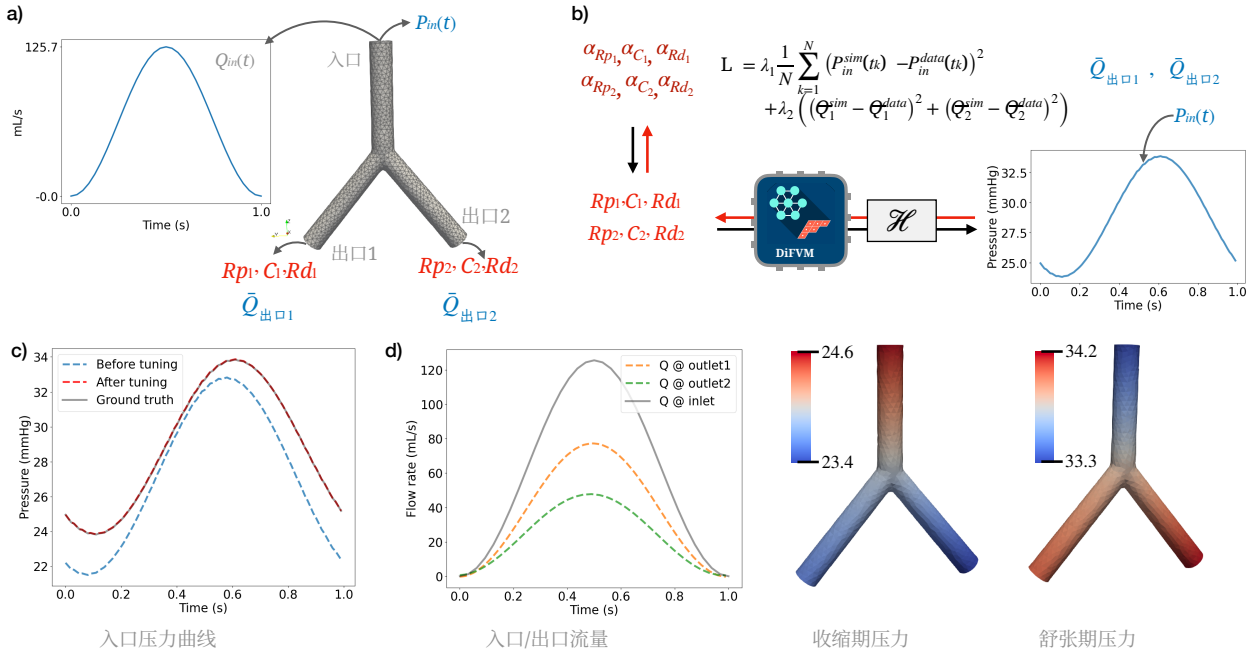

图 11：分叉模型中 RCR 边界参数的逆向识别。(a) 问题设置：在入口处指定脉动流入 $Q_{\text{in}}(t)$，每个出口则采用参数为 (R_p, C, R_d) 的 RCR 边界条件。(b) 逆向框架：通过最小化复合损失来优化出口参数 $\{R_{p,i}, C_i, R_{d,i}\}$，该损失结合了入口压力波形误差和出口时间平均流量差异。(c) 调整前（蓝色）、调整后（红色）的入口压力波形及真实值（黑色），表明成功准确地恢复了目标压力曲线。(d) 入口和两个出口的流量波形，证实参数推断后仍满足质量守恒。(e) 调整后收缩期和舒张期的压力场。

其中， A_k 和 $\mathbf{n}_k$ 分别表示第 k 个表面单元的面积和外向单位法向量。计算损失之前，会对压力和流量进行归一化，以平衡梯度幅值。总损失结合了入口压力波形误差和时间平均出口流量误差：

$$\mathcal{L} = \lambda_1 \mathcal{L}_P + \lambda_2 \mathcal{L}_Q, \tag{46}$$

其中

$$\mathcal{L}_P = \frac{1}{N} \sum_{k=1}^{N} \left(P_{\text{in}}^{\text{sim}}(t_k) - P_{\text{in}}^{\text{data}}(t_k) \right)^2, \tag{47}$$

$$\mathcal{L}_Q = \left(\overline{Q}_1^{\text{sim}} - \overline{Q}_1^{\text{data}} \right)^2 + \left(\overline{Q}_2^{\text{sim}} - \overline{Q}_2^{\text{data}} \right)^2. \tag{48}$$

