arXiv:2601.04400v1 [physics.flu-dyn] 7 Jan 2026

JAX-Shock：一种用于可压缩流动模拟的可微、GPU 加速、激波捕捉神经求解器

Bo Zhang[a,*]

[a]北伊利诺伊大学机械工程系，美国伊利诺伊州迪卡尔布，邮编 *60115*

摘要

理解激波-固体相互作用仍是可压缩流体动力学领域的一项核心挑战。我们提出了 JAX-Shock：一种完全可微、GPU 加速、高阶的激波捕捉求解器，可高效模拟可压缩纳维-斯托克斯方程。该框架完全基于 JAX 构建，利用自动微分实现基于梯度的优化、参数推断，以及深度学习增强模型的端到端训练。求解器结合五阶 WENO 重构与 HLLC通量，以高保真度解析激波和间断。为处理复杂几何形状，我们实现了一种浸入边界方法，以准确表示可压缩流场中的固体界面。此外，我们还引入了一个经过训练的神经通量模块，用数据驱动的修正来增强数值通量，从而显著提高精度和泛化能力。JAX-Shock 还支持用于激波相互作用预测的序列到序列学习，以及通过反向模式推断从数据中识别关键物理参数。与纯数据驱动方法相比，JAX-Shock 在增强泛化能力的同时保留了物理一致性。该框架为可微物理、基于学习的建模以及以复杂激波-固体相互作用为主导的可压缩流动工况下的逆向设计，提供了一个灵活的平台。

关键词： 神经求解器、可微物理、反向学习、激波-固体相互作用、浸入边界法

1. 引言

由可压缩欧拉方程描述的复杂激波-固体相互作用模拟，是国防、航空航天和物理科学领域的核心问题，应用广泛，涵盖导弹气动学 [1], 、超燃冲压发动机推进 [2], 和超音速飞行器设计 [3], 以及惯性约束聚变 [4] 和

*通讯作者。地址：美国伊利诺伊州迪卡尔布，邮编60115电子邮箱： bzhang@niu.edu (Bo Zhang)

超新星爆炸等天体物理现象 [5]。由于强间断、复杂流动结构以及流体与固体边界间复杂耦合现象并存，准确捕捉这些相互作用仍是计算流体力学（CFD）面临的核心挑战。高阶激波捕捉方法，例如结合近似黎曼求解器的加权本质无振荡（WENO）格式 [6, 7], ，大幅提升了计算流体力学对激波主导流动的预测能力。然而，通量格式和人工黏性 [8], 对于稳定数值激波至关重要，却长期阻碍着稳健且可微数值求解器的开发。这些组件固有的非线性和不连续性，使基于梯度的优化和敏感性分析尤其困难。为解决这些局限，本文提出一种用于可压缩流动模拟的可微、GPU 加速、激波捕捉神经求解器，为激波主导状态下的可微物理、基于学习的建模、逆向设计和高性能模拟奠定灵活的平台。

经典的无求解器深度学习方法通过神经网络学习有限维欧几里得空间之间的映射 [9, 10] ，并通过神经算子 [11]将这一能力扩展到函数的无限维巴拿赫空间。相比之下，可微编程提供了一个统一框架，连接科学计算与机器学习（ML） [12], ，使传统数值求解器能够与端到端可训练的机器学习架构无缝集成。这种集成支持直接利用仿真或实验数据，对物理参数进行基于梯度的优化、开展数据同化以及发现控制方程。值得注意的是，物理信息神经网络（PINN） [13] 已成为可微编程的一种重要实现形式，它将物理定律作为软约束嵌入损失函数，从而实现无求解器、受物理约束的学习。在 Python 中用于自动微分（AD）的先进工具中，TensorFlow [14],、PyTorch [15], 和 JAX [16] 已成为主流框架。TensorFlow 和 PyTorch 在机器学习社区中得到广泛采用，而 JAX 则因其可组合的函数变换（例如grad、vmap 和 pmap）以及通过加速线性代数（XLA）实现的即时（JIT）编译，在科学计算领域尤为受到青睐；这些特性共同支持高性能、GPU 加速且完全可微的数值仿真。

近期大量研究聚焦于开发可微混合神经求解器，将机器学习模型与传统数值模拟框架相结合，展现出其在可扩展、基于物理的可微计算方面的潜力。这一范式已通过基于梯度的端到端优化在众多领域得到验证，包括流体动力学 [17, 18, 19, 20, 21], ，以及有限元方法 [22], 、分子动力学 [23], 、纳米尺度热传递 [24], 和密度泛函理论 [25]等其他领域。在流体动力学领域，Sirignano 等人 [17]利用神经网络从数据中学习未知物理规律，并对控制偏微分方程进行扩充。与纯黑箱机器学习方法不同，Kochkov 等人 [18] 对二维湍流进行了建模，采

用端到端可微框架，实现了显著的计算加速，并能很好地泛化到未见过的流动状态。List 等人 [19] 进一步开发了一种可微数值求解器，使优化梯度能够通过多个求解器步骤传播，从而训练湍流模型，改善不可压缩纳维–斯托克斯方程欠分辨率、低分辨率解的效果。Bezgin 等人 [20] 推出了 JAX-Fluids，这是一款功能全面、完全可微的计算流体力学求解器，适用于可压缩两相流，可实现端到端优化，并将机器学习与计算流体力学无缝结合。

尽管取得了这些进展，但能够通过浸入边界法（IBM）模拟激波-固体相互作用的可微神经求解器仍有待探索。据我们所知，本研究提出了首个融合浸入边界法、用于模拟存在激波-固体相互作用的可压缩流的可微混合神经求解器，标志着朝着复杂流固系统的可微物理建模迈出了重要一步。此外，我们还引入了一种神经通量模块，通过数据驱动的修正增强数值通量，显著提升了精度和泛化能力。

