arXiv:2602.08670v1 [cs.CV] 9 Feb 2026

Technische Universität Berlin

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

SPCL

硕士论文

基于机器学习加速的地球物理流体求解器

白杨 学号: 459272 2022年11月30日

在柏林工业大学数学研究所科学计算硕士项目中完成

导师：TorstenHÃűfler 教授苏黎世联邦理工学院高性能计算系统研究所，瑞士

JÃűrgLiesen 教授柏林工业大学数学研究所，德国

Technische Universität Berlin

Declaration of originality

Title of work (in block letters):

A Machine Learning accelerated geophyysical fluid solver

Authored by:
For papers written by groups the names of all authors are required.

Name(s): Bai **First name(s):** Yang

I hereby declare that the thesis submitted is my own, unaided work, completed without any unpermitted external help. Only the sources and resources listed were used.

Place, date **Signature(s)**

Berlin, 30.11.2022 Yang Bai

摘要

机器学习方法已在许多领域取得成功，例如图像分类和自然语言处理。然而，如何将机器学习应用于具有数学约束的领域（如求解偏微分方程），仍有待探索。在将机器学习技术用于求解偏微分方程的各种方法中，数据驱动离散化方法展现出良好的前景：它能够在结构化网格上预测拟线性模板的系数，以计算函数在给定位置处的值或导数，从而加速并改进现有的偏微分方程求解器。与传统的有限差分或有限体积格式相比，该方法能够提高低分辨率模拟的精度和稳定性。同时，它还可以借鉴传统数值格式的优势，例如采用有限体积类型的公式来实现守恒律。

在本学位论文中，我们在另一种框架下实现了浅水方程和欧拉方程的经典求解器。实验表明，我们的经典求解器性能远优于 Pyclaw 求解器。随后，我们为基于机器学习的求解器提出了四种不同的深度神经网络。结果表明，其中两种方法能够给出令人满意的解。

摘要

机器学习方法已在许多领域得到验证，例如分类和自然语言处理。然而，如何将机器学习应用于具有数学约束的领域（如偏微分方程的求解），仍有待厘清。在各种将机器学习技术应用于偏微分方程求解的方法中，数据驱动离散化方法是加速和改进结构化网格上现有偏微分方程求解器的一种很有前景的途径。该方法通过预测准线性模板的系数，计算函数在特定位置处的值或导数。因此，与传统的有限差分法或有限体积法相比，它可以提高低分辨率模拟的精度和稳定性。此外，该方法也能受益于传统数值格式，例如通过调整有限体积类型的公式来实现守恒定律。

本文在不同的框架下实现了浅水方程和经典欧拉方程求解器。实验表明，经典求解器的表现远优于 Pyclaw 求解器。因此，本文为基于机器学习的求解器提出了四种不同的深度神经网络。结果表明，其中两种方法能够提供令人满意的解。

目录

1 引言 **7**

2 理论背景 **7**

2.1 浅水方程 .82.2 欧拉方程 .92.3 双曲方程的有限体积法 .92.3.1 离散守恒 .102.3.2 有限体积法格式的数值通量 .112.3.3 二阶格式 .142.3.4 双曲方程的算法 .142.4 球面上的逻辑矩形网格 .162.4.1 圆盘映射 .162.4.2 单位球面的映射 .162.5 机器学习加速计算流体力学 .17

2.5.1 空间导数的数据驱动近似 .172.5.2 数值通量的数据驱动近似 .182.5.3 边界状态的数据驱动近似 .18

3 经典 S 浅水方程求解器 2 **1**

3.1 一维浅水方程 .213.1.1 一阶求解器 .213.1.2 二阶求解器 .223.2 平面上的浅水方程 .233.2.1 不含源项的二阶求解器 .243.2.2 含源项的二阶求解器 .243.3 球面上的浅水方程 .253.4 不同框架下的计算开销 .27

4 经典解答 欧拉方程 **29**

4.1 一维欧拉方程 .294.1.1 一阶求解器 .294.1.2 二阶求解器 .304.2 平面欧拉方程 .31

5 浅水方程的嵌入式数据驱动求解器 **33**

5.1 使用卷积神经网络直接计算边界通量 .335.2 使用卷积神经网络生成的线性系数计算边界状态 . . . . . 355.3 使用卷积神经网络直接计算边界状态 .375.4 使用卷积神经网络生成的线性系数计算重构斜率. 42

6 结论 ns 与未来工作 4 **6**

6.1 结论 .46

6.2 未来工作.46

致谢

首先，我谨向我的导师和指导教师 Prof. Dr. Torsten Hoefler、Langwen Huang 致以诚挚的谢意。过去几个月里，感谢你们不仅在项目指导方面给予我支持，也关心我在 ETH 苏黎世的生活。你们引领我进入地球物理流体和深度学习的世界。尽管这个项目中仍有许多尚未充分探索的领域和不确定性，但你们在会议中提出的启发性想法，甚至提供的实用工程指导，都让我受益匪浅。没有你们的支持，这篇学位论文将不可能完成。

我谨向柏林工业大学的 Prof. Dr. JÃűrg Liesen 致谢，感谢他对学位论文格式提出的建议并进行修改。

我也谨向 ETH 苏黎世的 Prof. Dr. Siddhartha Mishra 致谢，感谢他分享改进我们工作的想法。

我谨向所有亲爱的朋友表达谢意，感谢他们的帮助与支持，陪伴我度过了在柏林工业大学和苏黎世联邦理工学院最艰难的时光。

最后但同样重要的是，我要感谢我的父母，感谢你们无条件的爱与支持。没有你们，就不会有今天的我。你们支持我出国留学，并在我需要时给我最温暖的港湾。

符号与缩略语

ρ 密度

p 压力

E 能量$h(x,t)$ 水深$z(x,y)$ 水床

q 物理量向量g 重力常数 9.8ω 地球自转速率 $7.292 \times 10^{-5}a$ 地球半径 $6.37122 \times 10^{6}S(x,q)$ 源项$L(q)$ 空间导数Q 解 q 的数值近似

F 通量函数R 旋转矩阵 $\hat{}F$ 数值通量格式$_F$ 通量的数值近似$\vec{x}$ 笛卡尔坐标向量$\vec{u}$ 笛卡尔速度分量向量x, y, z 笛卡尔坐标 $u(x,t)$ $v(y,t)$ $w(z,t)$ 笛卡尔速度分量

t 时间

ΔT 时间步长Δc 单元面积 Ω 区域体积$\partial\Omega$ 的边界 RK 方法 Runge–Kutta 方法 机器学习 神经网络 计算流体力学 卷积神经网络 有限体积法 浅水方程 周期性边界条件 总变差递减 修正线性单元 DACE 以数据为中心的并行编程 均方根误差

1 引言

由非线性偏微分方程描述的复杂物理系统的模拟，在工程学和物理科学中占有重要地位。浅水方程可用于模拟海洋和大气中的流体流动 [1]。此类系统的模型能够预测最终会受到污染、海岸侵蚀以及极地冰盖融化影响的区域。欧拉方程可应用于许多领域，包括气体动力学、流体力学和环境科学 [2]。

尽管经典求解器十分精密，但在模拟气候、天气和海啸时，针对这些问题所需尺度进行直接数值模拟是不可能的。传统方法采用较粗的网格，但这会牺牲精度。尽管机器学习的应用尚处于起步阶段，但利用机器学习加速计算流体力学问题正受到越来越多的关注。已有研究提出了一种机器学习框架，用于加速与时间相关的常微分方程和偏微分方程的数值计算（Mishra, 2018） [3]；Bar-Sinai 等人（2019）使用基于有限体积法的学习型数据驱动离散方法求解伯格斯方程，并通过实验表明，进行 4–8 倍的尺度扩展仍具有很高的精度 [4]。Magiera 等人（2020）将约束感知神经网络应用于黎曼问题 [5]。

Maddu（2021）提出了 STENCIL-NET 架构，用于对未知非线性偏微分方程进行数据驱动的自适应离散化 [6], 。该架构是一种类似加权本质非振荡（WENO）的解自适应数值格式 [7]。一种深度卷积神经网络（卷积神经网络）被用于预测全球网格上的若干关键大气变量（Weyn 等人，2020） [8]。Kochkov 等人（2021）通过将二维湍流中的网格规模扩大十倍，使计算时间提升了 80 倍 [9]。

据我们所知，目前尚无针对嵌入式数据驱动浅水方程求解器的研究。本文其余部分安排如下：第 2 章介绍欧拉方程和浅水方程，并阐述有限体积离散方法。随后简要介绍数据驱动方法，并介绍后续章节所需的球面网格。第 3 章和第 4 章分别在 torch 和 DACE 框架中开发浅水方程与欧拉方程的经典求解器，并将其与 Pyclaw 中的黎曼问题求解器进行比较。第 5 章使用四种方法实现了基于机器学习的浅水方程求解器（卷积神经网络）。机器学习方法的技术细节（例如神经网络结构）见第 2 章。此外，每种方法均进行总能量和位涡分析。最后，第 6 章总结全文并讨论未来工作。

2 理论背景

下一章分为五个部分，介绍后续实验中开发数值格式所需的理论背景。第一和第二部分将简要介绍浅水方程和欧拉方程。它们会出现间断现象，例如黎曼问题中的激波，其解与双曲性密切相关。第三部分简要推导双曲方程的有限体积法，并介绍后续实验中使用的一些数值格式。第四部分介绍球面浅水方程求解器所使用的网格。最后一部分将简要介绍数据驱动方法及其在偏微分方程求解中的应用。

2.1 浅水方程

当垂直速度和水面位移相对于波频率足够小时，在假设静水压力分布、水平速度沿垂直方向均匀分布且底坡较小的条件下，可以对纳维–斯托克斯方程沿水深积分，得到浅水流的控制方程。忽略由黏性和湍流引起的扩散项、科里奥利项以及表面剪切应力项后，浅水方程可表示为向量形式 [10]:

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=S, \tag{2.1}$$

其中， t 表示时间； x 和 y 是水平面上的笛卡尔坐标； Q表示守恒变量向量； F 和 G分别是沿 x和 y 方向的对流通量向量； S 表示源项，它还可进一步分为底坡源项 S_b 和摩擦源项 S_f (2.5)。

我们首先考虑单空间维度的情形。设 $h(x,\ t)$为河流的水位（水深）， $u(x,\ t)$为平均速度。设河床为平面。由守恒律可得一维浅水方程:

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hu \end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix} hu \\ hu^2+\frac{1}{2}gh^2 \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{2.2}$$

其中， g 为重力常数。该方程组体现了质量和动量守恒，并以水位为变量表示。如果河床具有由光滑函数 $S(x)$描述的剖面，则可得到带有源项的守恒律

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hu \end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix} hu \\ hu^2+\frac{1}{2}gh^2 \end{pmatrix}=\begin{pmatrix} 0 \\ -ghS'(x) \end{pmatrix}.$$

对于平面河床，有 $S(x)\ \equiv$const。因此，上述守恒律便可恢复。

在二维空间的情况下，河床由剖面 $S(x,y)$ 指定，而 $h(x,y,t)$ 表示水位。设 $u(x,y,t)$ 和 $v(x,y,t)$ 分别为沿x 和 y方向的平均速度。我们得到二维浅水方程的一个守恒律，其源项为

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hu \\ hv \end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix} hu \\ hu^2+\frac{1}{2}gh^2 \\ huv \end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix} hv \\ huv \\ hv^2+\frac{1}{2}gh^2 \end{pmatrix}=\begin{pmatrix} 0 \\ -gh\frac{\partial z}{\partial x}(x,y) \\ -gh\frac{\partial z}{\partial y}(x,y) \end{pmatrix}, \tag{2.3}$$

其中包含两个通量函数。该系统关于 x 和 y 对称。对于平面河床 $S(x,y)\ \equiv$ const.，可得到一个守恒律系统。可以证明该系统是双曲型的。若考虑摩擦源项 S_f ，则源项表示如下:

$$S=S_b+S_f=\begin{pmatrix} 0 \\ -gh\frac{\partial z}{\partial x}(x,y) \\ -gh\frac{\partial z}{\partial y}(x,y) \end{pmatrix}+\begin{pmatrix} 0 \\ -C_f u\sqrt{u^2+v^2} \\ -C_f v\sqrt{u^2+v^2} \end{pmatrix},$$

其中， C_f 是通过 $C_f=gn^2/h^{1/3}$， n计算得到的河床糙率系数， n表示曼宁糙率系数。

2.2 欧拉方程

欧拉方程描述了无黏可压缩流的动力学，是最著名的非线性双曲守恒律实例之一。欧拉方程基于质量守恒、线性动量守恒和总能量守恒建立。这里讨论的本质上是牛顿第二运动定律和热力学第一定律。考虑占据固定一维情形的流体，并分别用 ρ、 u和 E表示其密度、速度和比内能。请注意，这些量描述的是固定空间坐标和固定时间处的流体。如果 u不仅取决于（质量）密度，还取决于其他量，那么必须加入其他守恒量，才能构成未知量与方程数量相等的方程组。例如，我们将 [11],排列为

$$\begin{aligned}(\rho u)_t + \left(\rho u^2 + p\right)_x &= 0 \quad \text{(conservation of momentum)},\\ E_t + (u(E+p))_x &= 0 \quad \text{(conservation of energy)},\end{aligned}$$

其中， ρu 为动量密度， p为压力。根据气体动力学的物理定律，压力必须表示为 ρ， ρu，E的函数。例如，理想气体满足

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho u^2\right), \tag{2.4}$$

其中常数 $\gamma \in \mathbb{R}$ 取值如 $\gamma = \frac{5}{3}$。由此得到气体动力学的欧拉方程，它由三个守恒律组成。守恒量为

$$q(x,t) = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} := \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}.$$

如果忽略源项，则一维欧拉方程可表示为如下守恒律，

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.5}$$

对于压力，在式（2.4）的情况下，有 $p(q_1,\ q_2,\ q_3) = (\gamma - 1)\left(q_3 - \frac{1}{2}q_2^2/q_1\right)$。

2.3 双曲方程的有限体积法

本章将介绍用于标量守恒律的高效格式。即使在线性输运这一最简单的情形下，也不能使用中心差分来近似守恒律。设计线性输运方程高效格式的关键步骤，是沿信息传播的方向取导数，从而采用迎风格式。对于像 $q_t + aq_x = 0$ 这样的具有恒定系数的线性方程，信息传播方向由恒定速度场给出。对于像 $q_t + F(q)_x = 0$ 这样的非线性守恒律，波速取决于解本身，无法先验确定。因此，如何采用迎风差分并不明确。有限差分格式的核心思想，是用有限差分替代诸如 $q_t + F(q)_x = 0$ 这样的方程中的导数。这一过程要求解是光滑的，并且方程逐点满足 [2]。然而，标量守恒律的解不一定光滑，因此，用有限差分替代导数所必需的泰勒展开不再成立。因此，有限差分框架并不适合用于近似守恒律。相反，我们需要为标量守恒律的数值格式设计开辟一种新的范式。

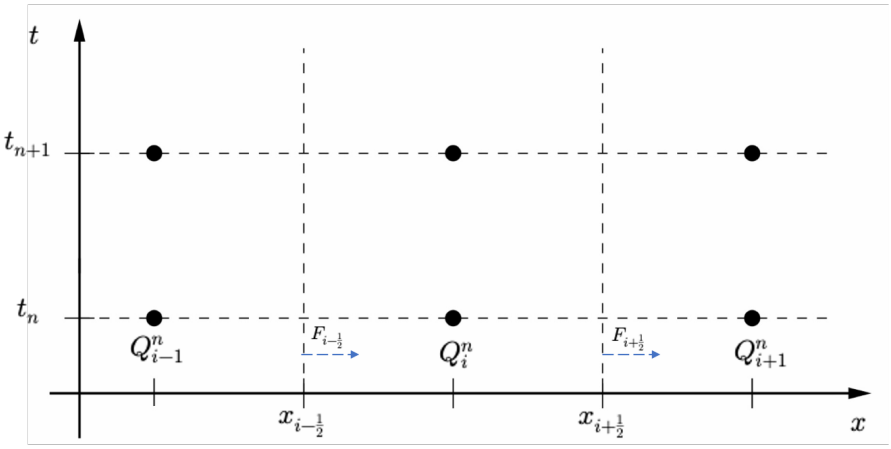


图 1：典型的有限体积网格，显示了单元平均值和通量。

2.3.1 离散守恒

任何数值近似的第一步，都是对空间和时间上的计算域进行离散化。

网格

为简单起见，我们考虑区域 $[x_L, x_R]$ 的均匀离散。离散点记作 $x_i = x_L + (i + 1/2)\Delta x$，其中 $i = 0$、.....、N，且 $\Delta x = \frac{x_R - x_L}{N+1}$。我们还定义中点值

$$x_{i-1/2} = x_i - \Delta x/2 = x_L + i\Delta x.$$

F或 $i = 0$、.....、$N + 1$。这些值定义了计算单元或控制体 s

$$\Omega_i = \left[x_{i-1/2}, x_{i+1/2}\right).$$

我们很快就会看到，有限体积法使用控制体 Ω_i，而非网格点 x_i。我们采用时间步长为 Δt 的均匀时间离散。时间层记作 $t^n = n\Delta t$。网格示意图见图 1。

单元平均值

有限差分法基于对偏微分方程解的点值进行近似。由于解并不连续，点值可能没有意义，因此这种方法不适用于守恒律。相反，我们改变视角，改用单元平均值：

$$Q_i^n \approx \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} q\left(x, t^n\right) dx, \tag{2.6}$$