使用 ADAM 优化器，初始学习率为 1×10^{-1}，并将所有目标参数的相对误差低于 1% 定义为收敛。模拟使用 $\Delta t = 0.001$ 秒，共

持续时间 $T_{\text{total}} = 6$ 秒 （$N_t = 6000$ 个时间步），足以达到稳定周期状态。真实值参数和推断值参数总结于表 2。

T表 2：双出口分叉情形的真实值和推断值 Windkessel 参数。

参数	Unit	初始猜测	真实值	推断值	相对误差
R_{p1}	dyn·s/cm^5	150.421	100.000	100.897	0.90%
C_1	cm^5/(dyn·s)	1.1732×10^{-3}	1.1111×10^{-3}	1.1111×10^{-3}	$< 0.01\%$
R_{d1}	dyn·s/cm^5	852.386	900.000	900.000	$< 0.01\%$
R_{p2}	dyn·s/cm^5	240.159	160.000	160.399	0.25%
C_2	cm^5/(dyn·s)	7.3481×10^{-4}	6.9444×10^{-4}	7.0129×10^{-4}	0.99%
R_{d2}	dyn·s/cm^5	1360.899	1440.000	1440.000	$< 0.01\%$

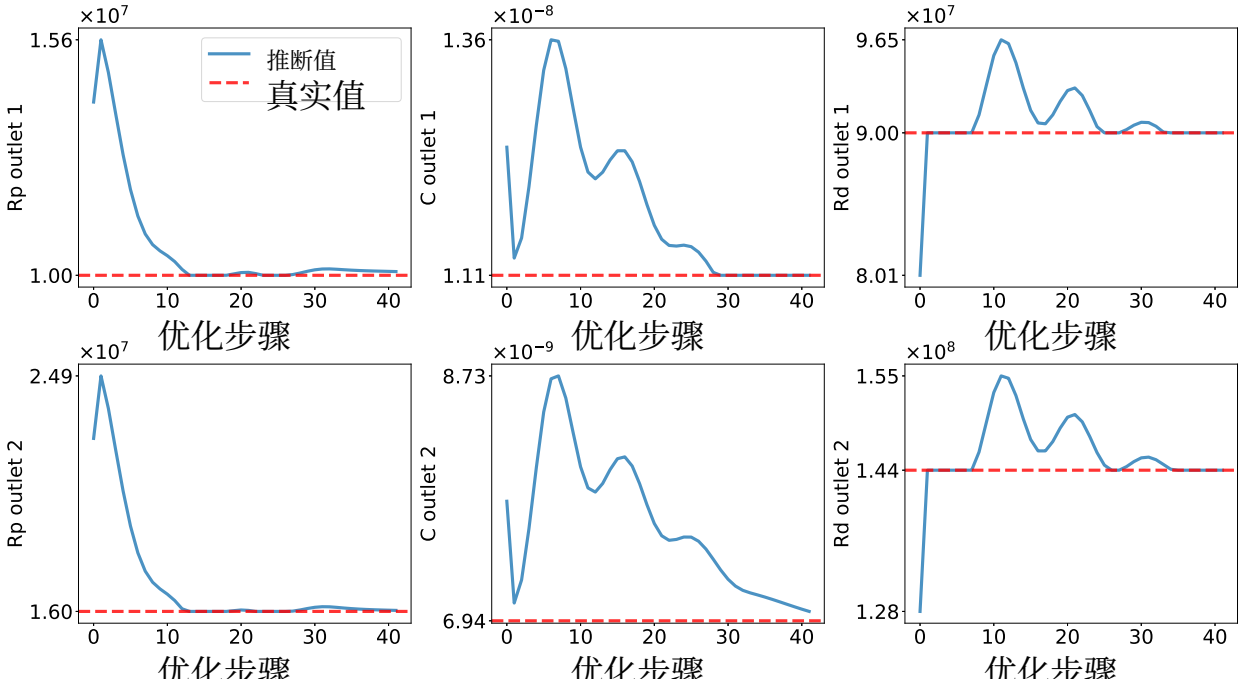

图12：优化过程中推断的RCR参数的收敛情况。优化迭代期间，两个出口的近端阻力 R_p、顺应性 C和远端阻力 R_d 的变化情况。蓝色实线表示推断的参数值；红色虚线表示真实值。所有参数均在约20–30次迭代内收敛至其真实值，表明温德克塞尔参数恢复稳定且准确。