2. 方法

本文开发了一种可微混合神经求解器，用于研究可压缩流体中的激波-固体相互作用。此类可压缩无黏流由欧拉方程描述；为求一般性，本文将其写为二维形式：

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0\,, \tag{1}$$

其中，守恒变量向量和通量张量定义为：

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ u(E+p) & v(E+p) \end{bmatrix},$$

其中，ρ、u、v、E和 p 分别表示密度、速度分量、总能量和压力。该系统由状态方程闭合：

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right), \tag{2}$$

其中，γ 为比热比。

本研究采用高阶精度的 WENO格式，并对候选模板进行凸重构，同时结合保正的 HLLC（Harten-Lax-van Leer-Contact）近似黎曼求解器 [6],，以实现稳健的激波捕捉。这种组合能够在单流体黎曼问题中准确解析间断，并正确捕捉波速。在有限体积框架内重构原始变量，该框架能够自然适配交错离散，从而确保平滑且

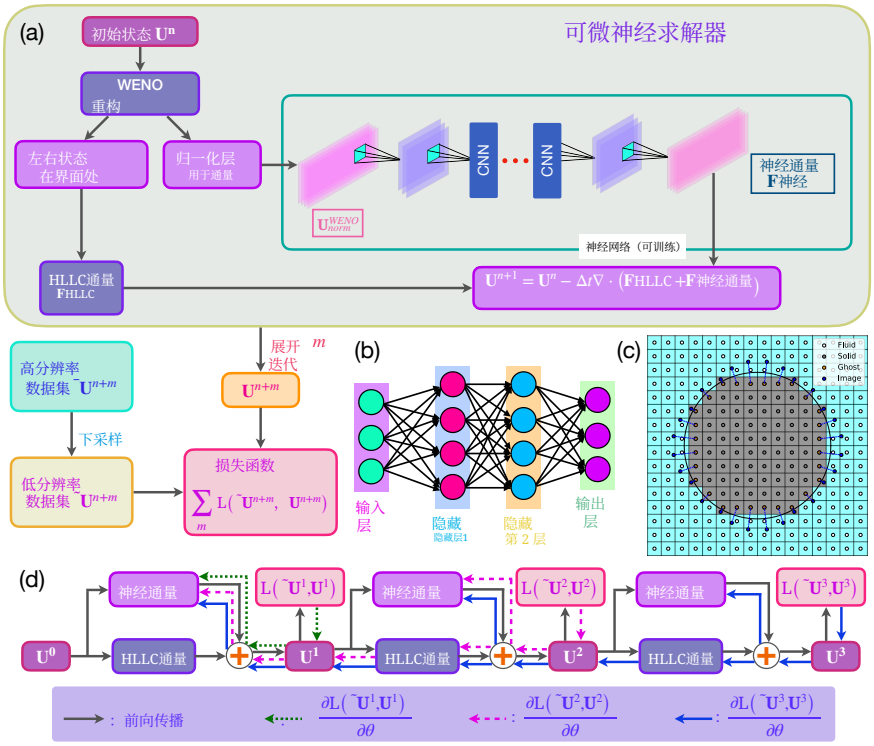

图1：我们可微框架的概览。(a) 用于二维模拟的 JAX-Shock 框架求解器工作流程，该框架集成了一系列卷积神经网络来计算神经通量。求解器展开 m 个时间步。初始状态输入框架，经WENO重构和归一化层处理后得到神经通量；随后结合 HLLC 通量与神经通量，形成最终通量。高分辨率参考数据经下采样生成低分辨率目标，并与求解器输出进行比较以计算损失。(b) 用于一维Sod激波管模拟、预测神经通量的多层感知机（MLP）网络。(c) 浸入边界法中流体、固体、虚拟和图像单元节点分类示意图。(d) 3 步设置下梯度反向传播的可视化：最终步骤的损失梯度会通过所有前序步骤及相应的网络输出进行传播。

实现稳定平流且不引入伪振荡。沿 x方向的 HLLC通量表示为

$$\mathbf{F}_x^{\mathrm{HLLC}} = \frac{1+\mathrm{sign}(s^*)}{2}[\mathbf{F}^{\mathrm{L}} + s^-(\mathbf{U}^{*\mathrm{L}} - \mathbf{U}^{\mathrm{L}})] + \frac{1-\mathrm{sign}(s^*)}{2}[\mathbf{F}^{\mathrm{R}} + s^+(\mathbf{U}^{*\mathrm{R}} - \mathbf{U}^{\mathrm{R}})], \tag{3}$$

其中，左右状态（$k = \mathrm{L}, \mathrm{R}$）通过加权本质无振荡重构获得，中间（星）状态定义为

$$\mathbf{U}^{*\mathrm{k}} = \frac{s^k - u^k}{s^k - s^*} \begin{bmatrix} \rho^k \\ \rho^k s^* \\ \rho^k v^k \\ E^k \frac{s^k - s^*}{s^k - u^k} + (s^* - u^k)\left(\rho^k s^* - \rho^k u^k \frac{s^k - s^*}{s^k - u^k}\right) \end{bmatrix}. \tag{4}$$

HLLC求解器中的波速定义为

$$s^+ = \max(0, s^{\mathrm{R}}), \quad s^- = \min(0, s^{\mathrm{L}}) \tag{5}$$

其中，左行波和右行波的波速估计为

$$s^{\mathrm{L}} = \min(u^{\mathrm{L}} - a^{\mathrm{L}}, u^{\mathrm{R}} - a^{\mathrm{R}}), \quad s^{\mathrm{R}} = \max(u^{\mathrm{L}} + a^{\mathrm{L}}, u^{\mathrm{R}} + a^{\mathrm{R}}) \tag{6}$$

其中，$a^k = \sqrt{\gamma p^k / \rho^k}$ 为声速。接触波速度计算如下:

$$s^* = \frac{p^{\mathrm{R}} - p^{\mathrm{L}} + \rho^{\mathrm{L}} u^{\mathrm{L}}(s^{\mathrm{L}} - u^{\mathrm{L}}) - \rho^{\mathrm{R}} u^{\mathrm{R}}(s^{\mathrm{R}} - u^{\mathrm{R}})}{\rho^{\mathrm{L}}(s^{\mathrm{L}} - u^{\mathrm{L}}) - \rho^{\mathrm{R}}(s^{\mathrm{R}} - u^{\mathrm{R}})}. \tag{7}$$