在每个时间层 t^n 将其作为我们近似所关注的主要对象。单元平均值对任意可积函数都有明确定义，因而对守恒律$q_t + F(q)_x = 0$的解也同样适用。有限体积法的目的是在每一个时间步长更新未知量的单元平均值。

守恒律的积分形式

守恒律的积分形式。假设某一时间层t^n 的单元平均值 Q_i^n 已知。如何求得下一时间层 t^{n+1}的单元平均值 Q_i^{n+1}？一种有限

体积法通过在区域 $\left[x_{i-1/2}, x_{i+1/2}\right) \times [t^n, t^{n+1}$上对守恒律 $q_t + F(q)_x = 0$ 积分，计算下一时间层的单元平均值。这得到

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} q_t dx dt + \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} F(q)_x dx dt = 0.$$

根据微积分基本定理可得

$$\begin{aligned} &\int_{x_{i-1/2}}^{x_{i+1/2}} q\left(x, t^{n+1}\right) dx - \int_{x_{i-1/2}}^{x_{i+1/2}} q\left(x, t^n\right) dx \\ &= -\int_{t^n}^{t^{n+1}} F\left(q\left(x_{i+1/2}, t\right)\right) dt + \int_{t^n}^{t^{n+1}} F\left(q\left(x_{i-1/2}, t\right)\right) dt. \end{aligned} \tag{2.7}$$

定义

$$\bar{F}_{i+1/2}^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F\left(q\left(x_{i+1/2}, t\right) dt \right. \tag{2.8}$$

并将式 (2.7) 两边除以 Δx，得到

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(\bar{F}_{i+1/2}^n - \bar{F}_{i-1/2}^n\right). \tag{2.9}$$

方程 (2.9) 表述了守恒关系：单元平均值的变化率由穿过单元边界的通量之差给出。图2对此作了示意。注意¯，关系式 (2.9) 并非显式形式，因为 F 需要预先知道精确解。有限体积格式的关键在于采用巧妙的方法来近似这些通量。

CFL条件

Courant-Friedrichs-Lewy 条件（简称 CFL 条件）是用于模拟对流或波动现象的非稳态数值方法的一种稳定性条件。考虑一个具有常系数的线性方程，例如 $q_t + aq_x = 0$，则 Courant 数定义为:

$$C = \frac{a\Delta t}{\Delta x},$$

其中 a 为速度大小， Δt 为时间步长， Δx 为网格单元之间的长度。库朗–弗里德里希斯–卢伊条件以Richard Courant、Kurt Friedrichs和Hans Lewy命名，他们在其论文 [12]中引入了该条件。该条件以Courant数表示为

$$C = \frac{a\Delta t}{\Delta x} \leqslant 1. \tag{2.10}$$

一旦确定了CFL数，我们就可以在每次迭代中计算所需的 Δt 。在第3、4、5章的后续实验中，我们在CFL数=0.3且 Δx 等于128的条件下预先计算 Δt ，然后根据分辨率比计算相应的 Δt。

2.3.2 有限体积法格式的数值通量

为了近似守恒律 $q_t + F(q)_x = 0$ 的数值解 Q ，ˆ ¯我们应使用数值通量格式 F ，以获得式（2.9）中的数值通量 F 。

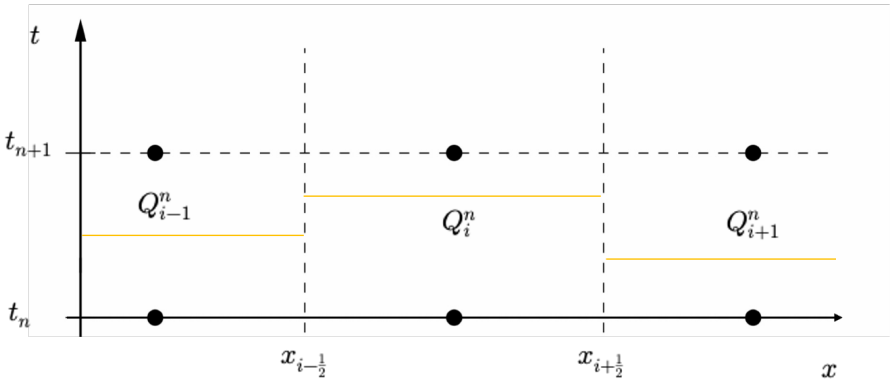


图2：单元平均值在每个界面处定义黎曼问题。

Lax-Friedrichs 通量

Lax-Friedrichs 通量 [13] 是一种简单的数值通量。只需知道 F 即可计算该通量：

$$\bar{F}^n_{i+1/2} = F^{LF}_{i+1/2} = \frac{1}{2}\left(F(Q^n_i) + F(Q^n_{i+1})\right) - \frac{\Delta x}{2\Delta t}\left(Q^n_{i+1} - Q^n_i\right). \tag{2.11}$$

Rusanov 通量

Rusanov 通量 [14]计算单元界面通量，而不考虑信息传播方向：

$$\bar{F}^n_{i+1/2} = F^{Rusanov}_{i+1/2} = \frac{1}{2}\left(F(Q^n_i) + F(Q^n_{i+1})\right) - s_R(Q^n_{i+1} - Q^n_i), \tag{2.12}$$

其中

$$s_R = \frac{\max\left(\left|F'\left(Q^n_i\right)\right|, \left|F'\left(Q^n_{i+1}\right)\right|\right)}{2}.$$

Roe 通量

求解非线性方程的一种常见方法是将其线性化。线性化是指将非线性通量函数 $q_t + F(q)_x = 0$ 替换为局部线性化形式

$$F(q)_x = F'(Q)Q_x \approx \hat{A}Q_x,$$

ˆ其中， $A \approx f'$ 是非线性通量函数围绕其进行线性化的常状态。ˆ 线性化状态的 $A(q_i, q_{i+1})$ 可以通过多种方式构造；构造时应满足一系列性质，并据此找到一种构造方法。

ˆ ˆ 1. A 应具有纯实特征值 $\lambda_i(q_i, q_{i+1})$，并具有对应的完备特征向量组 ˆv_i。

ˆ 2. 与精确雅可比矩阵的一致性： $A(q, q) = A(q)$。

ˆ 3. 守恒： $f(q_{i+1}) - f(q_i) = A(q_i, q_{i+1})(q_{i+1} - q_i)$

$$\bar{F}^n_{i+1/2} = F^{Roe}_{i+1/2} = \frac{1}{2}\left(F(Q^n_i) + F(Q^n_{i+1})\right) - \frac{1}{2}\sum_{p=1}^{n}\left|\tilde{\lambda}^{(p)}\right|\tilde{\alpha}^{(p)}\tilde{r}^{(p)}, \tag{2.13}$$

其中

$$\tilde{\alpha}^{(p)} = \tilde{l}^{(p)}\left(q_{i+1} - q_i\right),$$

$\tilde{\lambda}^{(p)}$、$\tilde{r}^{(p)}$、$\tilde{l}^{(p)}$ 分别为$\hat{A}$ 的特征值、右特征向量和左特征向量（p 从系统的方程数量 n 开始延伸）。也可写为

$$\bar{F}_{i+1/2}^{n}=F_{i+1/2}^{Roe}=\frac{1}{2}\left(F(Q_{i}^{n})+F(Q_{i+1}^{n})\right)-\frac{1}{2}\bar{R}|\bar{\Lambda}|\bar{R}^{-1}\left(Q_{i+1}^{n}-Q_{i}^{n}\right), \tag{2.14}$$

其中 $\bar{R}^{-1}$ 和 $\bar{R}$ 是将雅可比矩阵对角化的矩阵$\hat{A}$，Λ 是对角化后的雅可比矩阵$\hat{}$
$\hat{}$ 而 $|\Lambda|$ 是一个矩阵，其对角线上是 A 的特征值相对值。采用 Roe 通量的有限体积格式（2.13）或(2.14）称为 Roe 格式或 Murman-Roe 格式 [15]。

HLL通量

HLL格式 [16] 根据左右两侧的通量以及信息传播方向计算单元界面通量。$F_L=F(Q_i^n)$和 $F_R=F(Q_{i+1}^n)$分别是单元界面左侧和右侧对应的通量，s_L 和 s_R 分别描述在单元界面附近计算得到的最慢和最快传播速度：

$$\begin{cases} s_R=\max\left(\lambda_L,\lambda_R\right), \\ s_L=\min\left(\lambda_L,\lambda_R\right). \end{cases}$$

HLL求解器定义如下：

$$\bar{F}_{i+1/2}^{n}=\begin{cases} F_L & \text{if } 0\le s_L, \\ F_{i+1/2}^{Hll}=\frac{s_RF_L-s_LF_R+s_Ls_R\left(Q_{i+1}^{n}-Q_{i}^{n}\right)}{s_R-s_L} & \text{if } s_L<0\le s_R, \\ F_R & \text{if } s_R<0. \end{cases} \tag{2.15}$$

HLLE通量

HLLE [17] 与HLL格式类似：

$$\bar{F}_{i+1/2}^{n}=\begin{cases} F_L & \text{if } 0\le s_L, \\ F_{i+1/2}^{HllE}=\frac{s_RF_L-s_LF_R+s_Ls_R\left(Q_{i+1}^{n}-Q_{i}^{n}\right)}{s_R-s_L} & \text{if } s_L<0\le s_R, \\ F_R & \text{if } s_R<0, \end{cases} \tag{2.16}$$

但 s_L 和 s_R 定义为：

$$\begin{cases} s_R=\max\left(\lambda_R,\lambda_m,0\right), \\ s_L=\min\left(\lambda_L,\lambda_m,0\right), \end{cases}$$

其中 λ_m 由 Roe平均状态 q_m计算得到。HLLE 求解器非常稳健，因为它既能保持正性，又能自动满足熵不等式。然而，由于采用单一中间状态近似，HLLE 求解器无法解析非快速孤立间断，因此耗散性较强。

HLLC通量

HLLC [16] 是对 HLL 黎曼求解器的改进。

$$\bar{F}_{i+1/2}^{n}=F_{i+1/2}^{HLLC}=\begin{cases} F_L & \text{if } 0\le s_L, \\ F_{*L}=F_L+s_L\left(Q_{*L}-Q_{i}^{n}\right) & \text{if } s_L\le 0\le s_*, \\ F_{*R}=F_R+s_R\left(Q_{*R}-Q_{i+1}^{n}\right) & \text{if } s_*\le 0\le s_R, \\ F_R & \text{if } 0\ge s_R. \end{cases} \tag{2.17}$$

我们有四个未知向量：Q_{*L}、Q_{*R}、F_{*L}、F_{*R}。我们首先求解状态：Q_{*L}、Q_{*R}，然后求解通量：F_{*L}、F_{*R}。存在多种机制来定义中间状态的传播速度，而传播速度的选择会影响HLLC求解器的性能。此外，不同的方程具有不同的近似 [18, 19]。

2.3.3 二阶格式

空间离散

这里我们使用斜率限制器 [20] 来重构左右状态变量，这在抑制虚假数值振荡方面起着至关重要的作用。它们可以写成不同的等价形式，并且可能等价于相应的通量限制器。大多数严格针对一维空间中的单个非线性守恒方程或常系数方程推导而来。在后续实验中，我们主要使用MINMOD 限制器，它最初作为TVD 通量限制器被引入。MINMOD 限制器可用于如下重构二阶格式:

$$\begin{cases} Q^L_{i+1/2} = Q_i + MINMOD\left(\frac{Q_i - Q_{i-1}}{\Delta x}, \frac{Q_{i+1} - Q_i}{\Delta x}\right) * \frac{\Delta x}{2}, \\ Q^R_{i+1/2} = Q_{i+1} + MINMOD\left(\frac{Q_{i+1} - Q_i}{\Delta x}, \frac{Q_{i+2} - Q_{i+1}}{\Delta x}\right) * \frac{\Delta x}{2}, \end{cases} \tag{2.18}$$

其中,

$$MINMOD(a,b) = \begin{cases} a \text{ If } |a| < |b| \text{ and } ab > 0, \\ b \text{ If } |b| < |b| \text{ and } ab > 0, \\ 0 \text{ If } ab \le 0. \end{cases}$$

我们将其称为 MINMOD 格式。

时间离散

对于时间离散，我们可以使用经典的 Heun 方法 [21], 其中 $L(q)$ 是空间导数

$$\begin{aligned} q^{(1)} &= q^n + \Delta t L\left(q^n\right), \\ q^{n+1} &= \frac{1}{2}q^n + \frac{1}{2}q^{(1)} + \frac{1}{2}\Delta t L\left(q^{(1)}\right). \end{aligned} \tag{2.19}$$

为保持总变差递减（TVD）性质 $TV(u^{n+1}) \le TV(u^n) = \sum_j |u_{j+1} - u_j|$ 并提高时间精度，我们应采用高阶总变差递减龙格–库塔方法。四阶 TVD RK 方法 [22]如下:

$$\begin{aligned} q^{(1)} &= q^n + \Delta t L\left(q^n\right), \\ q^{(2)} &= \frac{3}{4}q^n + \frac{1}{4}q^{(1)} + \frac{1}{4}\Delta t L\left(q^{(1)}\right), \\ q^{n+1} &= \frac{1}{3}q^n + \frac{2}{3}q^{(2)} + \frac{2}{3}\Delta t L\left(q^{(2)}\right). \end{aligned} \tag{2.20}$$

2.3.4 双曲方程的算法

我们考虑一般的守恒律问题，其由以下偏微分方程表示，

$$\partial_t q + \nabla \cdot F(q) = 0. \tag{2.21}$$

然后，可以利用守恒律的积分形式，对控制体 Ω 应用有限体积法,

$$\partial_t q + \frac{1}{\Delta c}\oint_{\partial\Omega} F(q)\cdot\vec{n}ds = 0, \tag{2.22}$$

其中，$\vec{n}$是边界 ∂Ω上的外指单位法向量，$F(q)\cdot\vec{n}$是垂直于边界的通量。由此可得形式如下的离散有限体积法（2.9）:

$$Q^{n+1} = Q^n - \frac{\Delta t}{\Delta c}\sum_{i=1}^{N} h_i \bar{F}_i^n, \tag{2.23}$$

¯其中， F_i^n表示网格单元第 i条边上的平均法向通量的数值近似， N 为边的数量。在二维情形下， h_i 是第 i条边的长度，而Δc 是物理空间中测得的单元面积。在三维情形下， h_i 是单元界面的面积，而 Δc 是物理空间中的单元体积。现在，我们可以采用任意非结构网格方法，对本文讨论的网格上的双曲型问题进行数值近似。

在下一章中，我们将把它用于二维平面网格和逻辑矩形映射网格。对于逻辑矩形网格，物理空间可以是任意四边形区域，而计算空间是一个正方形。第3.3节展示了该方法如何在球面上近似求解浅水方程。在下一步中，我们可以将区域 Ω 划分为许多小单元 ω，每个单元也满足 (2.21)，于是得到：

$$\int_{\omega} \left(\partial_t q + \nabla F(q)\right) dA = 0,$$

与 (2.22) 类似，

$$\frac{\partial}{\partial t} q + \frac{1}{\Delta c} \oint_{\partial\omega} F(q) \cdot \vec{n} ds = 0.$$

单元间通量的近似值通过求解沿界面法向方向的一维方程组的黎曼问题得到。对于等熵方程，例如¯欧拉方程、浅水方程或声学方程，可以先将界面两侧单元平均物理量q 的速度分量旋转到界面的法向和切向方向，再计算网格单元界面处的通量 F 。使用这些修正后的数据求解一维黎曼问题，从而计算通量。最后，将通量的动量分量旋转回去，得到物理空间中的通量。

更具体地说，现在考虑 $\vec{n} = (n^1, n^2)$ 和 $\vec{t} = (t^1, t^2)$ 分别是单元界面的法向量和切向量。那么旋转矩阵 R 用于旋转速度分量（例如，对于浅水方程，其中 $q =(h,\ hu,\ hv)^T$ 具有如下形式，

$$q^\bullet = R \cdot q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n^1 & n^2 \\ 0 & t^1 & t^2 \end{pmatrix} \cdot \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}. \tag{2.24}$$

求解双曲方程时，算法 [23] 包括以下步骤：

1. 通过旋转速度分量，确定单元界面两侧的左右状态（$Q_L^{n\bullet}$和右侧 $Q_R^{n\bullet}$），即

$$Q_L^{n\bullet} = R \cdot Q_L^n, Q_R^{n\bullet} = R \cdot Q_R^n.$$

ˆ 2. 使用 2.3.2 中介绍的任意数值通量格式 F 计算 $F^{n\bullet}$

$$\bar{F}^{n\bullet} = \hat{F}(Q_L^{n\bullet}, Q_R^{n\bullet}).$$

¯3. 通过以下方式将数值通量 $F^{n\bullet}$ 旋转回笛卡尔坐标

$$\bar{F}^n = R^T \cdot \bar{F}^{n\bullet}.$$

¯4. 重复步骤 1、2、3，计算所有边的数值通量 F_i^n。

5. 使用有限体积法 (2.23) 更新单元状态 Q^{n+1} ，注意， h_i 和Δc 可以预先计算。

2.4 球面上的逻辑矩形网格

球面流形上的输运可采用多种网格，其中最常用的是经纬度网格（Lin & Rood，1996）、立方球网格（Nairetal.，2005；Putman & Lin，2007；Ronchietal.，1996）、阴阳网格（Hall & Nair，2013；Kageyama & Sato，2004）和逻辑矩形网格（Donna & Christiane & Randall，2007） [24]。经纬度网格的算法简单，但极区附近经线汇聚，会导致欧拉格式出现 Courant-Friedrichs-Lewy（CFL）稳定性问题和严重的极点奇异性；此外，为了稳定欧拉格式和拉格朗日方法中的极区流，还需要采用纬向全局方法。阴阳网格正交且易于实现，相比经纬度网格，其额外优势是网格间距准均匀。但要在全局矩形网格上实现高阶精度的质量守恒，需要使用全局质量修正器。立方球网格系统独特地覆盖整个球面，可轻松实现精确的局部质量守恒，且具有逻辑矩形、准均匀的特点，无需全局运算。其缺点是使用了六个独立的非正交坐标系，需要进行度量变换，并且需要考虑更复杂的边界条件。我们选择逻辑矩形网格，因为它具备我们偏好的全部四项特性：完全局部运算、局部质量守恒、均匀的笛卡尔网格以及严格的逻辑矩形网格。