全部六个参数在 41 个优化轮次后均收敛至与真实值相差不超过 1%（图 12）。近端和远端阻力均在约 12 个训练轮次内迅速收敛，这反映出流量分配对阻力值具有较强的敏感性。顺应性参数的收敛

速度较慢，因为压力波形主要受电容影响；远端阻力会暂时调整以补偿顺应性失配，之后与电容参数共同收敛。图 11（c）展示了优化前后的入口压力波形，表明目标波形得到了准确恢复。图 11（d）展示了入口和两个出口处的流量波形，证实了质量守恒。收缩期和舒张期的压力等值线（图 11（e））呈现出动态一致的压力分布：收缩期，入口至出口之间显著的压降驱动正向流动；舒张期，温塞尔边界中的电容释放导致暂时出现反向压力梯度，这与脉动流体物理相符。

4. 讨论

4.1. 计算效率

图13 报告了 DiFVM 在所有正向基准测试案例中的墙钟时间，并与运行在单个 CPU 核心、16 个 CPU 核心和 32 个 CPU 核心上的 OpenFOAM 进行比较，其中 DiFVM 运行于 NVIDIA L40 GPU 上。所有比较均采用相同的网格、边界条件和数值设置。需要注意的是，所有 DiFVM 计时均在单个

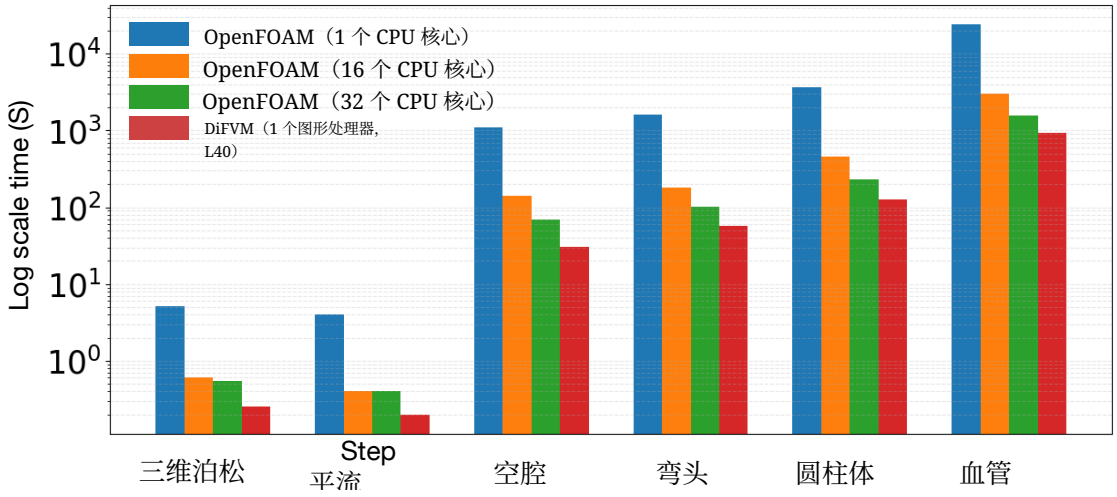

图13：在对数尺度下，DiFVM（1 个 Nvidia L40 GPU）与 OpenFOAM（1、16 和 32 个 CPU 核心）在所有正向案例中的墙钟时间比较。DiFVM 始终优于所有 CPU 配置，在所有案例中，相较于单核心 OpenFOAM 实现了 16–33×倍加速，相较于 32 核并行 OpenFOAM 实现了 1.6–2.3× 倍加速。