类似地，沿 y方向的 HLLC通量$\mathbf{F}_y^{\mathrm{HLLC}}$通过交换 x和 y 速度分量得到。随后组装得到总 HLLC通量张量:

$$\mathbf{F}^{\mathrm{HLLC}} = \begin{bmatrix} \mathbf{F}_x^{\mathrm{HLLC}} & \mathbf{F}_y^{\mathrm{HLLC}} \end{bmatrix}. \tag{8}$$

采用单元面记法，数值通量表示为

$$\mathbf{F}_x^{\mathrm{HLLC}} = \mathrm{HLLC}(\mathbf{U}^{\mathrm{L}}_{i+1/2,j}, \mathbf{U}^{\mathrm{R}}_{i+1/2,j}), \quad \mathbf{F}_y^{\mathrm{HLLC}} = \mathrm{HLLC}(\mathbf{U}^{\mathrm{L}}_{i,j+1/2}, \mathbf{U}^{\mathrm{R}}_{i,j+1/2}), \tag{9}$$

它们分别表示在单元 (i, j) 的面 $x_{i+1/2}$ 和 $y_{j+1/2}$ 上计算得到的 HLLC 通量。为进一步提高数值稳定性，守恒更新中加入了拉普拉斯型人工黏性项 $\nu\nabla^2\mathbf{U}$,

$$\mathbf{U}^{n+1} = \mathbf{U}^n - \Delta t \nabla \cdot \mathbf{F}^{\mathrm{HLLC}}(\mathbf{U}^{\mathrm{WENO}}) + \Delta t \, \nu \, \nabla^2 \mathbf{U}^n \tag{10}$$

其中，$\mathbf{U}^n$ 表示时间步 n的守恒变量，$\mathbf{U}^{n+1}$ 表示更新后的解，$\mathbf{U}^{\mathrm{WENO}}$ 表示由$\mathbf{U}^n$ 重构得到的加权本质无振荡守恒变量， Δt

时间步长，以及 ν 一个较小的数值扩散系数。该人工黏性项能有效抑制间断附近的非物理振荡，同时保持光滑区域的精度。

JAX-Shock框架有两种运行模式。第一种模式是一个完全可微的求解器，无需进行神经网络训练即可实现基于梯度的参数优化。具体而言，整个求解器均使用 JAX 实现，从而利用自动微分和即时编译实现高效的 GPU加速计算。第二种模式集成了神经通量模块，以增强低分辨率模拟，并通过端到端学习准确重构高分辨率流场。图1(a)展示了集成神经通量模块的求解器工作流程概览。

对于二维模拟，神经通量模型由一个包含三个卷积层的全卷积神经网络（卷积神经网络）参数化。前两层采用修正线性单元（ReLU）激活函数，最后一层则采用双曲正切（tanh）激活函数。该网络共有 6,388 个可训练参数，并通过“SAME”填充保持空间分辨率。这个卷积神经网络专为神经通量场的逐像素预测而设计，可确保计算域内的局部空间一致性。如图1(a)所示，神经通量模块以经过加权本质无振荡重构的归一化状态为输入，记作 $\mathbf{U}_{\text{norm}}^{\text{WENO}}$，并在每个更新步骤中参与数值通量的计算。这种形式使学习得到的通量修正能够改进基线 HLLC通量，从而提高整体解的精度。守恒更新方程表示为

$$\mathbf{U}^{n+1} = \mathbf{U}^{n} - \Delta t \nabla \cdot (\mathbf{F}^{\text{HLLC}}(\mathbf{U}^{\text{WENO}}) + \mathbf{F}^{\text{Neural}}(\mathbf{U}_{\text{norm}}^{\text{WENO}} \mid \theta)) + \Delta t \, \nu \nabla^2 \mathbf{U}^{n} \tag{11}$$

其中，$\mathbf{F}^{\text{Neural}}(\mathbf{U}_{\text{norm}}^{\text{WENO}}) : \mathbb{R}^{N_x \times N_y \times 4} \longrightarrow \mathbb{R}^{N_x \times N_y \times 4}$ 定义在大小为 $N_x \times N_y$ 的流场上，表示由神经通量算子 $\mathcal{F}^{\text{Neural}}$得到的神经通量，该算子由可学习网络权重 θ参数化：

$$\mathcal{F}^{\text{Neural}} : \mathbf{U}_{\text{norm}}^{\text{WENO}} \mapsto \mathbf{F}^{\text{Neural}}(\mathbf{U}_{\text{norm}}^{\text{WENO}} \mid \theta). \tag{12}$$

总神经通量由其各个方向分量组成，

$$\mathbf{F}^{\text{Neural}}(\mathbf{U}_{\text{norm}}^{\text{WENO}} \mid \theta) = \left[\mathbf{F}_{x}^{\text{Neural}}(\mathbf{U}_{\text{norm}}^{\text{WENO}_x} \mid \theta) \quad \mathbf{F}_{y}^{\text{Neural}}(\mathbf{U}_{\text{norm}}^{\text{WENO}_y} \mid \theta) \right] \tag{13}$$

其中，$\mathbf{U}^{\text{WENO}_{x_{\text{norm}}}}$ 和$\mathbf{U}^{\text{WENO}_{y_{\text{norm}}}}$ 分别表示沿 x 和 y 方向重构的归一化守恒变量。两个分量均由共享同一组可学习参数 θ的卷积神经网络生成。对于一维Sod激波管模拟，采用多层感知机（MLP）网络来预测神经通量，见图1(b)。如图1(c)所示，采用鬼单元浸入边界法（浸入边界法） [26]处理笛卡尔网格框架内的固体边界。通过实施滑移壁面边界条件，使法向穿透为零，同时保持