2.4.1 圆盘的映射

本节中，我们介绍一种将计算域 $[-1,1]\times[-1,1]$映射到单位圆盘的映射。为了描述该映射，我们重点关注正方形中的一个区域，在该区域内，计算坐标 (x_c, y_c) 满足 $|y_c| \le x_c$。此区域对应于正方形两条对角线之间的上三角区域。我们将该区域称为计算域的北部扇区。该扇区中的映射基于这样一个思路：将连接 $(d,-d)$ 和 (d,d) 的竖直线段映射为一段半径为 $R(d)$ 的圆弧，该圆弧经过点 $(D(d)\ -D(d))$ 和 $(D(d)\ D(d))$。如下文将进一步讨论，映射由函数 $R(d)$ 和 $D(d)$ 完全定义。容易计算得出，该半径为 $R(d)$ 的圆的圆心位于 $(x_0, y_0) = \left(D(d) - \sqrt{R(d)^2 - D(d)^2}, 0\right)$。于是，点 (x_c, y_c) 被映射到

$$
\begin{aligned}
y_p &= y_c D(d)/d, \\
x_p &= D(d) - \sqrt{R(d)^2 - D(d)^2} + \sqrt{R(d)^2 - y_p^2}.
\end{aligned}
$$

其他四个扇区也采用类似的映射。例如，水平线段上位于 $(-d,d)$ 和 (d,d) 之间的点，会被映射到半径为 $R(d)$ 且经过 $(-D(d)\ D(d))$ 和 $(D(d)\ D(d))$ 的圆弧上的点。

$R(d)$ 和 $D(d)$ 有许多种选择，可以生成实用的网格。例如，一种特别适用于单位圆盘的选择是

$$
\begin{aligned}
D(d) &= d/\sqrt{2}, \\
R(d) &= 1.
\end{aligned}
$$

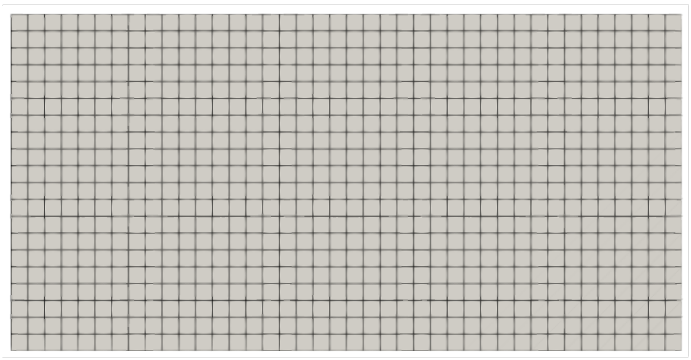
2.4.2 单位球面的映射

上述单位圆盘映射可直接用于为单位半球面或球面创建网格。例如，要对半球面进行参数化，我们只需将其描述为

圆盘映射中物理位置 (x_p, y_p) 的函数：

$$z_p = \sqrt{1 - (x_p^2 + y_p^2)}. \tag{2.25}$$

于是，点（x_p， y_p， z_p）位于上半球。我们可以对计算域 $[-3, -1] \times [-1, 1]$ 中的点应用类似的映射，将这些点映射到下半球。这两个映射将矩形 $[-3, 1] \times [-1, 1]$ 映射到球面。图3展示了所得网格。



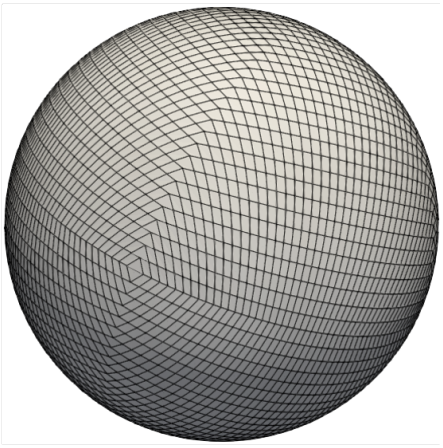


图3：在t　　左侧是计算网格，右侧图片中所示为球面网格　　e.

2.5 机器学习加速的计算流体力学

本节介绍两种使用神经网络求解计算流体力学的方法。第一种使用神经网络生成导数，第二种使用神经网络重构值来获得边界值，其思路与2.3.3类似。

2.5.1 空间导数的数据驱动近似

我们考虑一个具有连续解的通用偏微分方程 $u(x,\ t)$ [4]:

$$\frac{\partial u}{\partial t} = F\left(t, x, u, \frac{\partial u}{\partial x_i}, \frac{\partial u}{\partial x_i \partial x_j}, \cdots\right), \tag{2.26}$$

我们可以使用传统数值方法求解该问题，例如有限差分法和有限体积法。这些方法需要对空间域进行离散化，并计算近似的空间导数。近似空间导数的方法有多种，例如多项式展开和谱二分法。以一维有限差分欧拉法为例：

$$\frac{\partial u(x_i, t)}{\partial x} = \frac{u_{i+1}(t) - u_i(t)}{\Delta x} + \mathcal{O}(\Delta x),$$

其中，解 $u(x,\ t)$ 由其在节点 $u_i(t) = u(x_i,\ t)$ 处的值表示，而$\Delta x = x_i - x_{i-1}$ 是空间分辨率。

同样，我们可以得到其余的空间导数 $\frac{\partial^n u}{\partial x^n}(t) \approx \sum_j \alpha_j^{(n)} u_j$。传统方案对空间中的所有点使用一组预计算系数，而更先进的方法则根据局部规则在不同的系数组之间交替使用。我们可以利用机器学习来确定这些模板之间复杂的系数。

现在，方程(2.41)可以转化为一组耦合常微分方程，其中包含每个节点的值 $u_i(t)$:

$$\frac{\partial u_i(t)}{\partial t} = F\left(t, x, u_1(t), \cdots, u_N(t)\right), \tag{2.27}$$

然后，我们可以使用欧拉法或龙格–库塔法等线方法，计算每个空间节点上的方程数值解。

首先，我们从高分辨率数据中生成训练集，并从该数据集中学习导数的离散近似。这需要权衡计算成本；可以通过对小型系统进行高分辨率模拟、开发用于求解流形的局部近似，并利用这些近似以较低的空间分辨率求解更大规模系统的方程来缓解这一问题。

我们的网络表示为空间导数的广义有限差分公式：网络的输出是一列系数 α_1，$\ldots\alpha_N$，使得第 n 阶导数以伪线性公式的形式计算。作为伪线性滤波器，这些系数 $\alpha_i^{(n)}(u_1(t)\ u_2(t)\ldots)$ 随场值对时空的依赖而变化，这些场值位于相邻单元中。找到最优系数是该方法的关键。

2.5.2 数值通量的数据驱动近似

直接通过卷积神经网络计算边界通量

该方法使用卷积神经网络直接计算边界通量 F_i^l和 F_i^r，这些通量由单元平均状态 Q_i 得到。随后，我们可以使用 2.3.3 节介绍的 Heun 方法或总变差递减龙格–库塔方法更新状态值。我们在 5.1 节对该方法进行了数值实验。

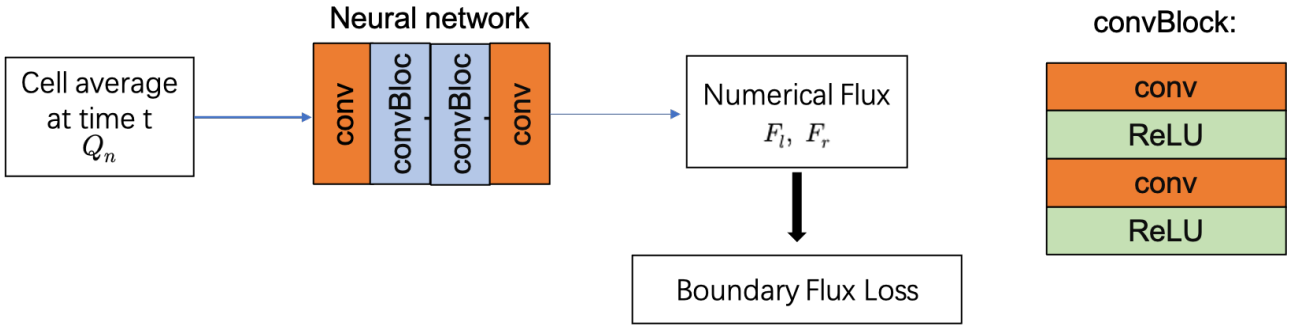


图 4：第一种神经网络结构：卷积神经网络直接输出边界数值通量。

2.5.3 边界状态的数据驱动近似

除了使用数据驱动方法直接获得时间导数（即通量值）外，我们还可以尝试使用神经网络将中心单元值重构为单元边界值，随后利用数值通量获得高精度通量。在本学位论文中，我们提出了三种基于卷积神经网络的重构方法。这些方法与高精度加权本质无振荡格式的原理相似，旨在获得稳定、无振荡且清晰的间断跃迁。神经网络能够获得局部区域的线性权重，从理论上说，这可以实现任意阶插值。

使用卷积神经网络生成的线性系数计算边界状态

该方法的思路是使用卷积神经网络计算插值到边界值的线性约束系数

然后，将这些系数与模板的相应值相乘，即可得到边界值。接着，我们便可以计算数值通量。这里，我们的训练集是将高分辨率结果粗化后得到的单元中心值和边界值。由于插值得到的通量可能会产生振荡，我们可以使用通量限制器消除不连续性，如下所示：

$$F\left(q_{i+\frac{1}{2}}\right) = F^{\text{low}}_{i+\frac{1}{2}} - \phi\left(r_i\right)\left(F^{\text{low}}_{i+\frac{1}{2}} - F^{\text{high}}_{i+\frac{1}{2}}\right),$$
$$F\left(q_{i-\frac{1}{2}}\right) = F^{\text{low}}_{i-\frac{1}{2}} - \phi\left(r_{i-1}\right)\left(F^{\text{low}}_{i-\frac{1}{2}} - F^{\text{high}}_{i-\frac{1}{2}}\right),$$

其中， F^{low} 是由低分辨率网格计算得到的低分辨率通量， F^{high} 是通过插值获得的高分辨率通量， r 表示解网格上连续梯度的比值，即 $r_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_i}$ 。我们在第5.2节中展示了采用这种方法进行的实验。

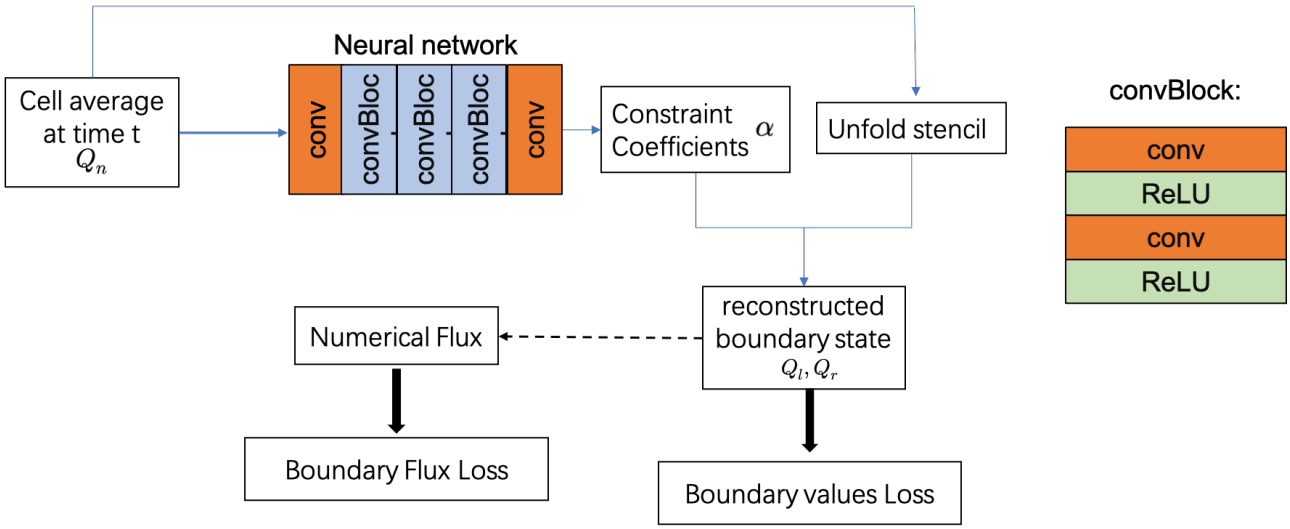


图5：第二种神经网络结构：利用卷积神经网络生成的线性系数计算边界状态。

直接通过卷积神经网络计算边界状态

在这种方法中，我们使用卷积神经网络直接从单元平均状态 Q_i重构边界处的状态 Q_i^l 和 Q_i^r 。这里可以直接使用通量向量 $F(q)$，也可以使用任意数值通量格式。请注意，此处 $Q_i^r = -Q_{i+1}^l$。对于神经网络，由于各守恒分量的性质不同（计算数值通量时，水深项必须为正）。对于神经网络的输入，我们首先对输入值进行归一化，并将其分为水深项 h 和动量项 hu。随后，我们使用若干层卷积函数，并在每层之间使用一层 GeLu 作为激活函数。水深项所用卷积神经网络的最后一层需要使用 Softplus，以确保重构的水深大于零。

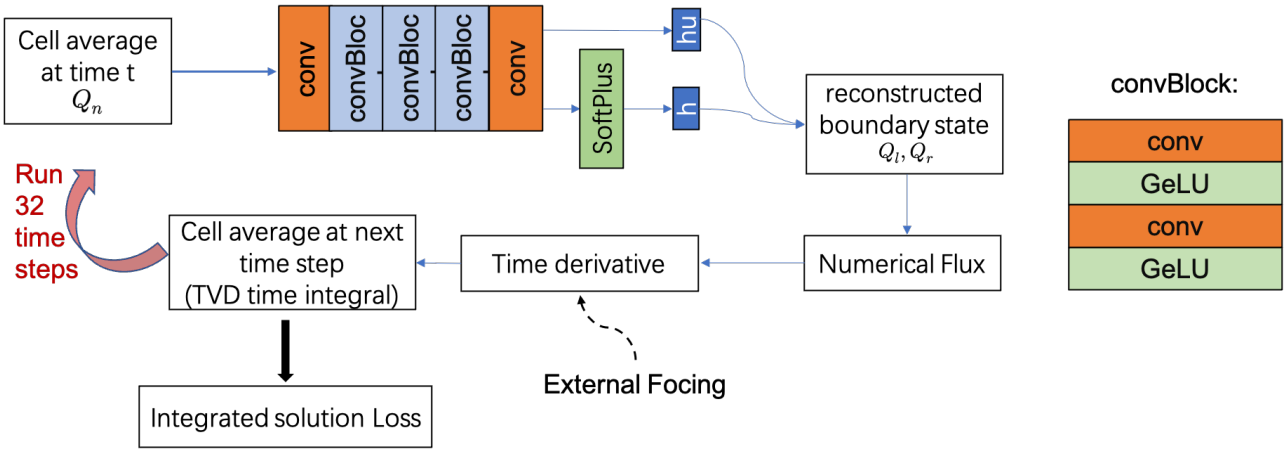


图6：第三种神经网络结构：直接通过卷积神经网络计算边界状态 .