NVIDIA L40 GPU，未采用任何多 GPU 或多节点并行。加速比曲线呈现出

随问题规模变化的典型趋势，反映了 GPU 向量化效率、内存带宽和算术强度之间的相互作用。对于最小规模的问题（三维泊松和阶跃平流），得益于 XLA 内核融合以及消除 Python 级循环开销，单个 GPU 相较单核中央处理器已实现 $\sim 16\times$ 的加速；不过，工作负载不足以让 GPU 的计算单元完全饱和。相对加速幅度最大的是中大型瞬态问题：与单核中央处理器相比，加速达到 ~ 25–$33\times$ ；与 32 核并行 OpenFOAM 相比，加速达到 $\sim 2\times$ 。这些问题的计算工作负载足够大，可以摊销 GPU 内核启动开销，同时仍处于带宽高效运行区间。值得注意的是，对于所有测试案例，单个 GPU 的速度仍比 32 核并行 OpenFOAM $\sim 1.4\times$ 快，这凸显了 DiFVM 基于图的向量化策略在单个加速器上的效率。

这些性能提升体现了 DiFVM 的双层向量化策略。在空间维度上，所有有限体积通量计算都被重新表述为基于静态网格图的批量化收集–计算–散布操作，并编译为融合的 XLA GPU 内核，从而消除了逐个单元顺序遍历。在时间维度上，PISO 时间推进循环通过 jax.lax.scan 展开为单个静态计算图，XLA 再将其编译为少量 GPU 可执行程序，使时间步之间无需进行主机与设备同步，避免了同步开销。总体而言，这些转换确保即使是最大的瞬态基准测试——涉及数千个时间步和迭代压力求解——也几乎完全在设备上运行，无需承担 Python 开销。

当前的基准测试仅展现了现代加速器硬件能力的一小部分。三维血管基准测试（$\sim 10^6$ 个单元）在仅执行前向传播的模式下约占用 2.3 GB GPU 内存，而 NVIDIA L40（48 GB）仍有大量余量。这表明，无需进行架构变更，就能处理规模大一至两个数量级的网格；这也意味着，随着问题规模扩大至接近硬件的全部容量，DiFVM 的性能优势将更加显著。对于长时间瞬态模拟中的反向模式自动微分，可通过梯度检查点管理内存：这会以额外执行一次前向传播为代价来降低内存占用，从而使原本会超出 GPU 内存容量的规模也能进行可微分模拟。

4.2. 局限性与未来工作

当前框架求解刚性壁面几何中的不可压缩纳维–斯托克斯方程。尚未纳入的高保真扩展包括流固耦合、通过大涡模拟（LES）进行湍流建模

或采用雷诺平均纳维–斯托克斯（RANS）闭合模型，以及与降阶一维/零维心血管网络模型的多尺度耦合。这些能力对于患者特异性血流动力学模拟至关重要，尤其适用于涉及顺应性血管壁、主动脉湍流或体循环建模的应用。在 Diff-FlowFSI 可微流固耦合框架的基础上，将 DiFVM 扩展至流固耦合和多物理场耦合，是未来工作的主要方向。 [57],

当前实现运行于单个图形处理器上。尽管本文考察的最大基准测试在单块 NVIDIA L40（48 GB）上仍有充足余量，但生产规模的模拟（例如患者特异性血管网络、高分辨率大涡模拟或全身血流动力学模型）需要包含 10^8–10^9 个单元的网格，超出单个图形处理器的容量。DiFVM 的基于图的形式在架构上兼容多 GPU 执行：静态网格图 $\mathcal{G}$ 可利用标准图划分算法（例如 METIS [67]）划分到多个设备上，分区间的光环交换则通过 MPI 或 JAX 原生的 jax.distributed 接口处理。在这种多 GPU 范式下，每个设备持有网格图 $\mathcal{G}$ 的一个子图，并在本地执行相同的散射/聚集消息传递内核，同时在分区之间传递边界面通量。由于图结构是静态的，通信模式在初始化时便已固定，并可在编译时进行一次性优化。通过 MPI–GPU 并行将 DiFVM 扩展至多 GPU 和多节点执行，是提升可扩展性的主要方向；这将释放可微非结构网格计算流体力学在生产级工程规模下的全部潜力，在这一规模下，相较于基于中央处理器的求解器，性能优势预计将进一步扩大。