速度切向分量。计算单元分为流体单元、固体单元、鬼单元或图像单元。

所提出的神经通量算子旨在从粗网格上的模拟中恢复高分辨率流动特征。高分辨率参考数据通过块平均过程 $d(\mathbf{U}):\mathbb{R}^{N_x\times N_y\times 4}\to\mathbb{R}^{N_x\times N_y\times 4}$ 进行下采样，以构建对应的低分辨率训练目标 $\{\tilde{\mathbf{U}}^{n+m}\}$，这些目标是在展开 JAX-Shock 求解器的 m步之后得到的。在对这些展开的 m 步进行训练时，优化目标定义为预测流动状态与目标流动状态之间的均方误差：

$$\mathcal{L}(\theta)=\frac{1}{m}\sum_{s=1}^{m}\mathcal{L}_s(\theta)=\frac{1}{m}\sum_{s=1}^{m}\left\|\mathbf{U}^{n+s}(\theta)-\tilde{\mathbf{U}}^{n+s}\right\|_2^2. \tag{14}$$

其中，$\mathcal{L}_s(\theta)$ 表示求解器中间步骤的损失，$\mathbf{U}^{n+s}(\theta)$ 表示由权重为 θ 的神经通量网络参数化的求解器预测，而 $\tilde{\mathbf{U}}^{n+s}=d(\mathbf{U}^{n+s}{}_{\mathrm{HR}})$ 表示由高分辨率真实状态 $\mathbf{U}^{n+s}{}_{\mathrm{HR}}$ 得到的下采样低分辨率目标。训练过程中，通过对完整的求解器展开步骤序列进行反向传播，计算所有中间损失 $\mathcal{L}_s(\theta)$ 关于神经通量网络参数 θ 的梯度，如图1(d)所示。将求解器更新写作

$$\mathbf{U}^{n+k}(\theta)=\mathcal{S}\big(\mathbf{U}^{n+k-1}(\theta),\,\theta\big),\quad k=1,\dots,m, \tag{15}$$

其中，$\mathcal{S}$ 表示一次融入神经通量的求解器步进。随后，可利用链式法则形式化地表示中间损失相对于 θ 的梯度：

$$\frac{\partial\mathcal{L}_s}{\partial\theta}=\sum_{k=1}^{s}\frac{\partial\mathcal{L}_s}{\partial\mathbf{U}^{n+s}}\left(\prod_{j=k+1}^{s}\frac{\partial\mathbf{U}^{n+j}}{\partial\mathbf{U}^{n+j-1}}\right)\frac{\partial\mathbf{U}^{n+k}}{\partial\mathbf{F}_{k-1}^{\mathrm{Neural}}}\frac{\partial\mathbf{F}_{k-1}^{\mathrm{Neural}}}{\partial\theta}. \tag{16}$$

其中，$\mathbf{F}_k^{\mathrm{Neural}}$表示第 k [19]步的神经通量。

3. 结果

Sod 激波管问题 [27] 是评估可压缩流求解器性能的经典基准，为对流通量的构造、激波捕捉精度和波传播保真度提供了理想而严格的测试。左右状态的初始条件为

$$(\rho,u,P,\gamma)_{\mathrm{L}}^{\mathrm{T}}=(1,0,1,1.4)_{\mathrm{L}}^{\mathrm{T}},\quad(\rho,u,P,\gamma)_{\mathrm{R}}^{\mathrm{T}}=(0.125,0,0.1,1.4)_{\mathrm{R}}^{\mathrm{T}}. \tag{17}$$

如图 2(a) 所示，初始间断处会产生三种不同的波：向左传播的稀疏波、向右传播的接触间断以及向右传播的激波。随着空间分辨率 N 提高，

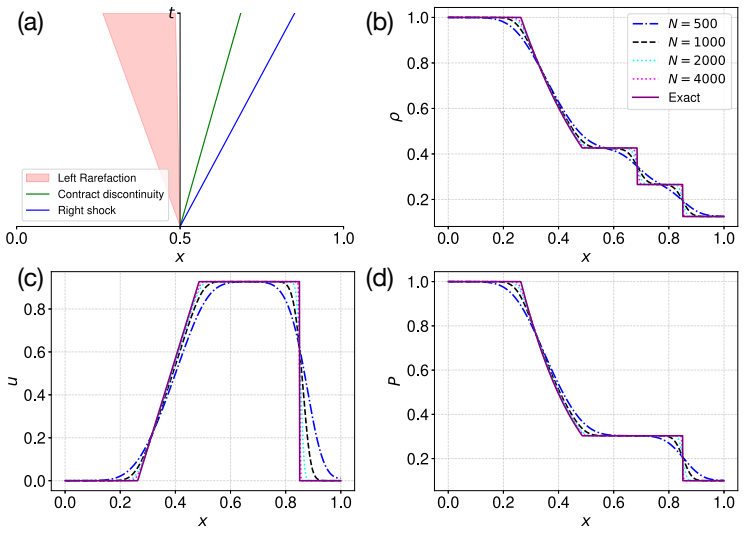

图 2： $t = 0.2$处的Sod问题：(a) 波结构、(b) 密度、(c) 速度和 (d) 压力，均由本文的神经求解器计算得到。实线表示精确解，其他线条表示不同网格分辨率下的结果。