从卷积神经网络获得边界状态后，我们计算 Q_i^r 和 Q_i^l ，并更新每个单元状态。将数值求解器纳入训练损失可确保实现完全“模型一致”的训练，从而提高训练稳定性。在接下来的实验中，我们使用低分辨率损失连续 32 步所得结果之和作为训练误差。训练期间，我们希望尽量缩小高成本高分辨率模拟与模型在粗网格上生成的模拟之间的差异。我们通过监督训练实现这一目标：将预测的守恒分量与真实分量值之间的累积点误差用作损失函数：

$$Loss = \sum_{i=1}^{T} \mathrm{MSE}\left(Q_{exact}\left(t_i\right), Q_{pred}\left(t_i\right)\right), \tag{2.28}$$

其中，均方误差表示均方误差。请注意， $T = 32$ Q_{exact} 是每个时间步长上的真实值，而 Q_{pred} 是从神经网络获得的状态。实验细节见第5.3节。

C使用卷积神经网络生成的线性系数计算重构斜率 ts

在此方法中，我们使用卷积神经网络为每个守恒分量的重构斜率生成一组约束系数 alpha。请注意，我们也可以对这些系数进行初始化，以确保达到一阶精度。将相应的模板与重构值相乘后，我们可以得到边界值 Q_r 和 Q_l。在时间积分中，我们采用与前一种方法相同的方式来确保解的稳定性。然而，无法保证插值高度始终为正。因此，在训练过程中还应设置额外的损失。另一种方法是在 (2.11) 中使用 Lax-Friedlich 数值格式 [5], ，但与其他数值通量相比，其误差相对较大。此外，我们可以添加总能量损失函数以提高稳定性。我们将在第 5.4 节介绍这种方法的实验。

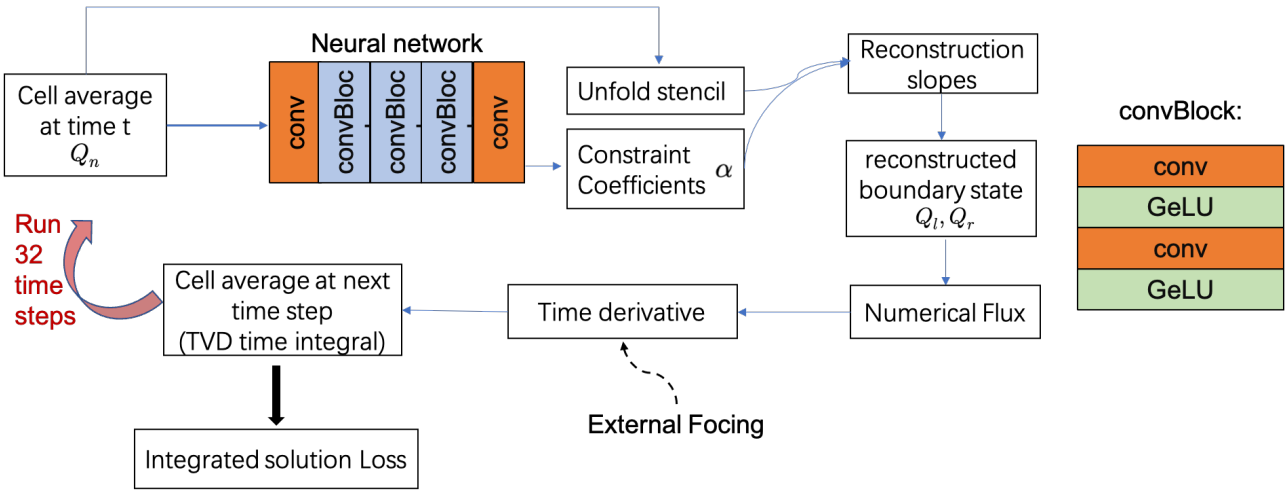


图 7：第四种神经网络结构：利用卷积神经网络生成的线性系数计算重构斜率。

3 浅水方程的 ClassicSolver

本章中，我们针对每种数值通量，测试一维、二维和球面情形下的经典浅水方程求解器，并将其与 Pyclaw 的求解器进行比较。我们使用水深项的相对误差：

$$Error = \frac{\sum_{i=1}^{n} \frac{h_i^{high} - h_i^{low}}{h_i^{high}}}{n},$$

其中， n 表示单元数量， h_i^{high} 是 Pyclaw 中粗化高分辨率解的结果。此外， h_i^{low}是使用我们的求解器得到的测试解。另外，我们比较了 Pyclaw、torch 和 Dace框架在 CPU 和 GPU 上的运行时间。

3.1 一维浅水方程

我们在 2.1 节中介绍了一维浅水方程，其中 $q = (h, hu)^T$ 表示状态向量，其包含水深 h 和动量。通量向量表示为：

$$F(q) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}.$$

3.1.1 一阶求解器

这里我们在区域 $x \in [0, 1]$ 中使用溃坝问题作为测试算例，并采用周期性边界条件（周期性边界条件）作为边界条件。

$$h = \begin{cases} 1 & x < 0.5, \\ 0.35 & x \geqslant 0.5. \end{cases}$$

图8展示了采用 2.3.2 节中介绍的数值通量运行 1 秒后的结果。参考解基于 Pyclaw 中细网格上的 HLLE SharpClaw 求解器获得。

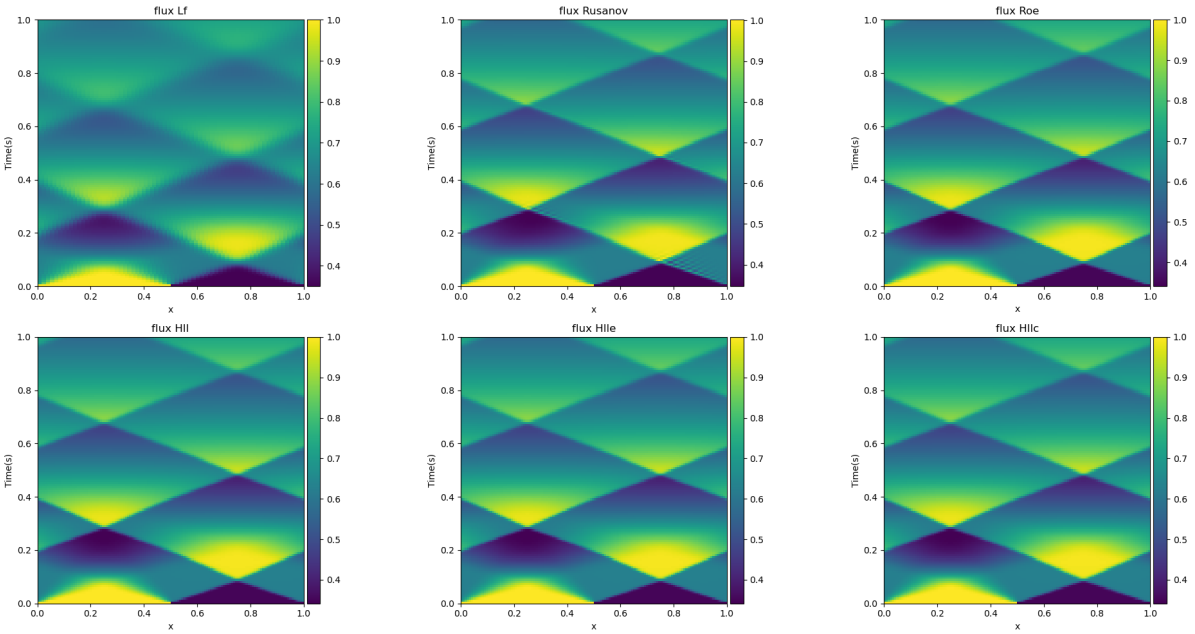


图8：使用不同数值通量求解不同测试案例中分辨率为128×128 的一阶一维浅水方程黎曼问题。

误差 Flux / nx	LF	Rusanov	Roe	HLL	HLLE	HLLC
64	9.15 %	2.92 %	2.75 %	2.81 %	2.75 %	2.78%
128	5.15 %	1.50 %	1.48 %	1.51 %	1.47 %	1.49%
256	2.67 %	0.90 %	0.95 %	0.96 %	0.94 %	0.95%
512	1.36 %	0.56 %	0.61 %	0.62 %	0.61 %	0.61%

表1：使用不同数值通量和加密网格的一维浅水方程一阶格式相对误差。

表1显示，随着网格增大，所有数值解的相对误差都在减小。其中，Lax-Friedlich格式的误差最显著，而在低精度下，Roe和HLLE的表现最好。这与图8一致，L-F方法给出的结果最模糊。

3.1.2 二阶求解器

我们采用与上一节相同的测试算例，并使用 MINMOD 斜率限制器格式 (2.18)，以获得具有高阶精度的数值解，

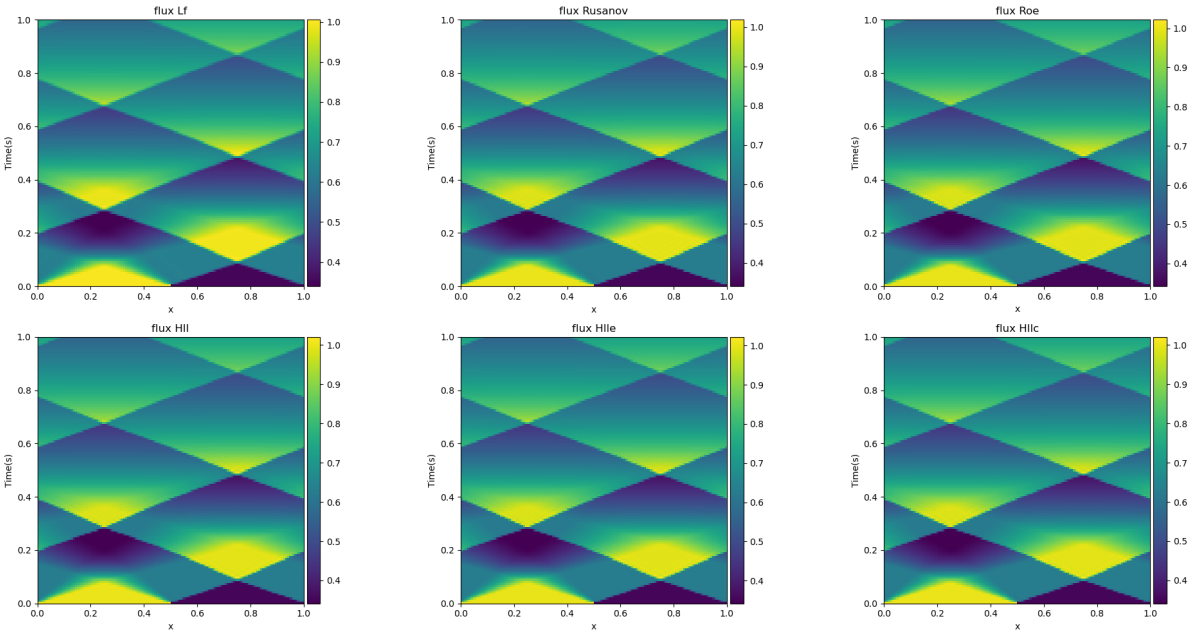


图9：使用不同数值通量对具有128×128 分辨率的二阶一维浅水方程黎曼问题测试案例的解。

误差 Flux / nx	LF	Rusanov	Roe	HLL	HLLE	HLLC
64	0.99 %	0.75 %	0.72 %	0.73 %	0.72 %	0.73%
128	0.46 %	0.39 %	0.36 %	0.37 %	0.36 %	0.36%
256	0.34 %	0.32 %	0.30 %	0.30 %	0.30 %	0.30%
512	0.27 %	0.27 %	0.26 %	0.26 %	0.26 %	0.26%

表 2：采用二阶格式、不同数值通量及加密网格时的一维浅水方程 h 相对误差。

图 8 和图 9 显示，二阶格式的结果优于一阶方法。在低分辨率下，相对误差甚至降低了四倍以上。此外，与一阶方法的结果一致，Lax-Friedlich 的表现仍然最差，而其余数值通量格式的相对误差几乎相同。在接下来的章节中，我们将比较三种二阶数值通量格式的结果：Rusanov、Roe 和 HLLE。

3.2 平面 SWEson

这 二维笛卡尔坐标下的浅水方程 $\vec{x} = (x, y)$ 可表述为：

$$\partial_t q + \nabla \cdot F(q) = S(\vec{x}, q),$$

其中， $q = (h, hu, hv)^T$ 表示由水深 h 和两个笛卡尔速度分量 $\vec{u} = (u, v)^T$ 构成的状态向量，这些量都是时空的函数。通量矩阵的形式为

$$F(q) = \begin{pmatrix} hu & hv \\ hu^2 + \frac{1}{2}gh^2 & huv \\ huv & hv^2 + \frac{1}{2}gh^2 \end{pmatrix}.$$

当河床不是平面时，源项为 $S(\vec{x})$，它仅作用于动量，如式 (2.3) 所示，

$$S(x) = \begin{pmatrix} 0 \\ -gh\frac{\partial z}{\partial x}(x,y) \\ -gh\frac{\partial z}{\partial y}(x,y) \end{pmatrix}.$$

3.2.1 无源项的二阶求解器

首先，我们在区域 $x \times y \in [0,1] \times [0,1]$中采用静水问题作为测试算例。水深项采用二维高斯分布 $\exp(-((x-0.5)^2+(y-0.5)^2)/\sigma^2)+0.5, \sigma = 0.5$ 作为初始条件，并将 PBC（周期性边界条件）作为边界条件。图 10 展示了使用 2.3.2 节介绍的数值通量运行 $1.5s$ 后，在具有 400×400 个网格单元的网格上得到的水深。参考解基于 Pyclaw 中细网格 1000×1000 上的 HLLE SharpClaw 求解器获得。

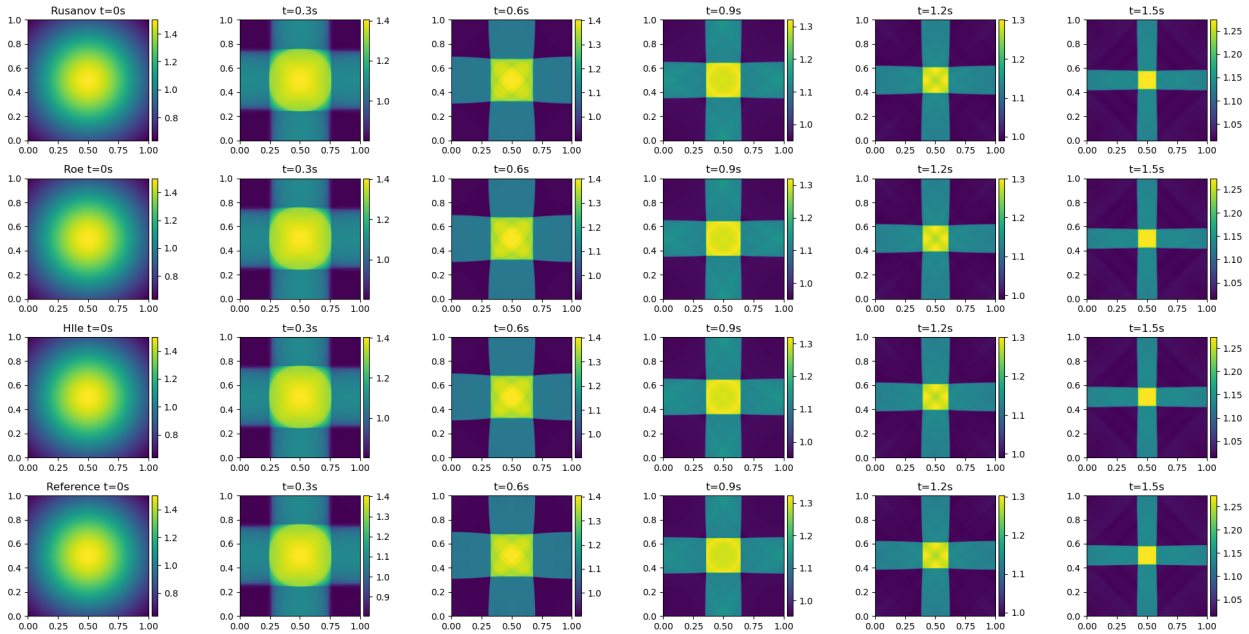


图 10：使用不同数值通量时，分辨率为 400×400 的二维浅水方程测试案例的解。

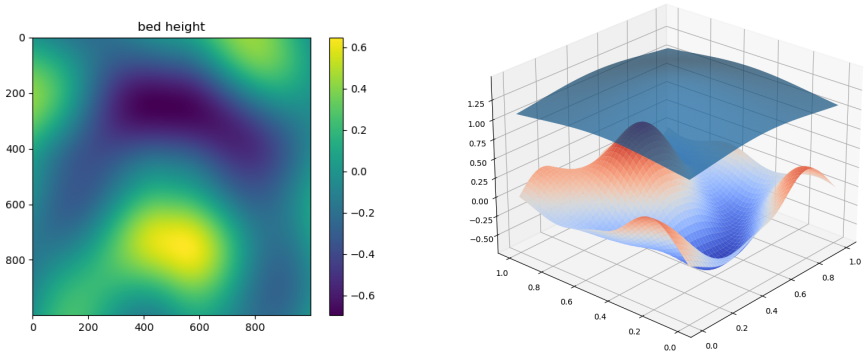
误差 Flux / nx	Rusanov	Roe	HLL	HLLE	HLLC
100×100	0.44 %	0.43 %	0.43 %	0.44 %	0.43%
200×200	0.19 %	0.19 %	0.19 %	0.19 %	0.19%
400×400	0.08 %	0.07 %	0.07 %	0.07 %	0.07%
800×800	0.05 %	0.05 %	0.05 %	0.05 %	0.05%
1000×1000	0.05 %	0.03 %	0.04 %	0.03 %	0.03%

表 3：采用不同数值通量和更密网格的二阶格式计算二维浅水方程的 h 相对误差。

3.2.2 含源项的二阶求解器

我们使用二维柏林噪声生成水床 $z(x, y)$，如图11所示。此外，我们采用与测试算例相同的初始条件。这里我们使用 Roe 作为

数值通量和壁面边界条件。参考解来自 Pyclaw 中基于 1000×1000 网格的水深求解器。



(a) 二维水床　　(b) 三维水床和水面

图11：由二维Perlin噪声生成的水床。

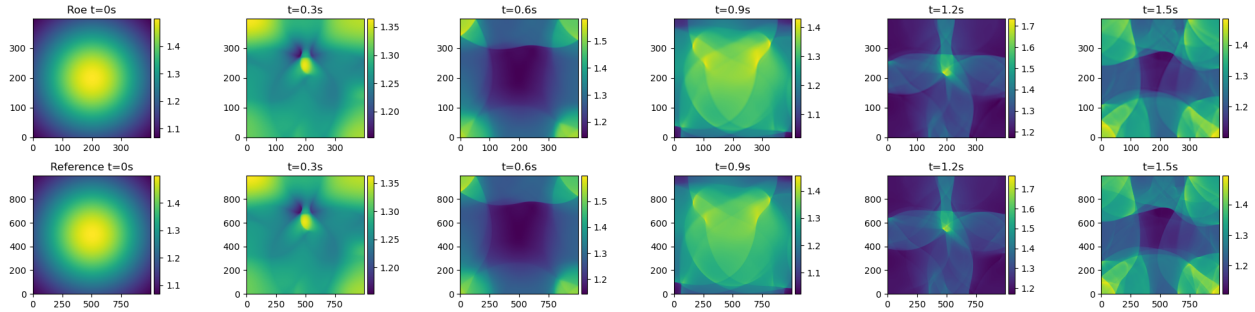


图12：分辨率为 400×400 时二维浅水方程的解，水床　　.