5. 结论

我们介绍了 DiFVM，这是首个能够原生运行于非结构多面体网格上的 GPU加速、端到端可微有限体积计算流体力学求解器。其核心贡献并非对现有求解器进行渐进式改进，而是从架构层面进行重构：通过建立有限体积法离散与图神经网络消息传递之间的结构同构，DiFVM 将非结构网格的不规则连通性——历来是 GPU 向量化的主要障碍——转换为静态编译时数据结构，从而适用于完全融合的 XLA GPU 内核。这种重构精确且无损；每次通量计算、PISO 压力—速度耦合迭代以及 Rhie–Chow 修正，均表示为网格图上的散射/收集操作

，且不会对底层数值方法作任何近似。JAX 的反向模式自动微分支持贯穿完整仿真流程的端到端自动微分（AD）；通过梯度检查点以及对泊松方程求解进行隐式微分，确保长时间瞬态模拟的内存效率。

在六个几何和物理复杂度逐步提升的基准测试中，从典型流动到特定患者的心血管血流动力学，正向验证表明，DiFVM 与 OpenFOAM 的结果在定量上吻合；对于所有场量和几何形状，相对误差始终低于 $\mathcal{O}(10^{-5})$。在单个 NVIDIA L40 GPU 上，DiFVM 在所有案例中均持续优于 32 核并行 OpenFOAM，且性能优势随问题复杂度增加而扩大。这些结果并未借助任何多 GPU 并行；DiFVM 的静态图结构在架构上兼容跨多个设备进行域分解，这表明在生产级工程规模下，其相对于基于 CPU 的求解器的性能优势将进一步扩大。

通过两个复杂度逐步提升的反问题，展示了端到端可微性：方腔流中的标量边界参数恢复，以及脉动分叉流中六个 Windkessel 参数的同步推断。基于稀疏的压力和流量观测，所有参数均收敛至真实值的 1% 以内。这些结果表明，梯度能够正确地穿过数千个时间步、迭代求解过程和非线性通量计算，从而证实了自动微分流程在瞬态计算流体力学模拟的全部复杂环节中均能正确运行。

DiFVM 解决了可微计算流体力学生态系统中的一个结构性空缺：在定义了最重要的航空航天和生物医学应用的非结构网格上，缺少具备 GPU 原生执行能力和梯度计算能力的求解器。通过在单一基于 JAX 的框架中统一有限体积法离散、GPU 原生执行和端到端可微性，DiFVM 为梯度驱动工作流程提供了缺失的计算基础，包括偏微分方程约束优化、贝叶斯参数推断、数据同化，以及在以往无法进行可微分模拟的几何复杂域上开展混合机器学习-计算流体力学建模的环路内训练。

参考文献

[1] K. Fujii，《航空航天科学进展》中关于航空航天领域计算流体力学的进展与未来展望——风洞及其他，41（6）（2005）455–470。

[2] S. E. Jujjavarapu、T. Kumar、S. Gupta，《生物医学工程中的计算流体力学》，载于：

《生物与生物医学过程中的计算流体力学应用：生物技术应用》，Springer，2024 年，第 101–125 页。

[3] M. H. Zawawi、A. Saleha、A. Salwa、N. Hassan、N. M. Zahari、M. Z. Ramli、Z. C. Muda，《计算流体力学（CFD）基础综述》，载于：《AIP会议论文集》，第2030卷，AIP Publishing LLC，2018年，第020252页。

[4] J. I. Rojas-Sola、C. García-Baena、M. J. Hermoso-Orzáez，《计算流体动力学仿真软件综述：优势、劣势及主要应用》，《磁流体动力学与等离子体研究杂志》21（4）（2016年）417–424。

[5] S. Pant、C. Corsini、C. Baker、T.-Y. Hsia、G. Pennati、I. E. Vignon-Clementel，《心血管血流动力学降阶模型中的逆问题：数据同化与心率变异性方面》，《英国皇家学会界面杂志》14（126）（2017）20160513。

[6] J.-X. Wang、X. Hu、S.C.Shadden，颅内压力的数据增强建模，《生物医学工程年鉴》47 (3) (2019年) 714–730。

[7] T. Bui-Thanh、M.Damodaran、K.Willcox，利用适当正交分解进行气动数据重构与逆向设计，AIAA期刊 42 (8) (2004) 1505–1516。

[8] J. Li、S. He、J. R.Martins，跨声速气动外形优化中用于确保低速性能的数据驱动约束方法，《航空航天科学与技术》92 (2019年) 536–550。