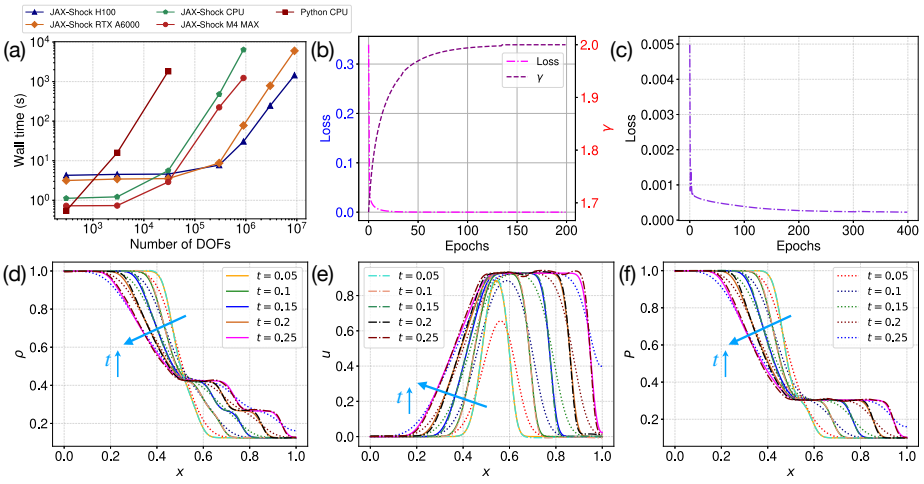

图 3：Sod激波管问题。(a) JAX-Shock的计算性能和可扩展性，展示了一维Sod问题中墙钟时间随自由度数量的变化。 (b) 比热比 γ的反向学习（优化）及相应的训练损失历史。(c) 学习神经通量算子的训练损失历史。密度 (d)、速度 (e) 和压力剖面 (f) 的时间演化：神经通量增强的低分辨率解（($N = 500$，点划线）、高分辨率参考解（$N = 1000$，实线）以及未使用神经通量的基线低分辨率解（$N = 500$，点线）。

数值解逐渐收敛至精确解。此外，计算所得的解与解析解高度吻合，如图 2(b)–(d) 所示。

我们在多种计算架构上，针对不同网格分辨率下的 Sod问题，对 JAX-Shock 的性能和可扩展性进行了基准测试。这些架构包括运行 Windows 操作系统的 Intel(R) Core(TM) Ultra 7 165U CPU @ 2.05GHz、运行 macOS Sequoia 15.5 的 Apple M4 Max GPU（40 核，配备 128GB 统一内存），以及运行 Ubuntu 22.04.5 LTS 的 NVIDIA GPU（显存分别为 48 GB 的 RTX A6000 和 94 GB 的 H100 Hopper NVL）。图 3(a) 展示了墙钟时间随自由度（DOFs）数量的变化。与在 CPU 上执行相比，JAX-Shock 在 GPU 上展现出显著的性能优势。规模最大的算例包含 9×10^6 个自由度，在 RTX A6000 上耗时 5992 s，在 H100 上耗时 1445 s，因此 H100 相较于 A6000 的加速比为 4.1×。相比之下，CPU 和 Apple M4 MAX 配置下可行的最大算例分别包含 9×10^5 个自由度，耗时分别为 6375 s 和 1234 s。作为对比，一个使用显式 for 循环（不使用 JAX）的朴素 Python 实现，在 CPU 上处理仅有 3×10^4 个自由度的算例就耗时 1822 s。相较于这一基线，H100 和 A6000 的加速比分别为 396.5× 和 515.1×。对于包含 9×10^5 个自由度的模拟，与基于 JAX 的 CPU 和 M4 MAX 实现相比，H100 的加速比分别为 $208\times$ 和 40.2× 。图 3(b) 展示了通过反向学习（优化）过程，利用 Sod激波管数据推断比热比 γ 的过程，学习率恒定为 0。1。在该实验中，γ 被设为可微求解器中的可学习参数，并通过基于梯度的优化反复更新，以最小化预测解与参考解之间的 L_2 损失。图中展示了估计的 γ 向其真实值演化的过程及相应的训练损失历史，表明该过程收敛迅速，并证实 JAX-Shock 能够直接从流场数据中恢复底层物理参数。JAX-Shock 中的神经通量模块使用包含 $N = 500$ 个网格点的低分辨率模拟进行训练。反向学习模型和神经通量模型均使用 Adam优化器进行训练。如图 3(c) 所示，采用学习率为 10^{-3} 的恒定学习率计划时，训练损失迅速收敛。加入神经通量模块后，低分辨率模拟前向传播得到的密度、速度和压力剖面随时间的演化，与高分辨率参考结果（$N = 1000$）高度吻合，并且显著优于基线低分辨率模拟（$N = 500$），如图 3(d)–(f) 所示。研究还评估了模型对未见未来时刻的泛化能力。值得注意的是，神经通量模型仅训练至最终时刻 $t = 0.2$，但 JAX-Shock 外推至 $t = 0.25$ 时仍能保持准确预测，如图 3(d)–(f) 所示。

在下一个实验中，我们模拟移动法向激波与半径为 0.25、中心位于 (0.6,1) 的圆柱体之间的相互作用。模拟在二维计算域 $(x, y) \in [-0.5, 4] \times [0, 2]$ [7] 内进行。流场初始化为具有高密度和高压的上游值，

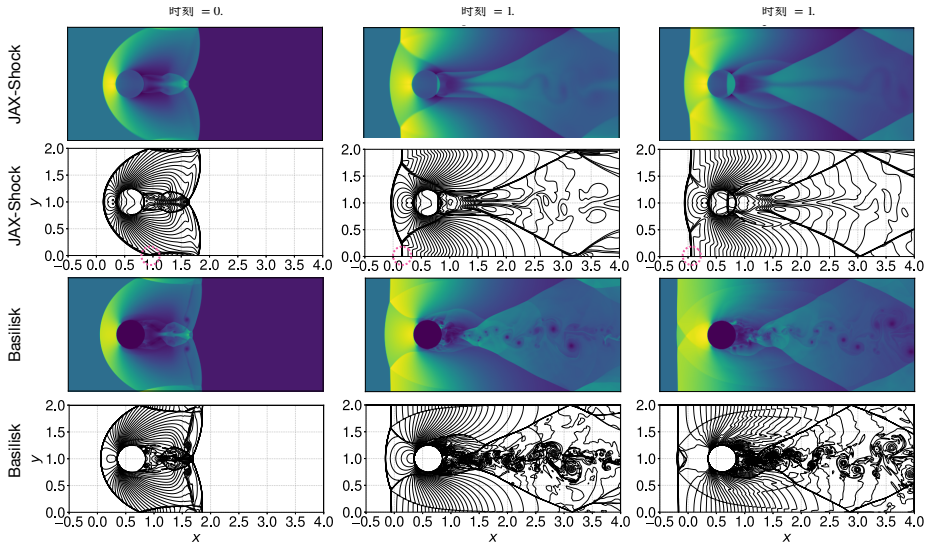

图4：激波与圆柱体的相互作用：比较 JAX-Shock 和 Basilisk 在三个时刻 $t = 0.3$、1.0 和 1.2 计算得到的密度场。每组面板均显示彩色等值线（上）和黑色线等值线（下），等值线级别为 32 个，范围从 0.8 到 18。