最终的图12表明，我们的求解器与 Pyclaw 中的求解器一致。

3.3 球面上的浅水方程

例如，在全球大气或海洋模型中，浅水方程通常在球面上求解。现在，方程采用三维笛卡尔坐标$\vec{x} = (x,\ y,\ z)$ 表述，其中 $q = (h,\ hu,\ hv,\ hw)^T$ 表示由水深h 和三个笛卡尔速度分量 $\vec{u} = (u,\ v,\ w)^T$ 组成的状态向量。通量矩阵的形式为：

$$F(q) = \begin{pmatrix} hu & hv & hw \\ hu^2 + \frac{1}{2}gh^2 & huv & huw \\ huv & hv^2 + \frac{1}{2}gh^2 & hvw \\ huw & hw & hw^2 + \frac{1}{2}gh^2 \end{pmatrix}.$$

为了获得守恒更新，我们还应在每个时间步中，从 Q^n 的每个网格单元平均值中减去修正项。如果令所有法向量均指向外侧，则有：

$$Q^{cor} = -F(Q^n) \cdot \sum_{i=1}^{N} h_i \vec{n}_i. \tag{3.1}$$

源项 $S(x,\ q)$ 仅作用于动量方程，其形式为：

$$S(x,q) = -\frac{2\omega}{a} z(x \times hu) + (x \cdot (\nabla \cdot \tilde{f}))x. \tag{3.2}$$

式 (2.1) 右侧的第一个源项是地球自转产生的科里奥利力。这里， ω 和 a 分别是角速度和地球半径。式 (2.1) 中的第二个源项是一个强迫项，用于保证流体速度始终位于球面上，即与球面上的位置矢量 $\vec{x} = (x,\ y,\ z)$ 垂直。~这里， f 由通量矩阵中与动量方程相关的部分构成。笛卡尔形式的优点是它不依赖于所采用的坐标变换。三维方程的近似仅限于球面，因此我们需要更新球面网格单元中的 Q 。我们的离散化遵循第 2.3.4 节概述的一般流程。具体细节如下：

1. 计算单元边长 h_i 和单元面积 Δc，这里分别指物理区域中的距离和面积。采用大圆距离和面积。2. 在切平面内，为界面的每一侧定义指向界面的法向方向 $\vec{n}$ 和切向方向 $\vec{t}$；其中， $\vec{t}$ 可利用边矢量定义， $\vec{n}$ 可通过 $\vec{t}$ 与 $\vec{t}$ 的向心矢量的叉积计算，现有 $\vec{n},\ \vec{t} \in \vec{\mathbb{R}}^3$。3. 使用旋转矩阵 R （见式 (2.24)）旋转网格单元值中的速度分量。4. 求解一维黎曼问题，并使用 R^T 将通量 $\bar{F}^{n\bullet}$ 的动量分量旋转回笛卡尔坐标 $\bar{F}^n$ 。5. 重复步骤 1–4，以获得所有单元的界面通量。6. 使用高阶龙格–库塔方法计算修正项 (3.1) 和科里奥利力 (3.2)。7. 更新所有单元的状态；随后，在每个时间步，将动量分量投影到各单元中心的切平面上。

接下来，我们将 Rossby–Haurwitz波作为测试问题。在下面的示例中，我们考虑对波数为四的罗斯贝-豪威茨流型进行近似，这是旋转球面上的浅水方程的标准测试问题。有关初始条件的详细描述，请参阅 Williamson 等人。 [25] 豪威茨指出，该流型以恒定角速度自西向东移动，且形状保持不变。尽管罗斯贝-豪威茨波并非浅水方程的解析解，但其波结构会长期绕 Z轴旋转。在我们的模拟中，特征时间尺度设为一天，特征长度尺度设为地球半径。

图13展示了在包含 1000×500 网格单元的网格上，于不同时间获得的水深。这里我们使用带有（MINMOD）通量限制器的Roe通量格式二阶求解器。

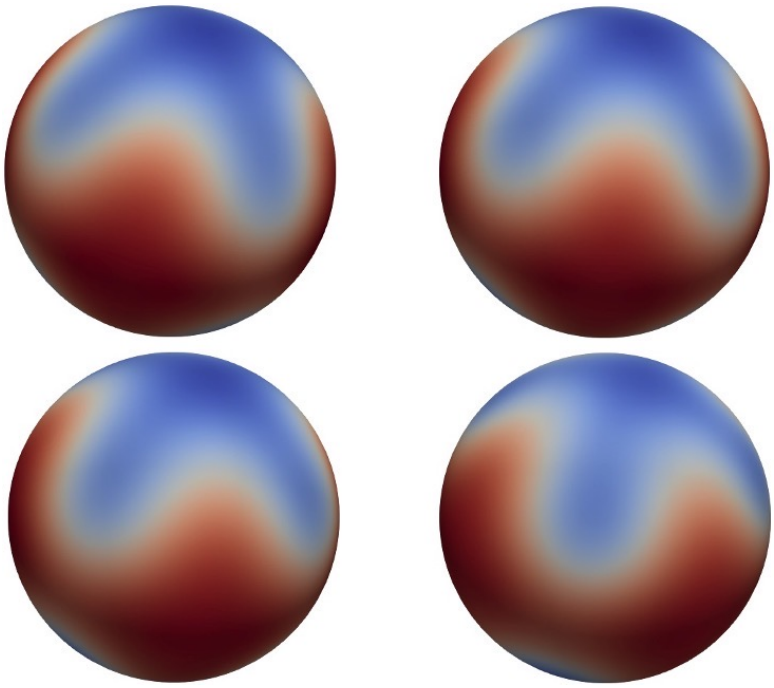


图13：以罗斯贝-豪尔维茨波为初始数据的球面浅水方程。所示为流体高度 h。上面两幅图（时间为 t=0和 1 天）为俯视图。下面两幅图（时间为 t=2 天和 4 天）为仰视图。

误差 Flux / nx	Rusanov	Roe	HLL	HLLE	HLLC
100×50	0.42%	0.37%	0.42 %	0.42 %	0.37%
200×100	0.15 %	0.14 %	0.15 %	0.15 %	0.14%
400×200	0.07 %	0.07 %	0.08 %	0.08 %	0.07%

表 4：采用三种分别包含 100×50、200×100 和 400×200 个网格单元的网格及不同数值通量格式计算罗斯贝-豪尔维茨波。该表显示了 1 天时的 h 相对误差。

3.4 不同框架下的计算开销

我们在Dace（以数据为中心的并行编程）[26]和Torch框架上实现了上述经典浅水方程求解器。本节将其速度性能与Pyclaw中的求解器进行比较。需要注意的是，Pyclaw不支持在GPU框架下进行计算，在处理复杂问题时这是一个缺点。我们统一使用二阶精度的Roe黎曼求解器。此外，我们还测试了不同设备上的性能，其中CPU使用AMD EPYC 7742 @ 2.25GHz，GPU测试设备使用NV RTX3090。

平面浅水方程求解器的结果如图14所示。在DACE框架下，我们需要100秒来预编译求解器代码。预编译完成后，我们可以输入任意时段和分辨率进行计算。

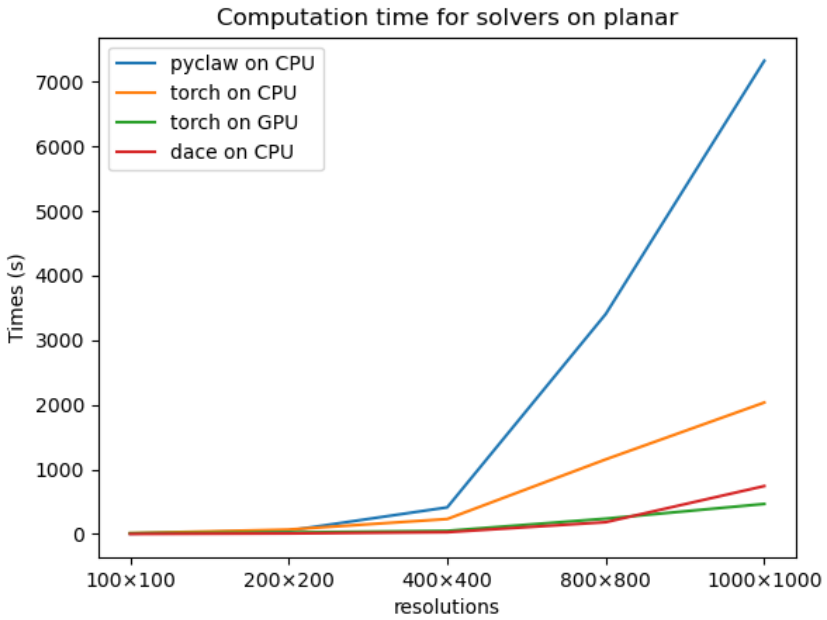


图14：比较不同框架下二维浅水方程求解器的性能。分辨率范围为 100×100 至 1000×1000。

可以看到，无论在哪种框架下，我们的求解器表现都远优于Pyclaw。在CPU上运行时，Dace的表现优于torch；尽管在Dace下运行时代码需要一些时间进行预编译，但在处理高精度多时间步时，这可以成为一个巨大优势。此外，我们使用相同的设备和方法，比较了不同框架下球面上求解器的计算效率。球面上的浅水方程求解器比平面求解器复杂得多，因此在Dace框架下需要更多预编译时间，大约需要400秒。

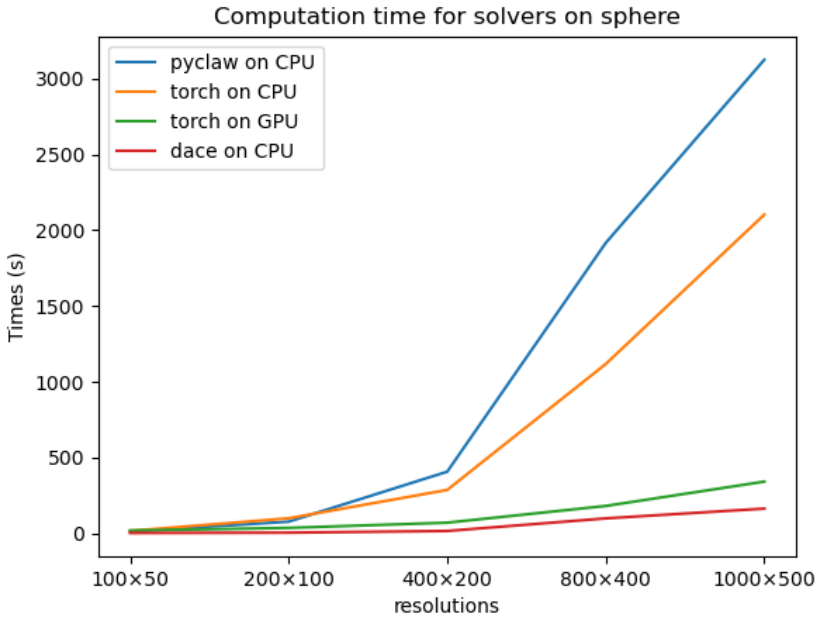


图15：使用 Rossby–Haurwitz 波比较浅水方程球面求解器在 1.5 秒内的性能。分辨率范围为 100×50 至 1000×500。

从图15可以看出，在不同设备上，我们的所有求解器在速度方面都优于 Pyclaw 的球面求解器。Pyclaw 球面求解器的最高分辨率为1000×500，而我们的求解器可以达到任意精度。当需要对高精度大气数据进行建模时，这一点至关重要。球面上的求解器比平面上的求解器复杂得多。令人惊讶的是，我们发现，在 Dace框架 上运行的球面求解器在 CPU 上的表现甚至优于在 GPU 上运行的 torch。处理复杂问题时，使用 Dace框架 可以提高效率。

4 欧拉方程的经典求解器

4.1 一维欧拉方程

2.2 节中给出的一维欧拉方程，其中 $q = (\rho, \rho u, E)^T$ 表示状态向量，包含密度 ρ和动量。通量向量表示为：

$$F(q) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix} = \begin{pmatrix} q_2 \\ q_2^2/q_1 + p\,(q_1, q_2, q_3) \\ (q_2/q_1)\,(q_3 + p\,(q_1, q_2, q_3)) \end{pmatrix},$$

其中 $p\,(q_1, q_2, q_3) = (\gamma - 1)\,\left(q_3 - \frac{1}{2} q_2^2/q_1\right)$。

4.1.1 一阶求解器

使用以下测试初始条件，$\gamma = 7/5$

$$q = \begin{cases} (1, 0, \frac{1}{\gamma - 1})^T & x < 0.5, \\ (0.125, 0, \frac{0.1}{\gamma - 1})^T & x \geqslant 0.5. \end{cases}$$

图16展示了采用2.3.2节中介绍的数值通量运行1秒后的结果。参考解基于来自 Pyclaw HLLE SharpClaw 求解器的细网格 1024×1024。

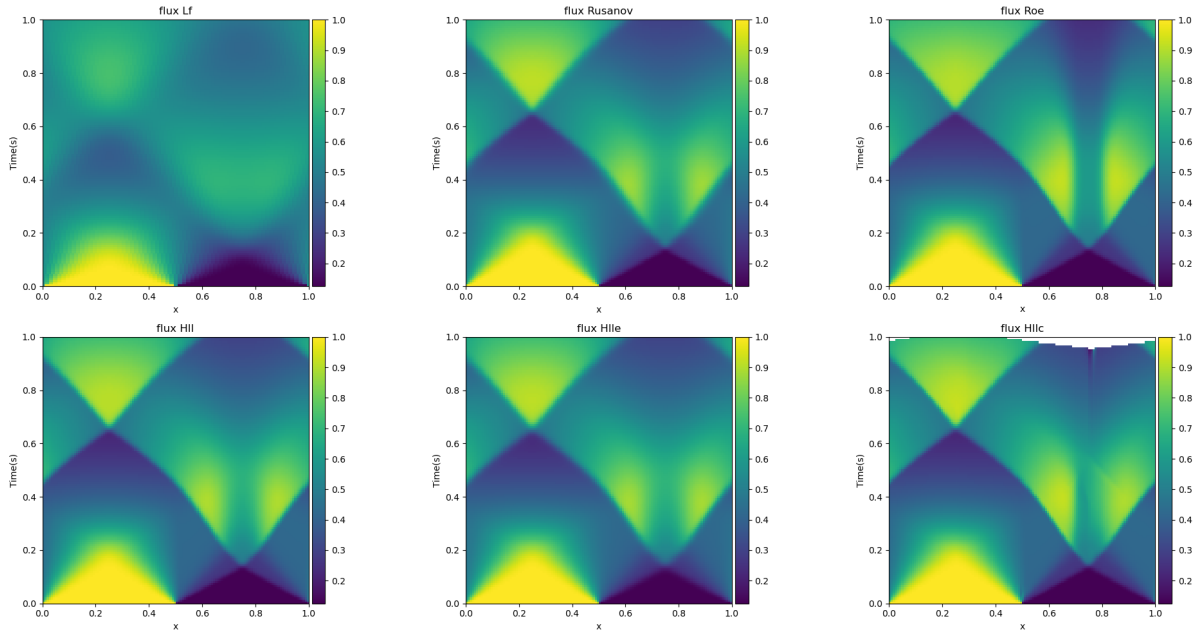


图16：测试算例中，使用不同数值通量求解分辨率为 128×128 的一维欧拉方程一阶解。

从图16可以看出，与第3章中的浅水方程求解器相同，L-F方法得到的解最为模糊。需要注意的是，在计算结束时，HLLC 格式求解器会生成不稳定的解。

误差 Flux / nx	LF	Rusanov	Roe	HLL	HLLE	HLLC
64	52.52%	22.41%	10.96%	19.91%	23.09 %	17.52 %
128	42.58 %	14.82 %	6.54 %	13.69 %	15.65 %	Nan
256	29.68 %	9.91 %	4.25 %	9.64 %	10.49 %	Nan
512	19.33 %	7.05 %	2.66 %	6.72 %	7.50 %	Nan

表5：采用不同数值通量的一维欧拉方程一阶格式的相对误差及加密网格。

根据表5，随着网格增大，所有数值解的相对误差都会减小。此外，Lax-Friedlich格式的误差最大，而Roe格式在所有分辨率下的表现最佳。由于HLLC格式在最后时刻产生了不稳定的解，因此无法计算误差。