[9] M. Lino、S. Fotiadis、A.A.Bharath、C.D. Cantwell，流体动力学模拟的当前及新兴深度学习方法，《英国皇家学会会刊A辑：数学、物理与工程科学》479 (2275) (2023)。

[10] J.-X. Wang, J. Huang, L.Duan, H. Xiao，使用物理信息机器学习预测高马赫数湍流边界层中的雷诺应力，《理论与计算流体动力学》33 (1) (2019年) 1–19。

[11] S. L. Brunton、B. R. Noack、P. Koumoutsakos，《流体力学中的机器学习》，《流体力学年度评论》52（1）（2020）477–508。

[12] J.-X. Wang、H. Xiao，复杂结构中湍流的数据驱动的计算流体力学建模，《国际传热与流体流动杂志》62（2016年）138–149。[13] A. F. da Silva、T. Colonius，空气动力流动的基于集合的状态估计器，《AIAA期刊》56（7）（2018年）2568–2578。[14] R. Maulik、V. Rao、J. Wang、G. Mengaldo、E. Constantinescu、B. Lusch、P. Balaprakash、I. Foster、R. Kotamarthi，利用机器学习降阶模型进行高效的高维变分数据同化，《地球科学模型开发》15（8）（2022）3433–3445。[15] M. Wang、T. A. Zaki，壁面湍流中的变分数据同化：从外部观测到壁面应力和压力，《流体力学杂志》1008（2025年）A26。

[16] S. Pan, K. Duraisamy，数据驱动的闭合模型发现，《应用动力系统 SIAM 杂志》17(4) (2018年) 2381–2413。

[17] J. Zhuang, D. Kochkov, Y.Bar-Sinai, M. P. Brenner, S. Hoyer，二维湍流中的被动标量对流学习型离散化，《物理评论：流体》6 (6) (2021年) 064605。

[18] K. Duraisamy, G. Iaccarino, H. Xiao，数据时代的湍流建模，《流体力学年度评论》51(1) (2019年) 357–377。

[19] J.-X. Wang, J.-L. Wu, H. Xiao，基于直接数值模拟数据重建雷诺应力建模偏差的物理信息机器学习方法，《物理评论：流体》2(3) (2017) 034603。[20] X. Yang, S. Zafar, J.-X. Wang, H. Xiao，通过物理信息神经网络进行预测性大涡模拟壁面建模，《物理评论：流体》4 (3) (2019年) 034602。

[21] H. D. Akolekar、J. Weatheritt、N. Hutchins、R. D. Sandberg、G. Laskowski、V. Michelassi，《用于增强低压涡轮尾流混合预测的机器学习代数雷诺应力模型的开发与应用》，《涡轮机械杂志》141（4）（2019年）041010。

[22]A. Prakash, K. E. Jansen, J. A. Evans，《面向大涡模拟的各向异性网格不变数据驱动亚格子应力建模》，《计算力学与工程方法》422 (2024) 116807。

[23] J. Wu, H. Xiao, R. Sun, Q. Wang，采用显式数据驱动雷诺应力闭合的雷诺平均纳维–斯托克斯方程可能存在病态问题，《流体力学杂志》869 (2019) 553–586。[24] A. Beck, D. Flad, C.-D. Munz，面向数据驱动大涡模拟闭合模型的深度神经网络，计算物理学杂志 398 (2019) 108910。[25] J. Sirignano, J. F. MacArt, J. B. Freund，DPM：一种深度学习偏微分方程增强方法及其在大涡模拟中的应用，计算物理学杂志 423 (2020) 109811。[26] J. F. MacArt, J. Sirignano, J. B. Freund，神经网络亚格子尺度湍流模型的嵌入式训练，《物理评论：流体》6 (5) (2021) 050502。[27] Y. Fang, Y. Zhao, F. Waschkowski, A. S. Ooi, R. D. Sandberg，通过多工况计算流体力学驱动训练构建更通用的湍流模型，《AIAA期刊》61 (5) (2023) 2100–2115。[28] X. Fan, J.-X. Wang，面向流固耦合的可微混合神经建模，计算物理学杂志 496 (2024) 112584。[29] F. Sapienza, J. Bolibar, F. Schäfer, B. Groenke, A. Pal, V. Boussange, P. Heimbach, G. Hooker, F. Pérez, P.-O. Persson 等，可微分方程编程：综述，arXiv 预印本 arXiv:2406.09699 (2024)。[30] B. List, L.-W. Chen, N. Thuerey，利用可微流体求解器进行学习型湍流建模：基于物理的损失函数与优化时域，《流体力学杂志》949 (2022) A25。[31] V. Shankar, D. Chakraborty, V. Viswanathan, R. Maulik，可微湍流：将闭合问题转化为偏微分方程约束优化，《物理评论：流体》10 (2) (2025) 024605。[32] Y. Bar-Sinai, S. Hoyer, J. Hickey, M. P. Brenner，学习偏微分方程的数据驱动离散化，《美国国家科学院院刊》116 (31) (2019) 15344–15349。