形成速度为 5 的尖锐正激波，其位置在圆柱体左侧；下游区域则设为低密度、低压条件，具体如下：

$$\{\rho, u, v, p, \gamma\} = \begin{cases} \{7, 4, 0, 29, 1.4\} & -0.5 \leq x < 0.3, \\ \{1.4, 0, 0, 1, 1.4\} & 0.3 \leq x \leq 4. \end{cases} \tag{18}$$

左边界设为与初始流场左侧状态一致的流入条件，右边界设为流出条件。顶部和底部边界视为反射壁面。图4展示了使用 JAX-Shock 和 Basilisk [28]在三个时刻 $t = 0.3$、1.0 和 1.2 获得的密度场。Basilisk 模拟采用 $L12$ 网格分辨率且不进行网格自适应，共包含 $\sim 7.3 \times 10^6$ 个单元；JAX-Shock 计算则使用包含 $\sim 8.4 \times 10^6$ 个单元的均匀网格。两种求解器均捕捉到了入射激波与圆柱体之间的复杂相互作用，包括入射激波、反射激波和透射激波的形成，以及随后出现的尾流动力学。在 $t = 1.0$ 和 $t = 1.2$时，沿 x轴对称地产生了两个清晰界定的激波三重点，在两种解中均清晰可见。然而，与 Basilisk 相比，JAX-Shock 中圆柱体后的涡脱落不太明显，尽管前者采用了高阶 WENO5重构和 HLLC通量格式，而 Basilisk 使用二阶 Bell-Collela-Glaz无向格式和 minmod斜率限制器。尽管 WENO5 在形式上具有五阶精度

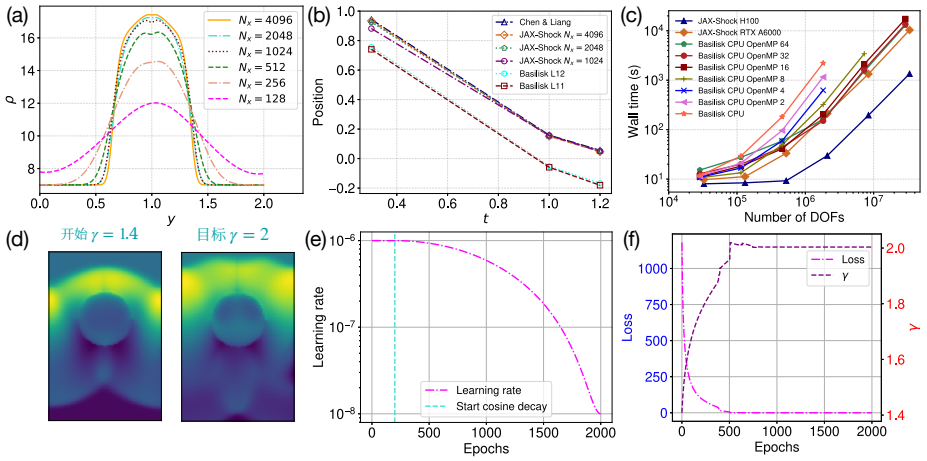

图 5：移动正激波与圆柱体的相互作用。(a) 基于密度沿横向方向在 $x = 0.2$ 和 $t = 0.3$处的线剖面进行的收敛性研究。(b) 通过比较入射激波与底壁交点的时间演化（见图4中的虚线圆），并在多个网格分辨率下与 Basilisk 和参考数据 [7]进行基准对比，验证 JAX-Shock。(c) JAX-Shock 的计算性能与可扩展性，展示了测得的激波-圆柱相互作用墙钟时间随自由度数量的变化。(d) 密度场的受限反向学习域，展示初始猜测 ($\gamma = 1.4$) 和目标值 ($\gamma = 2$)。(e) 反向学习过程中使用的学习率退火计划。(f) 比热比 γ 的反向学习（优化）及相应的训练损失历史。

在平滑区域，JAX-Shock 引入的额外人工耗散可能会抹平涡量层，进而削弱细尺度尾迹结构。

如图 5(a) 所示，随着网格加密，密度分布逐渐收敛。图 5(b) 显示，JAX-Shock 预测的入射激波与底壁交点的时间演化，在所有加密网格上都与参考数据吻合良好，而 Basilisk 的结果则存在明显偏差。我们评估了 JAX-Shock 在 GPU 上的计算性能和可扩展性，并将其与采用 OpenMP 并行化、运行于 CPU 上的 Basilisk 进行比较。JAX-Shock 运行于 Ubuntu 22.04.5 LTS 系统，使用 2.0 GHz Intel(R) Xeon(R) Gold 6438Y+ CPU（32 个核心）和 NVIDIA GPU（RTX A6000，48 GB；H100 Hopper NVL，94 GB）。采用 OpenMP 的 Basilisk 则运行在相同的 CPU 平台上。图 5(c) 展示了墙钟时间随自由度数量的变化。对于规模最大的 $\sim 3.4 \times 10^7$ 个自由度问题，RTX A6000 耗时 10354 s，而 H100 仅用 1350 s 即完成同一算例，提速 7.6$\times$ 倍。相比之下，Basilisk 的 OpenMP 加速在自由度数量较大时效果有限：对于一个包含 2.9×10^7 个自由度的模拟，将 CPU 核心数从 16 增至 64，提速仅为 1.3$\times$ 倍。对于规模相当的问题，H100 和 A6000 相较于使用 64 个核心的 Basilisk OpenMP，分别实现了 9.6$\times$ 倍和 1.2$\times$ 倍的提速。相较于 Basilisk CPU 算例（含 1.8×10^6 个自由度），H100 和 A6000 在包含 2.1×10^6 个自由度的模拟中分别实现了 74.4$\times$ 倍和 10.5$\times$ 倍的提速，尽管它们处理的问题规模更大。图 5(d) 展示了受限训练域，其定义为 $(x,\ y) \in [0, 1.5] \times [0, 1]$，包含 8192 个单元。该子域用于比热比 γ 的反向学习：从初始猜测出发，通过迭代逐渐收敛至目标值。为加速优化并提升稳定性，我们采用了基于预热余弦衰减调度的学习率退火策略，见图 5(e)。在以恒定学习率 10^{-6} 训练 200 轮后，启动余弦衰减，将学习率从 10^{-6} 降至 10^{-8}。如图 5(f) 所示，在该退火调度下，训练损失迅速下降，表明反向学习过程稳定且高效地收敛。