4.1.2 二阶求解器

我们在浅水方程求解器中采用相同的二阶方法，以实现高阶欧拉方程求解器。

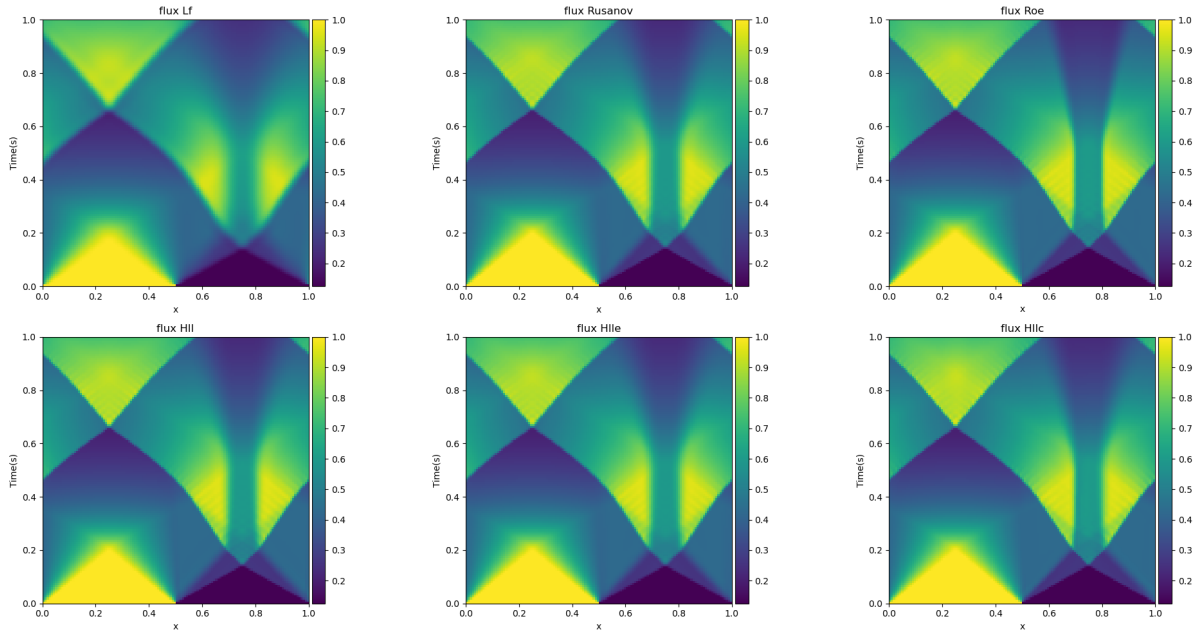


图17：使用不同数值通量求解测试案例时，二阶一维欧拉方程在分辨率为 128×128下的解。

误差 Flux / nx	LF	Rusanov	Roe	HLL	HLLE	HLLC
64	7.65 %	4.93 %	2.63 %	4.89%	5.14%	4.36%
128	4.88%	3.15%	1.76%	3.10 %	3.32 %	2.90%
256	2.19 %	1.63 %	0.68 %	1.60 %	1.73 %	1.44%
512	1.53 %	0.70 %	0.37 %	0.68 %	0.73 %	0.57%

表 6：采用二阶格式、不同数值通量及加密网格时，一维欧拉方程的 ρ 相对误差。

图16和图17表明，二阶格式的结果比一阶格式精确得多。此外，表5和表6显示，达到二阶精度时，相对误差甚至降低了八倍。此外，二阶HLLC格式最终不会产生异常解。与浅水方程求解器一样，Roe仍然是最佳解。

4.2 平面上的欧拉方程

笛卡尔坐标下的二维欧拉方程 $\vec{x} = (x,\ y)$ 可具有守恒向量 $q = (\rho,\ \rho u,\ \rho v,\ E)^T$，其分量分别表示密度 ρ、能量 E 和两个笛卡尔速度分量 $\vec{u} = (u,\ v)^T$，且均为空间和时间的函数。通量矩阵的形式为：

$$F(q) = \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & rhouv \\ \rho uv & \rho v^2 + p \\ u(E+p) & v(E+p) \end{pmatrix}$$

我们采用的初始条件是 Liska & Wendroff（2003）论文中的二维黎曼问题之一 [27], 该问题也是 Pyclaw 中的示例，并设置了壁面边界条件。图 18 展示了在网格上获得的密度，其中包含 400×400 个网格单元，并使用 Rusanov、HLL 和 HLLE 数值通量运行了 0。$3s$。我们在二阶求解器中使用了与浅水方程相同的斜率通量限制器。此外，参考解基于细网格上的 HLLE SharpClaw 求解器 1000×1000 从 Pyclaw 获得。

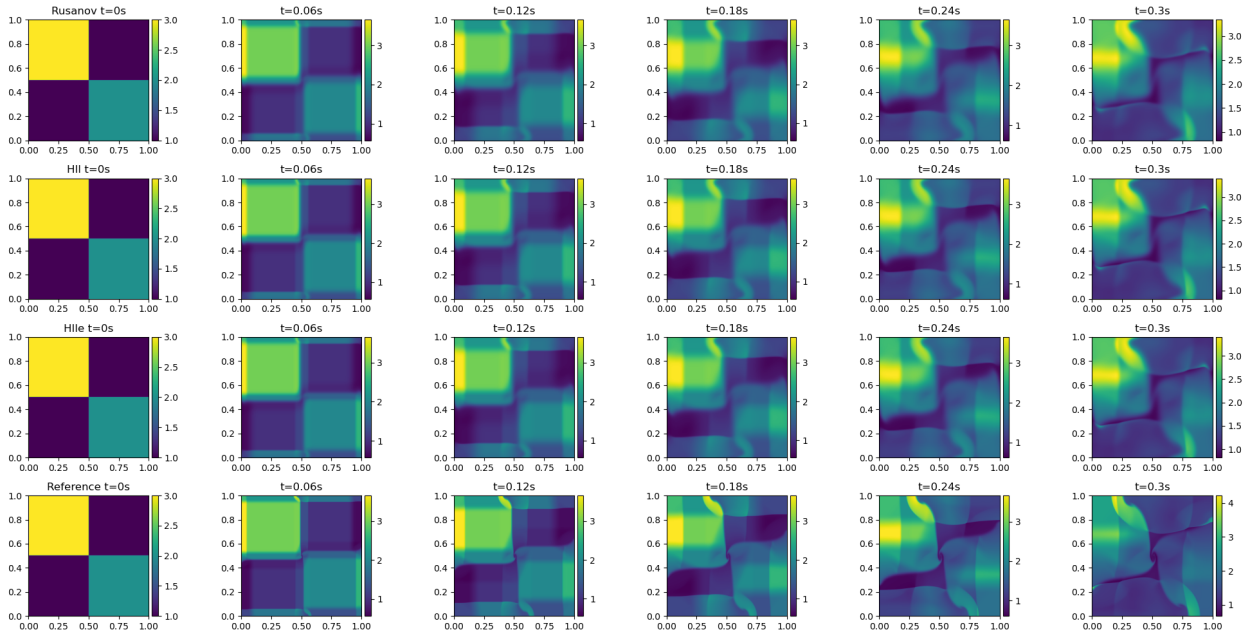


图 18：使用不同数值通量计算的各测试算例中，分辨率为 400×400 的一阶二维欧拉黎曼问题的解。

误差 Flux / nx	Rusanov	Roe	HLL	HLLE
100×100	10.99 %	6.35 %	10.46 %	10.44%
200×200	8.72%	4.29%	8.05%	9.05%
400×400	6.43%	2.58 %	5.82 %	5.85 %

表 7：使用不同数值通量和加密网格的一阶格式二维欧拉方程 ρ 相对误差。

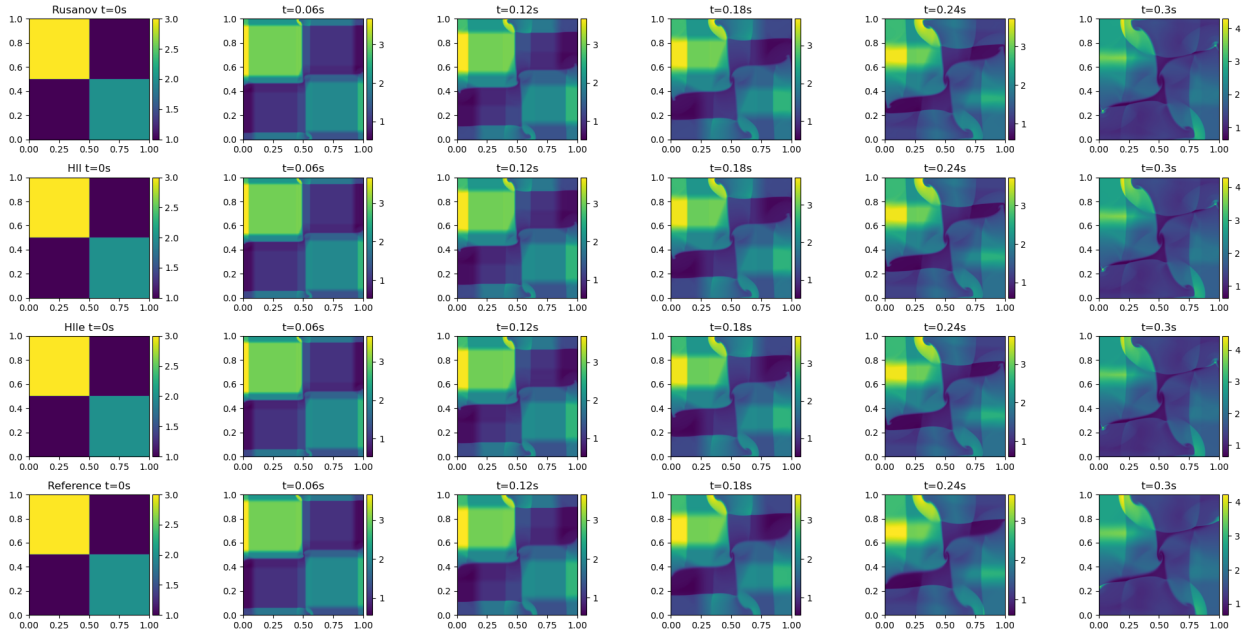


图 19：使用不同数值通量求解测试案例的二阶二维欧拉黎曼问题，分辨率为 400×400。

误差 Flux / nx	Rusanov	Roe	HLL	HLLE
100×100	4.89 %	4.60 %	4.63 %	4.65 %
200×200	2.76 %	2.71 %	2.76 %	2.79 %
400×400	2.08 %	2.04 %	2.14 %	2.16 %

表8：采用不同数值通量和加密网格时，二维欧拉方程 ρ 的二阶格式相对误差。

比较图18和图19可知，二阶求解器的解明显比一阶求解器更清晰，这与表7和表8的结果一致。与3.4节中的浅水方程求解器一样，我们的求解器计算速度远快于 plaw 求解器。

5 浅水方程的嵌入式数据驱动求解器

本章采用2.52和2.53中介绍的四种数据驱动方法来实现基于机器学习的浅水方程求解器。此外，使用总能量和位涡来检验经典求解器和数据驱动求解器的能量耗散 [33], 其表达式如下：

$$\begin{aligned} potenial &= 0.5 * gh^2 \\ kinetic &= 0.5 * (u^2 + v^2 + w^2) * h \\ energy &= potential + kinetic \end{aligned} \tag{5.1}$$

5.1 直接通过卷积神经网络计算边界通量

球面上的浅水方程

我们使用球面柏林噪声 [29]，生成了一个包含 300 个不同初始条件、分辨率为 1000×500的训练集。守恒变量的数量级与罗斯贝-豪尔维茨波一致。随后，将每个初始案例在 Pyclaw 中运行 0.5 天。最终，我们获得了总数为 200,000 个训练集，其中每个状态都稳定且连续。图20展示了一个示例。