[33] K. Um, R. Brand, Y. R. Fei, P. Holl, N. Thuerey, 《求解器在环：从可微物理中学习以与迭代偏微分方程求解器交互》，《神经信息处理系统进展》第 33 卷（2020），6111–6122。

[34] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, 《机器学习加速计算流体力学》，《美国国家科学院院刊》118(21)（2021）e2101784118。

[35] D. A. Bezgin, S. J. Schmidt, N. A. Adams, 《用于非经典欠压缩激波的数据驱动、物理信息有限体积格式》，计算物理学杂志 437（2021）110324。

[36] F. D. A. Belbute-Peres, T. Economon, Z. Kolter, 《结合可微偏微分方程求解器与图神经网络进行流体流动预测》，载于：国际机器学习会议，PMLR，2020年，第 2402–2411 页。

[37] X.-Y. Liu、M. Zhu、L.Lu、H. Sun、J.-X. Wang, 《用于时空动力学的多分辨率偏微分方程保持学习框架》，Communications Physics 7 (1) (2024) 31。

[38] D. Akhare、T. Luo、J.-X. Wang, 《Diffhybrid-uq：可微混合神经建模的不确定性量化》，arXiv 预印本 arXiv:2401.00161 (2023)。

[39] D. Akhare、T. Luo、J.-X. Wang, 《Hybridndiff-uq：混合神经可微建模的不确定性量化》，理论与应用力学快报 (2025) 100609。

[40] D. Akhare、P. Du、T. Luo、J.-X. Wang, 用于时空动力学的隐式神经可微模型, Computer Methods in Applied Mechanics and Engineering 446 (2025) 118280。doi:https://doi.org/10.1016/j.cma.2025.118280。网址：https://www.sciencedirect.com/science/article/pii/S0045782525005523

[41] X. Fan、D. Akhare、J.-X. Wang, 基于扩散的二维时空湍流超分辨率神经可微建模, 《计算机辅助力学与工程方法》433（2025年）117478。

[42] W. Shang、J. Zhou、J. Panda、Z. Xu、Y. Liu、P. Du、J.-X. Wang、T. Luo, Jax-bte：一款图形处理器

加速的可微分声子玻尔兹曼输运方程求解器，《npj《计算材料》》11 (1) (2025年) 1–12。

[43] D. Akhare、Z. Chen、R. Gulotty、T. Luo、J.-X. Wang，《用于等温化学气相渗透工艺的概率物理集成神经可微建模》，npj《计算材料》10 (1) (2024) 120。

[44] T. D. Economon、F. Palacios、S. R. Copeland、T. W. Lukaczyk、J. J. Alonso，《SU2：用于多物理场仿真和设计的开源套件》，AIAA期刊 54 (3) (2016年) 828–846。

[45] S. Mitusch、S. Funke、J. Dokken，《dolfin-adjoint 2018.1：面向 FEniCS 和 Firedrake 的自动伴随》，《开放源代码软件期刊》4 (38) (2019年) 1292。

[46] C. A. Mader、J. R. Martins、J. J. Alonso、E. Van Der Weide，《ADjoint：快速开发离散伴随求解器的一种方法》，AIAA期刊 46 (4) (2008年) 863–873。

[47] S. D. I. Software，《Simcenter STAR-CCM+用户指南》，版本 2021.1，载于：重叠网格的自适应网格细化，西门子，2021年，第 3067–3070 页。