神经通量模型采用分段常数学习率调度进行训练：前 3500 次迭代的初始学习率为 10^{-4}，之后在剩余训练过程中将学习率降低至 10^{-5}。训练损失历史如图 6(a) 所示，损失逐渐下降，并在经过多个训练轮次后收敛。图 6(b) 对比了学习得到的神经通量模型预测的密度场随时间的演化与真实值及基准求解器的结果。在与基准求解器相同的分辨率（128×64）下，神经通量模型能够准确捕捉尖锐激波，同时也与真实解（256×128）高度吻合。

神经通量模型训练至最终时刻 $t = 0.2$。图 7 对比了沿 y 方向预测的密度、速度和压力剖面随时间的演化与参考解及基准结果。在直线 $x = 0.29$ 上的 $t = 0.18$ 和 $t = 0.2$ 处（该直线在图 6(b) 中以红色虚线标出），神经通量模型的预测与真实值更加吻合，而基准结果则存在明显偏差。

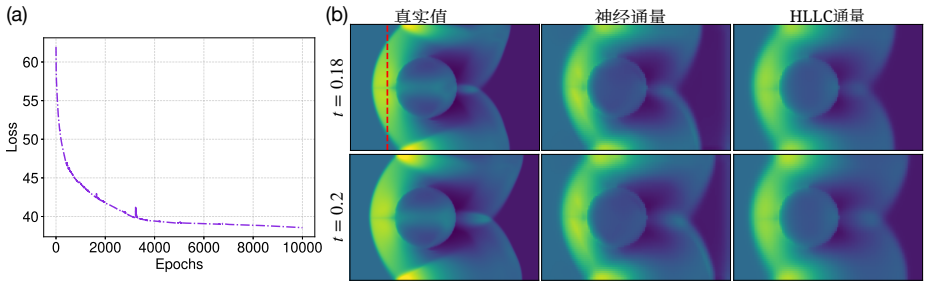

图 6：用于模拟移动正激波与圆柱体相互作用的学习得到的神经通量算子。(a) 学习神经通量算子的训练损失历史。(b) 参考求解器（256×128）、学习得到的求解器（128×64）和基线求解器（128×64）预测的密度场演化。

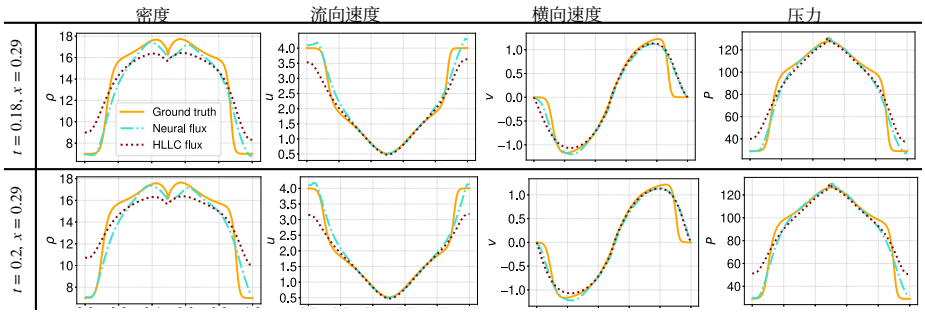

图 7：比较沿 y 方向学习得到的预测密度、速度和压力剖面的时间演化与参考解及基准结果。

4. 结论

本研究推出了 JAX-Shock，这是一种完全可微、GPU 加速的高阶激波捕捉神经求解器，可用于涉及复杂激波-固体相互作用的可压缩流。该求解器在 JAX 自动微分框架内集成了 WENO重构、HLLC近似黎曼求解器和浸入边界法，从而能够端到端地通过非线性激波动力学和固-流界面传播梯度——这些能力对于传统计算流体力学工具而言仍难以实现。

该求解器在 Sod问题和移动激波冲击圆柱体这两种情况下均实现了高保真度。在多种硬件架构上进行的基准测试表明，该求解器在现代 GPU 上具有出色的加速效果和可扩展性，其中 NVIDIA H100 的整体性能最佳。

该框架的一项关键贡献是神经通量模块，它通过数据驱动修正增强数值通量，同时保留底层激波捕捉格式的稳定性和物理结构。这种混合形式显著提升了粗网格预测精度，无需承担精细网格模拟的全部成本，即可恢复高分辨率特征。该求解器支持多步可微展开、序列

-

序列学习和反向模式推断，从而为强间断主导的情形下的参数估计与设计优化提供统一途径。

结果表明，JAX-Shock 在典型激波问题和具有挑战性的激波-固体构型中均能实现高保真度，同时还具备纯数据驱动方法或传统纯物理方法所不具备的显著灵活性。该框架为新一代可微流体求解器奠定了基础，使其能够将基于物理的模拟与现代机器学习方法相结合。