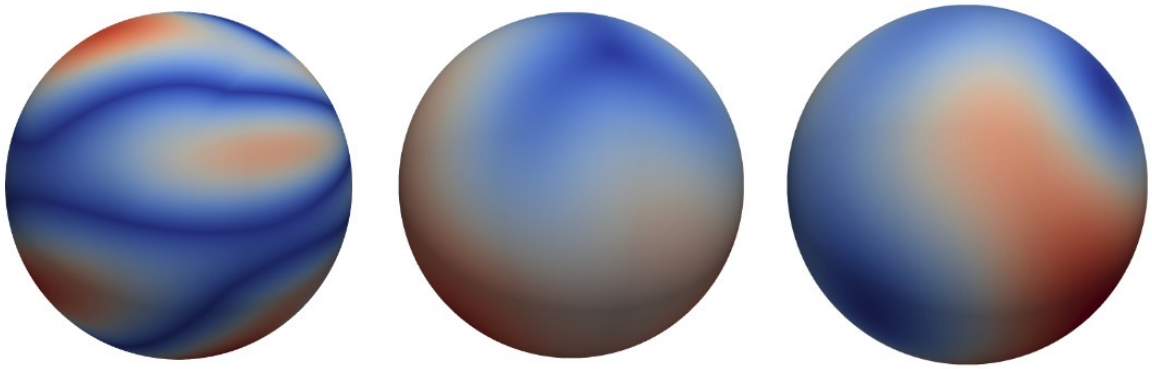


图20：通过球面上的柏林噪声生成的球面训练集。

然后，我们使用 Roe格式计算粗化后单元的通量，如图25所示。“original”和“center”分别表示高分辨率和粗化后的低分辨率下各单元的通量。

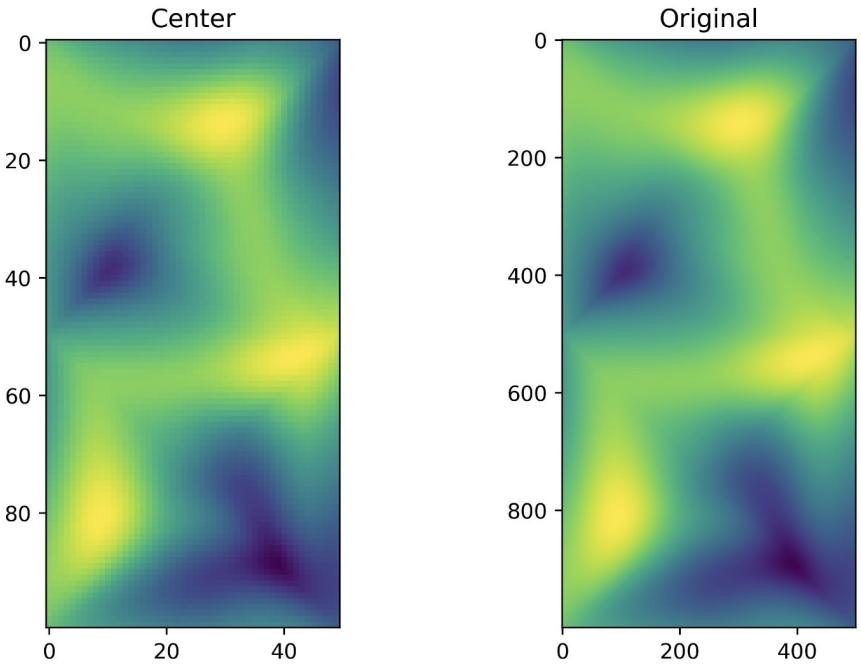


图21：在计算网格上，将单元通量从高分辨率网格插值到低分辨率网格。

经过200个训练轮次后，我们将神经网络嵌入经典求解器中，

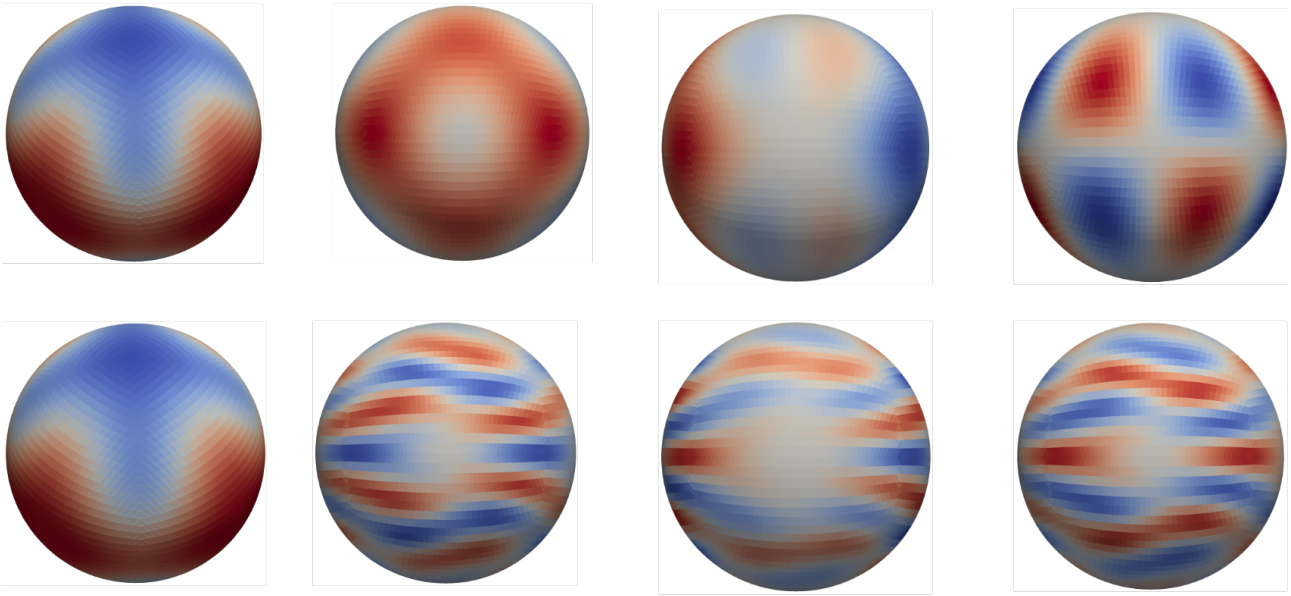


图22：上方四张图为初始值的守恒向量（h， hu， hv， hw）^{T}。下方四张图为使用第一种神经网络方法运行一天后的结果。

如图 22 所示，求解器运行一天后，水深 h 保持不变，而其余动量分量（hu, hv, hw）^{T}的结果完全错误。

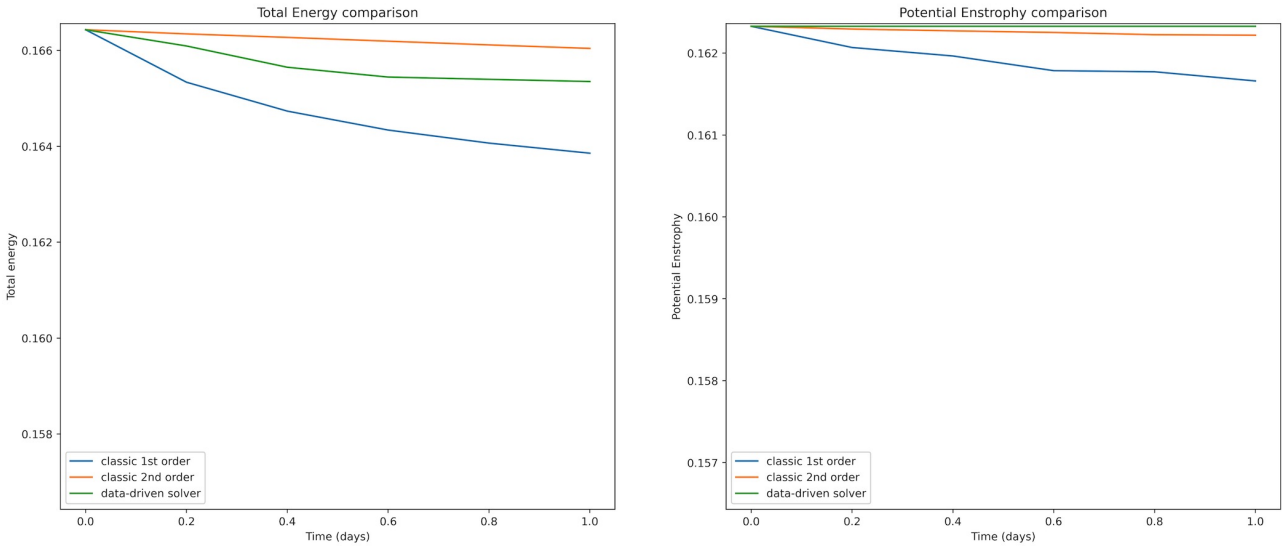


图23：在 100×50 网格上使用第一种神经网络方法进行罗斯比-豪威茨波试验时，总能量守恒（左）和位涡（右）的演变。

经过能量测试后，我们可以看到总能量损失介于一阶精度与二阶之间，而位涡保持不变。尽管能量守恒表现良好，但这种神经网络结构并不可行。

5.2 计 算边界状态（使用卷积神经网络生成的线性系数） s

球面上的浅水方程

这里我们使用与上一节5.1中相同的训练集。然后，我们尝试了两种插值方法（SciPy [31]和xESMF [32]），以便在网格粗化后获得四个边界值。图24展示了将我们的网格从 1000×500 粗化到 100×500 （因子为十）的结果。“中心”表示低分辨率网格中的中心值，“左”、“上”、“右”和“下”表示四个边界值，“原始”表示高分辨率网格中的中心值。

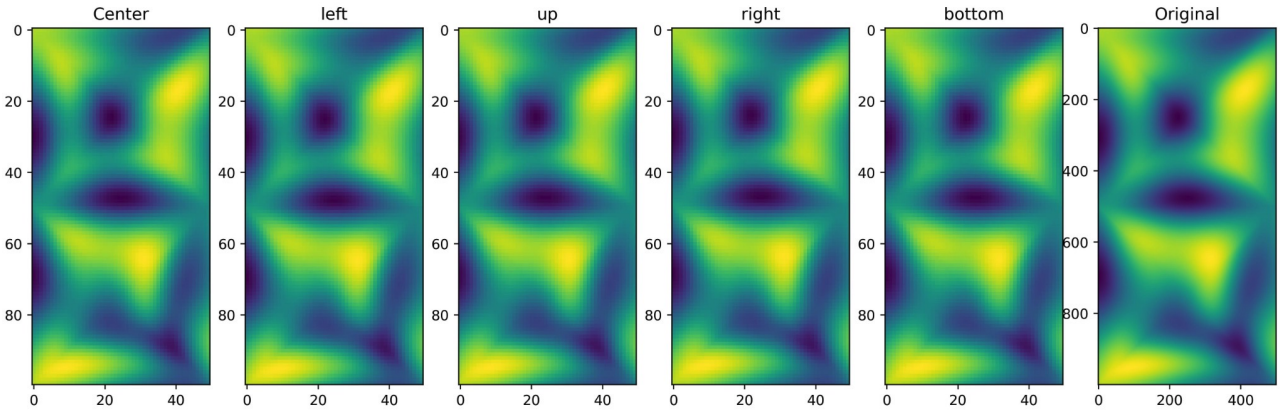


图24：在计算网格上对高到低分辨率网格插值中心值。

之后，我们将粗化后的低分辨率网格的中心值作为输入值，并将四个边界作为输出值。训练200 个训练轮次后，我们将神经网络嵌入经典求解器中。

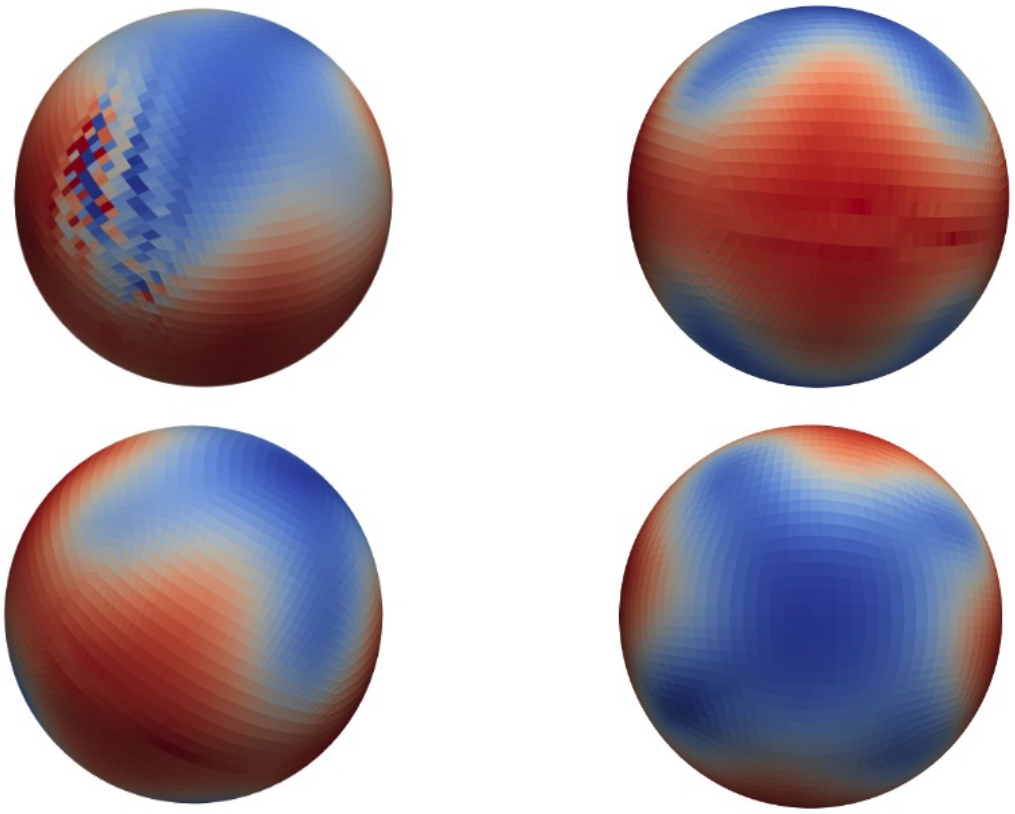


图25：上方两幅图是未使用通量限制器的 Roe格式结果。下方两幅图是使用通量限制器的 Roe格式结果。

如果不使用 2.53 中的通量限制器，结果就会不稳定。运行 0.5 天后，Roe格式会产生无效输出 Nan。这可能是因为神经网络得到的解中存在负高度项，导致计算数值通量时结果为 Nan。

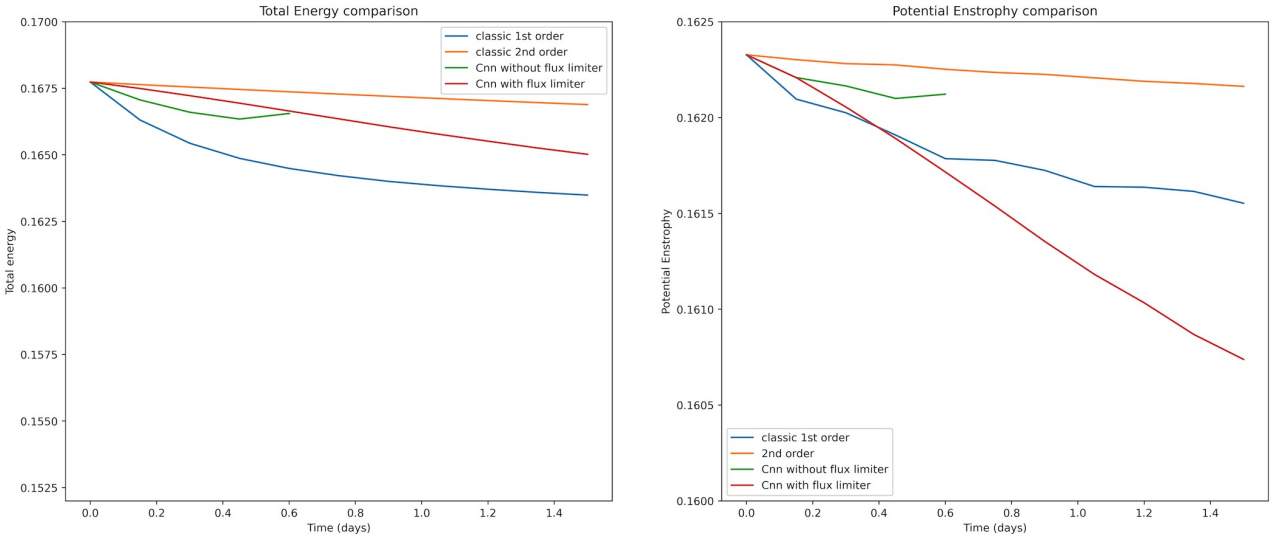


图26：使用第二种神经网络方法进行 Rossby–Haurwitz 波测试时， 100×50 网格上总能量守恒（左）和位涡（右）的演变。

经过能量测试后，采用 100×50 分辨率的经典一阶求解器会在一天后将总能量从0.1674降至0.1638，二阶求解器则降至0.1671。与此同时，

不带通量限制器的神经网络从0.5天起出现无效值。此外，带通量限制器的神经网络在总能量方面表现更好，但势熵损失严重。这表明更多势能被转化为动能，且求解器误差变得不准确，这与图25中的结果一致。

我们还在下面的图27中给出了一个示例。“中心值”表示一阶精度下使用的单元平均值。“重构值”是使用斜率限制器重构到界面的值，而“卷积神经网络”表示使用卷积神经网络重构得到的边界值。此外，“真实值”表示我们使用连续初始条件得到的边界实际值。

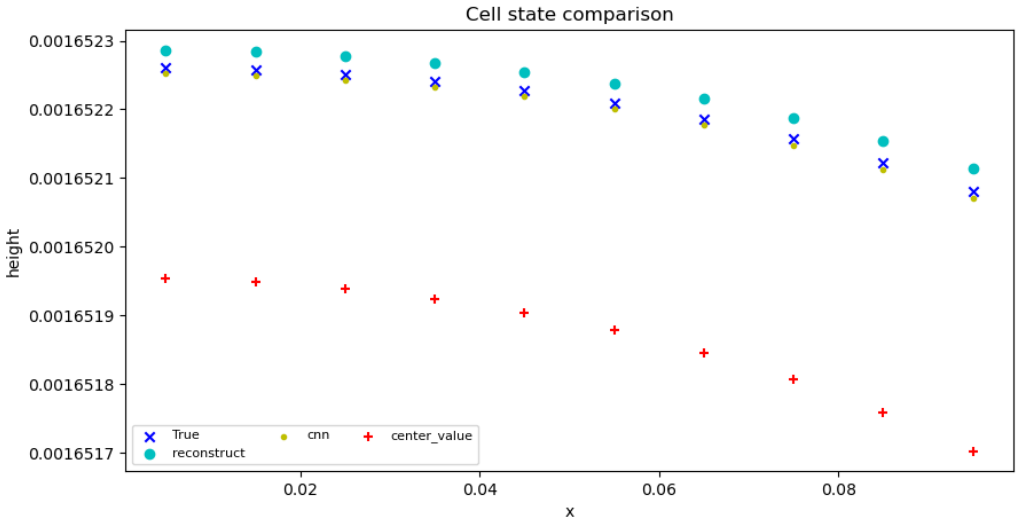


图27：使用不同重构方法得到的单元状态对比。

可以看到，使用卷积神经网络得到的边界值已非常接近真实值，且明显优于使用斜率限制器得到的值。

总的来说，这种神经网络结构并不可行。尽管边界处的插值值接近实际值，但插值结果中可能会间歇性地出现不连续性，从而导致不稳定的输出值。此外，虽然使用通量限制器后结果变得连续，但数值耗散严重，结果甚至不如一阶格式。

5.3 直接通过卷积神经网络计算边界状态

一维

对于一维情形，我们使用高斯过程 [28] 作为训练集的初始条件，生成大量高分辨率（2048）样本，然后使用第3章中的二阶求解器求解，结果如图28所示。随后，我们将所得数值解粗化至低分辨率64（32×）、128（16×）和256（8×），在取其 $nT = 2560$ 步之后。

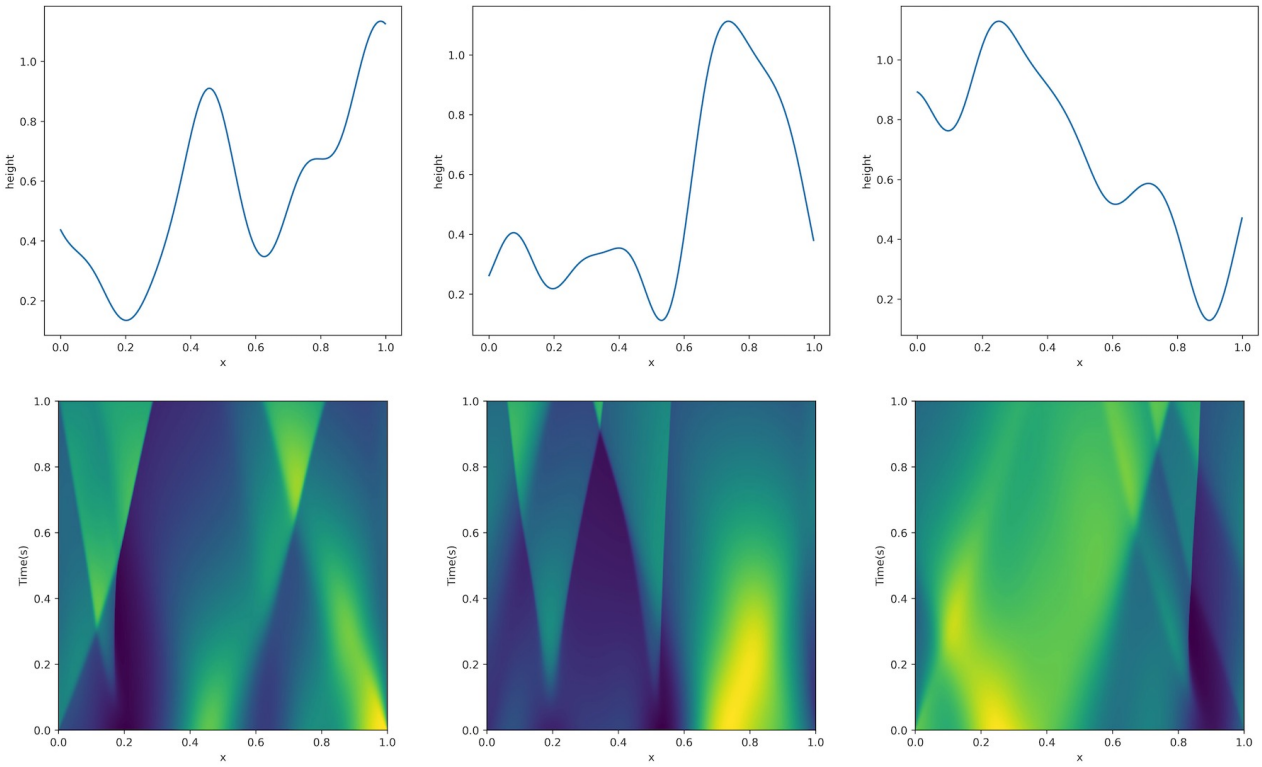


图28：由一维高斯过程生成的一维训练集 .