[48] J. Bradbury、R. Frostig、P. Hawkins、M. J. Johnson、C. Leary、D. Maclaurin、G. Necula、A. Paszke、J. VanderPlas、S. Wanderman-Milne、Q. Zhang，JAX：可组合的 Python+NumPy 程序转换(2018年)。网址 http://github.com/jax-ml/jax

[49]A. Paszke、S. Gross、F. Massa、A. Lerer、J. Bradbury、G. Chanan、T. Killeen、Z. Lin、N. Gimelshein、L. Antiga 等，《PyTorch：一种命令式风格、高性能的深度学习库》，《神经信息处理系统进展》第 32 卷（2019年）。

[50] M. Abadi、P. Barham、J. Chen、Z. Chen、A. Davis、J. Dean、M. Devin、S. Ghemawat、G. Irving、M. Isard 等，{TensorFlow}：一个用于 {大规模} 机器学习的系统，载于：第 12 届 USENIX 操作系统设计与实现研讨会（OSDI 16），2016年，第 265–283 页。

[51] J. Bezanson, A.Edelman, S. Karpinski, V. B. Shah, Julia：一种数值计算的新方法，SIAM评论 59 (1) (2017) 65–98。 doi:10.1137/141000671。网址 https://epubs.siam.org/doi/10.1137/141000671

[52] S. Posey, J. Luitjens, O. Hennigh, S. Oberlin，面向计算流体力学的基于GPU的高性能计算与人工智能发展，美国夏威夷毛伊岛，7月（2022）11–15。[53] P. Holl, N. Thuerey，ϕ-flow：面向 PyTorch、TensorFlow 和 JAX 的可微分模拟，载于：第四十一届国际机器学习会议，2024。[54] G. D. Weymouth, B. Font，WaterLily.jl：一种可微且与后端无关的 Julia 求解器，用于模拟不可压缩黏性流和动态物体，arXiv 预印本 arXiv:2407.16032（2024）。[55] D. A. Bezgin, A. B. Buhendwa, N. A. Adams，Jax-fluids 2.0：迈向可压缩两相流可微计算流体力学的高性能计算，《计算机物理通讯》308（2025年）109433。[56]A. Franz, H. Wei, L. Guastoni, N. Thuerey，PICT——一种可微、GPU加速的多区块 PISO 求解器，用于与模拟耦合的流体动力学学习任务，arXiv 预印本 arXiv:2505.16992（2025年）。[57] X. Fan, X.-Y. Liu, M. Wang, J.-X. Wang，Diff-flowfsi：一种针对高保真湍流和流固耦合模拟优化的GPU可微计算流体力学平台，《计算力学与工程方法》448（2026）118455。[58] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. Battaglia，利用图网络学习基于网格的模拟，载于：国际学习表征会议，2020。[59] H. G. Weller, G. Tabor, H. Jasak, C. Fureby，利用面向对象技术进行计算连续介质力学的张量方法，《物理学计算》12（6）（1998）620–631。[60] A. Updegrove, N. M. Wilson, J. Merkow, H. Lan, A. L. Marsden, S. C. Shadden，SimVascular：用于心血管模拟的开源流程，《生物医学工程年鉴》45（3）（2017）525–541。[61] C. M. Rhie, W.-L. Chow，采用尾缘分离的翼型湍流流动数值研究，《AIAA期刊》21（11）（1983）1525–1532。[62] H. Jasak，有限体积法的误差分析与估计及其在流体流动中的应用，博士论文，伦敦帝国学院（1996）。

[63] R. I. Issa，通过算子分裂求解隐式离散流体流动方程，《计算物理学杂志》62 (1) (1986) 40–65。[64]M. Blondel、V. Roulet，可微编程的要素，arXiv 预印本 arXiv:2403.14606 (2024)。[65] N. M. Wilson、A. K. Ortiz、A. B. Johnson，血管模型库：医学影像数据与血流模拟结果的公共资源，《医学器械杂志》7 (4) (2013) 0409231–409231。[66] N. Westerhof、J.-W. Lankhaar、B. E. Westerhof，动脉风箱模型，《医学与生物工程与计算》47 (2) (2009) 131–141。[67] G. Karypis、V. Kumar，一种用于划分非规则图的快速高质量多层方案，《SIAM科学计算杂志》20 (1) (1998) 359–392。