致谢

本研究得到了北伊利诺伊大学研究与创新合作承诺部（RIPS）的资助。

数据可用性

支持本研究结论的数据可在提出合理请求后向通讯作者索取。

参考资料

参考文献

[1] P Tembhurnikar、MD G Sarwar 和 D Sahoo。《武器舱导弹弹射在不同马赫数下的计算流体动力学分析》。流体动力学，60(1):14，2025。

[2] R Kumar 和 A Ghosh。支板稳定超燃冲压发动机燃烧室中隔离段激波对燃料流量调制的不稳定性。《航空学杂志》，129 (1331):42–62，2025。[3] Fengshou Xiao、Zhufei Li、Zhiyu Zhang、Yujian Zhu 和 Jiming Yang。V形钝头前缘上的高超声速激波相互作用。《AIAA杂志》，56(1):356–367，2018。[4] Richard Edward Olson、RJ Leeper、A Nobile 和 JA Oertel。预热效应对间接驱动惯性约束聚变消融层材料中激波传播的影响。《物理评论快报》，91(23):235002，2003年。[5] Bernhard Müller。核心坍缩超新星及其前身星的流体力学。《计算天体物理学生活评论》，6(1):3，2020年。[6] E. Johnsen 和 T. Colonius。在可压缩多组分流动问题中实现加权本质无振荡格式。《计算物理学杂志》，219:715–732，2006年。[7]Kuangxu Chen 和 Chunlei Liang。面向非结构曲面四边形网格上双曲守恒律的高阶混合本质无振荡谱差分方法。《流体物理学》，37(7):076107，2025。[8] Bo Zhang、Bradley Boyd 和 Yue Ling。使用质量—动量—能量一致的流体体积法对可压缩界面多相流进行直接数值模拟。《计算流体》，236:105267，2022年。[9] Bo Zhang。用于预测翼型周围相干结构演化的基于翼型的卷积自编码器和长短期记忆神经网络。《计算流体》，258:105883，2023年。[10] Bo Zhang。通过物理同化卷积自编码器对翼型上方非定常流进行非线性模态分解。《流体物理学》，35(9):095115，2023年。[11] Bo Zhang。用于纳维–斯托克斯方程的巴拿赫神经算子。《流体物理学》，37(8):086166，2025。[12] Mike Innes、Alan Edelman、Keno Fischer、Chris Rackauckas、Elliot Saba、Viral B Shah 和 Will Tebbutt。连接机器学习与科学计算的可微编程系统。arXiv预印本 arXiv:1907.07587，2019年。[13]Maziar Raissi、Paris Perdikaris 和 George E Karniadakis。物理信息神经网络：用于求解涉及非线性偏微分方程的正向与逆向问题的深度学习框架。《计算物理学杂志》，378:686–707，2019年。

[14] Martín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Isard 等。《TensorFlow：一种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*），第265–283页，2016年。[15] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga 等。《PyTorch：一种命令式风格的高性能深度学习库》。神经信息处理系统进展，第32卷，2019年。[16] James Bradbury、Roy Frostig、Peter Hawkins、Matthew J. Johnson、Chris Leary、Dougal Maclaurin、George Necula、Adam Paszke、Jake VanderPlas、Skye Wanderman-Milne 和 Qiao Zhang。《JAX：可组合的 Python+NumPy 程序转换》。http://github.com/google/jax，2018年。[17] Justin Sirignano、Jonathan F MacArt 和 Jonathan 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》，423:109811，2020年。[18] Dmitrii Kochkov、Jamie A Smith、Ayya Alieva、Qing Wang、Michael P Brenner 和 Stephan Hoyer。《机器学习加速计算流体力学》。《美国国家科学院院刊》，118(21):e2101784118，2021年。[19] Björn List、Li-Wei Chen 和 Nils Thuerey。《采用可微流体求解器的学习型湍流建模：基于物理的损失函数与优化视界》。《流体力学杂志》，949:A25，2022年。[20] Deniz A Bezgin、Aaron B Buhendwa 和 Nikolaus A Adams。《JAX-Fluids：一种用于可压缩两相流的全可微高阶计算流体力学求解器》。《计算物理学通讯》，282:108527，2023年。[21] Bo Zhang、Xiantao Fan 和 Jian-Xun Wang。《一种用于高效模拟空化流的可微混合神经求解器》。载于《美国物理学会通报》，美国华盛顿特区，2023年。APS。[22] Tianju Xue、Shuheng Liao、Zhengtao Gan、Chanwook Park、Xiaoyu Xie、Wing Kam Liu 和 Jian Cao。《JAX-FEM：一种用于自动逆向设计和机制数据科学的可微 GPU 加速三维有限元求解器》。《计算物理学通讯》，291:108802，2023年。[23] Samuel S Schoenholz 和 Ekin D Cubuk。《JAX MD：一个可微物理学框架》。《统计力学杂志：理论与实验》，2021(12):124016，2021年。[24] Bo Zhang、Wenjie Shang、Jyoti Panda、Jiahang Zhou、Jianxun Wang 和 Tengfei Luo。《JAX-BTE：一种用于纳米电子学深度学习加速热建模的可微混合神经求解器》。载于*ASME 2024*年夏季传热会议，美国加利福尼亚州阿纳海姆，2024年。

[25] Li Li、Stephan Hoyer、Ryan Pederson、Ruoxi Sun、Ekin D Cubuk、Patrick Riley 和 Kieron Burke。《将 Kohn-Sham 方程作为正则化器：将先验知识融入机器学习物理学》。《物理评论快报》，126(3):036401，2021年。[26] R. Mittal 和 G. Iaccarino。《浸入边界方法》。《流体力学年度评论》，37:239–261，2005年。[27] Gary A Sod。《非线性双曲守恒律方程组的若干有限差分方法综述》。《计算物理学杂志》，27(1):1–31，1978年。[28] S. Popinet。《一种用于表面张力驱动界面流的精确自适应求解器》。《计算物理学杂志》，228(16):5838–5866，2009年。