经过20轮训练后，我们分别得到了用于粗化 $8\times$、 $16\times$和 $32\times$的神经网络。神经网络嵌入式求解器的结果如图29所示。

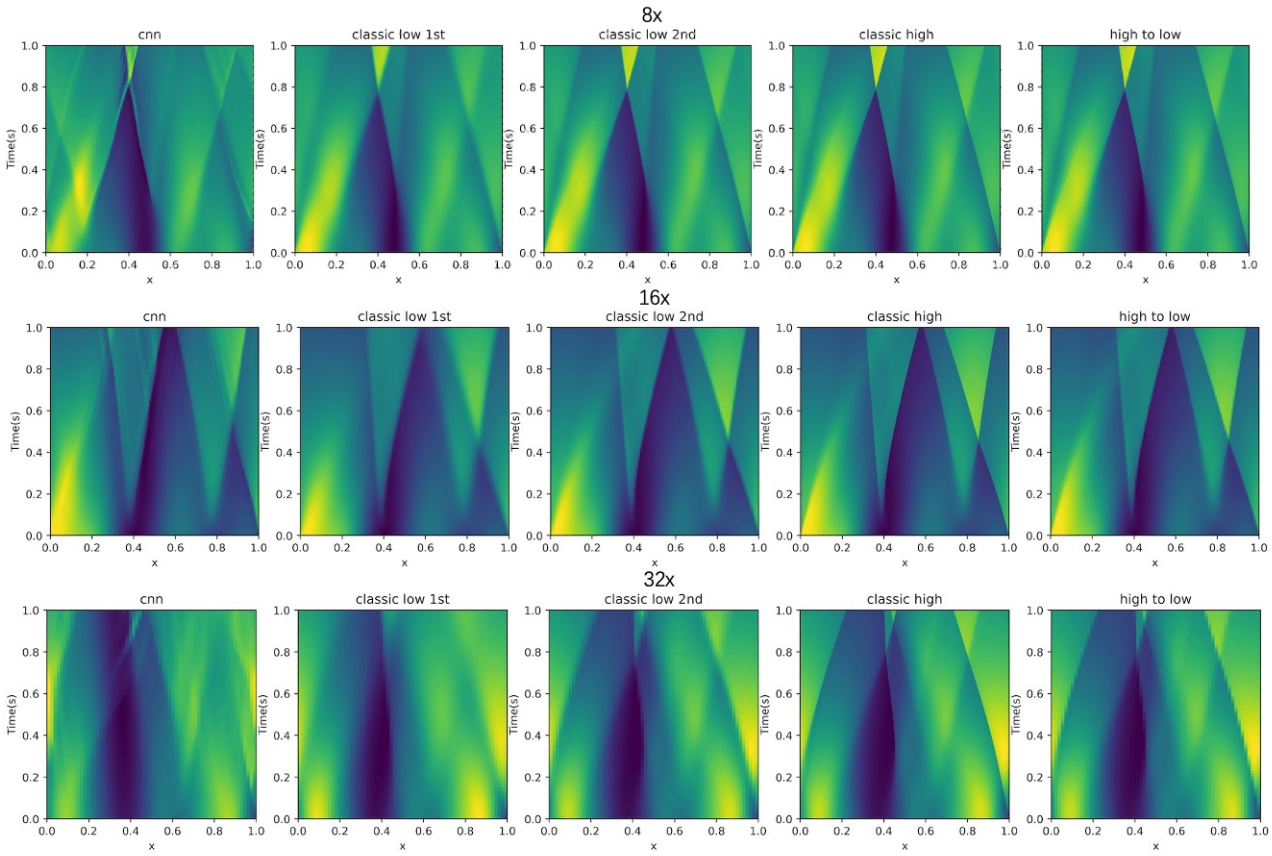


图29：使用第三种神经网络方法的一维机器学习求解器的解。

可以看出，这种基于神经网络方法的求解器运行 1 秒后的整体结果表现良好，但仍有一些区域存在轻微噪声，导致结果不够准确。训练后，我们的目标解是将最右侧的高精度结果粗化到低精度网格所得的解，它比第二个经典求解器的结果更加清晰。

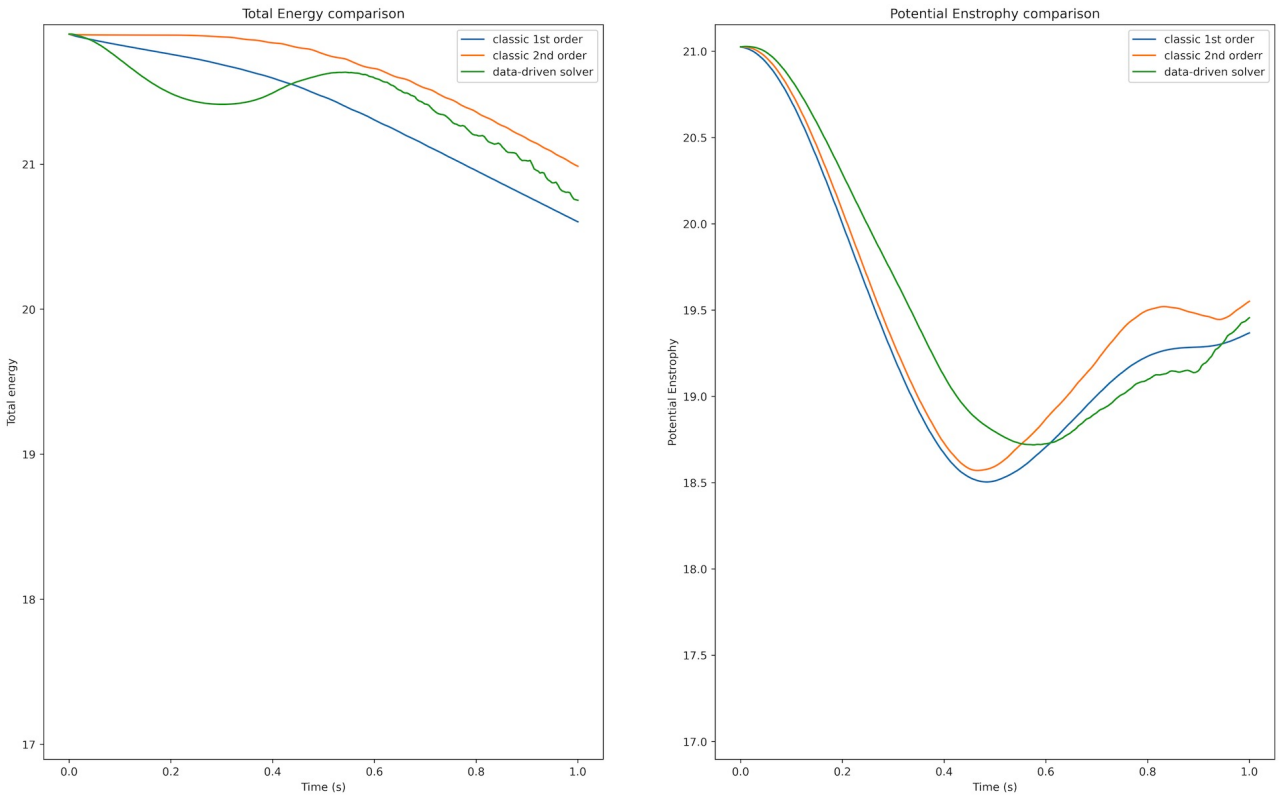


图30：使用第三种神经网络方法时，128 分辨率网格上总能量守恒（左）和位涡（右）的演化。

上图表明，使用卷积神经网络直接输出边界值，比第一种神经网络方法合理得多。总能量和势能与经典的一阶和二阶格式相近。不过，一些噪声可能会导致结果略有偏差，但总体而言，这种方法是可行的。

平面浅水方程

在二维情形中，我们使用二维Perlin噪声 [30] 作为初始条件（图31）。此外，我们确保水深和动量分量都处于合理范围内。我们使用神经网络输出另一个动量项 hv。之后，与一维情形类似，在 1024×1024 分辨率下运行二维求解器，然后将网格粗化8倍和16倍，得到低分辨率网格，并进行2560个时间步。

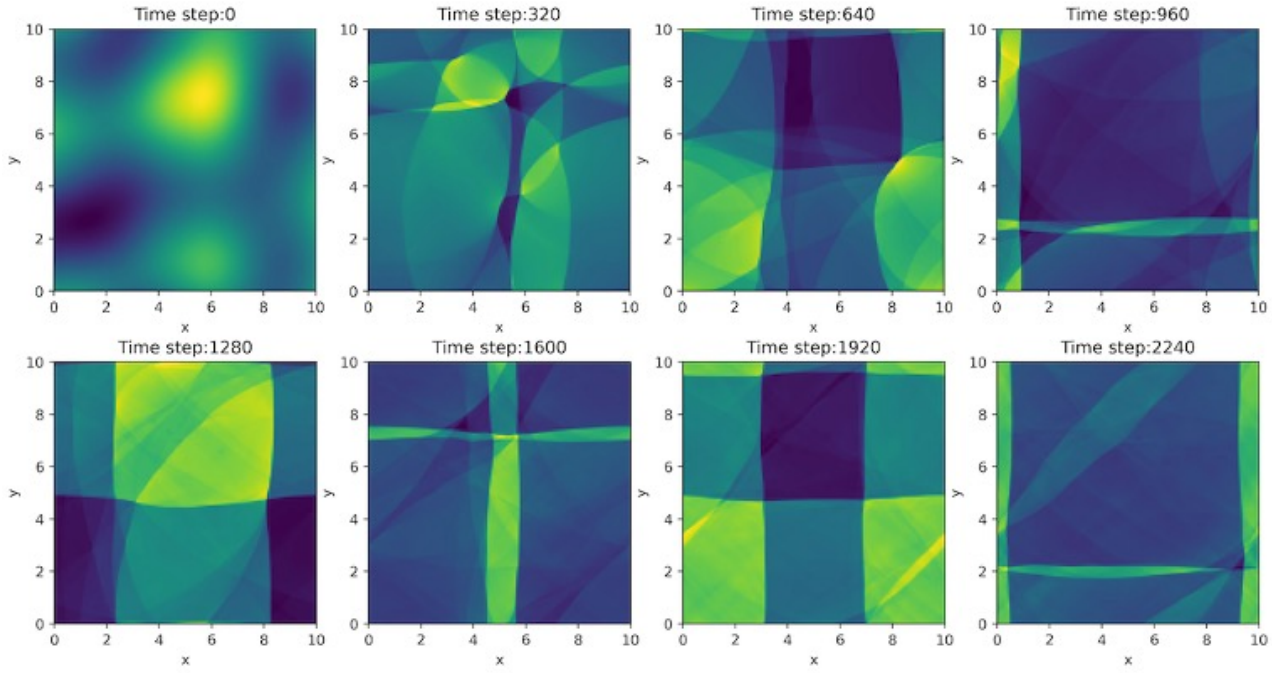


图31：由二维Perlin噪声生成的二维训练集。

请注意，对于二维神经网络，我们使用卷积神经网络直接将每个单元的中心值重构到四个边界上，即 Q_i^l, Q_i^u, Q_i^r, Q_i^b。然后我们使用 Q_i^l, Q_{i-1}^r和 Q_i^b,Q_{i-1}^u来计算通量 F_i^l和 F_i^b。根据有限体积法守恒定律，我们可以得知$F_i^r = -F_{i+1}^r$和 $F_i^u = -F_{i+1}^b$。

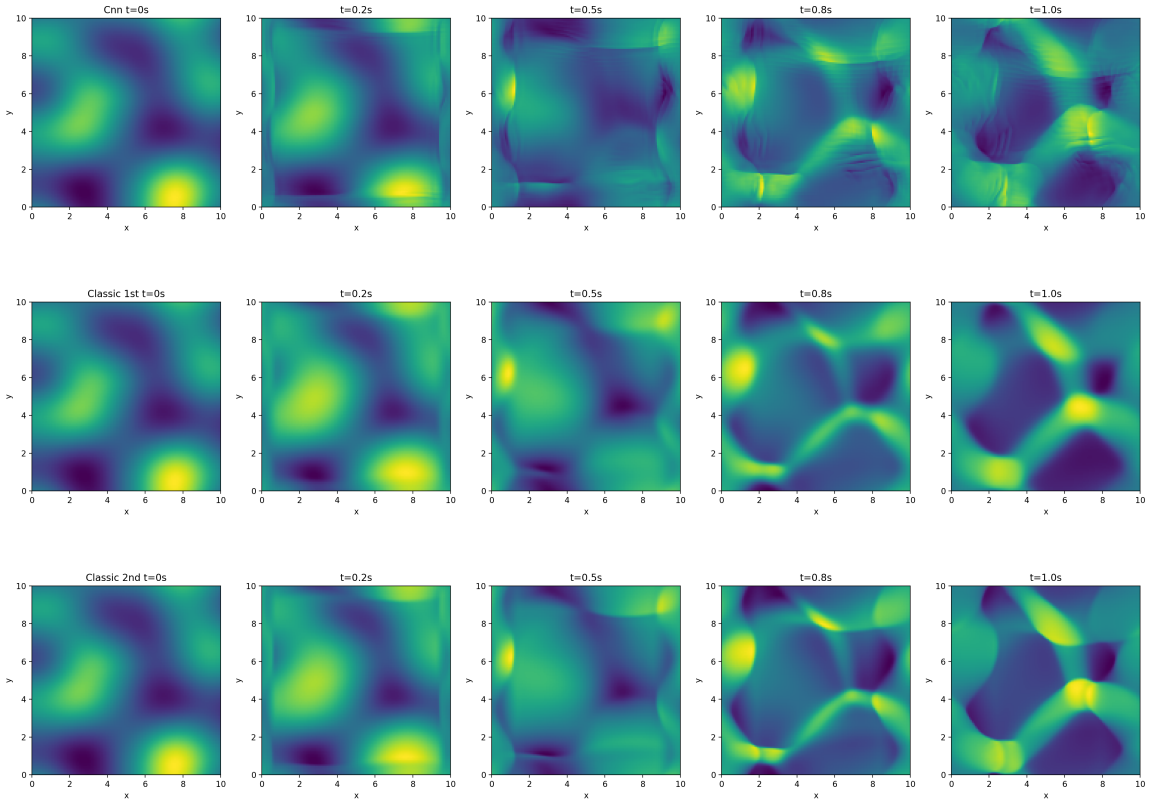
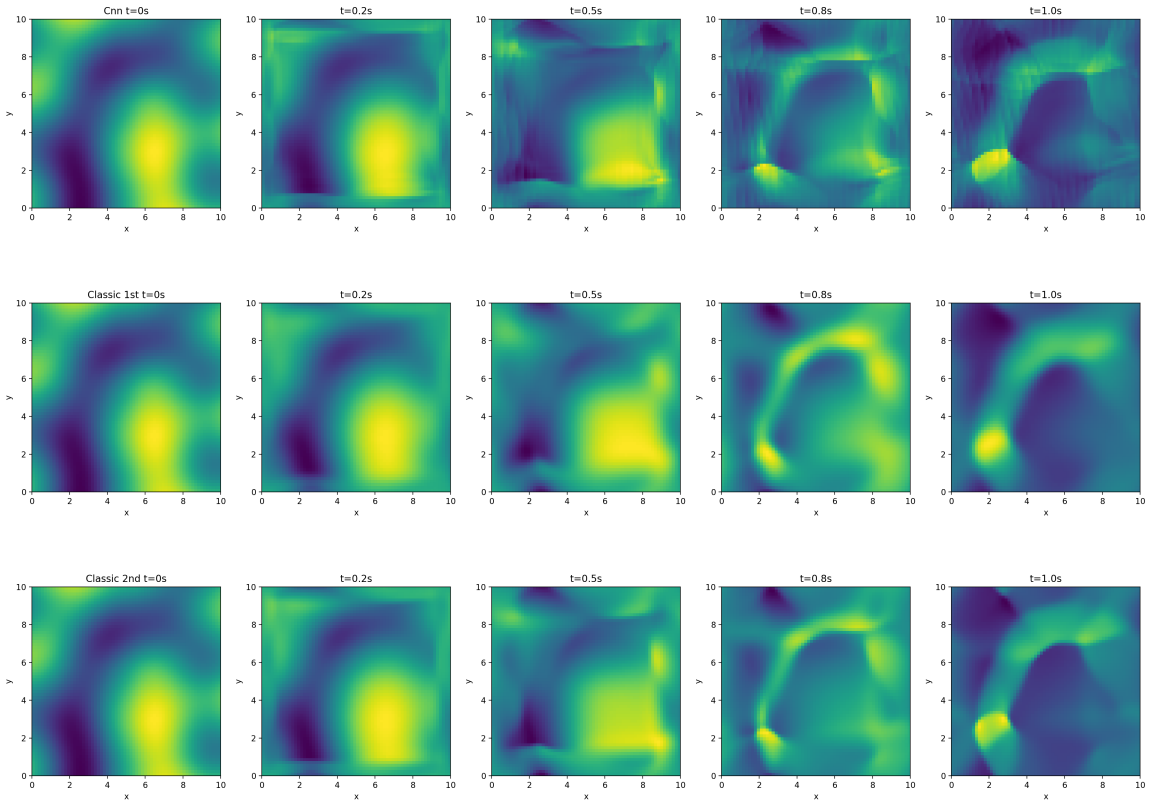


图32：使用第三种神经网络方法的二维机器学习求解器的解（8×）。



Fig图 33：使用第三种神经网络方法的二维机器学习求解器的解（16×）　　.

在使用十种不同的初始条件训练十个训练轮次后，我们发现结果与一维情形相似。我们的二维机器学习求解器结果总体上是正确的。图32和图33表明，高分辨率粗化后的解包含更多细节，这与图31中卷积神经网络插值后的解一致，而这些细节无法由低分辨率经典求解器获得。

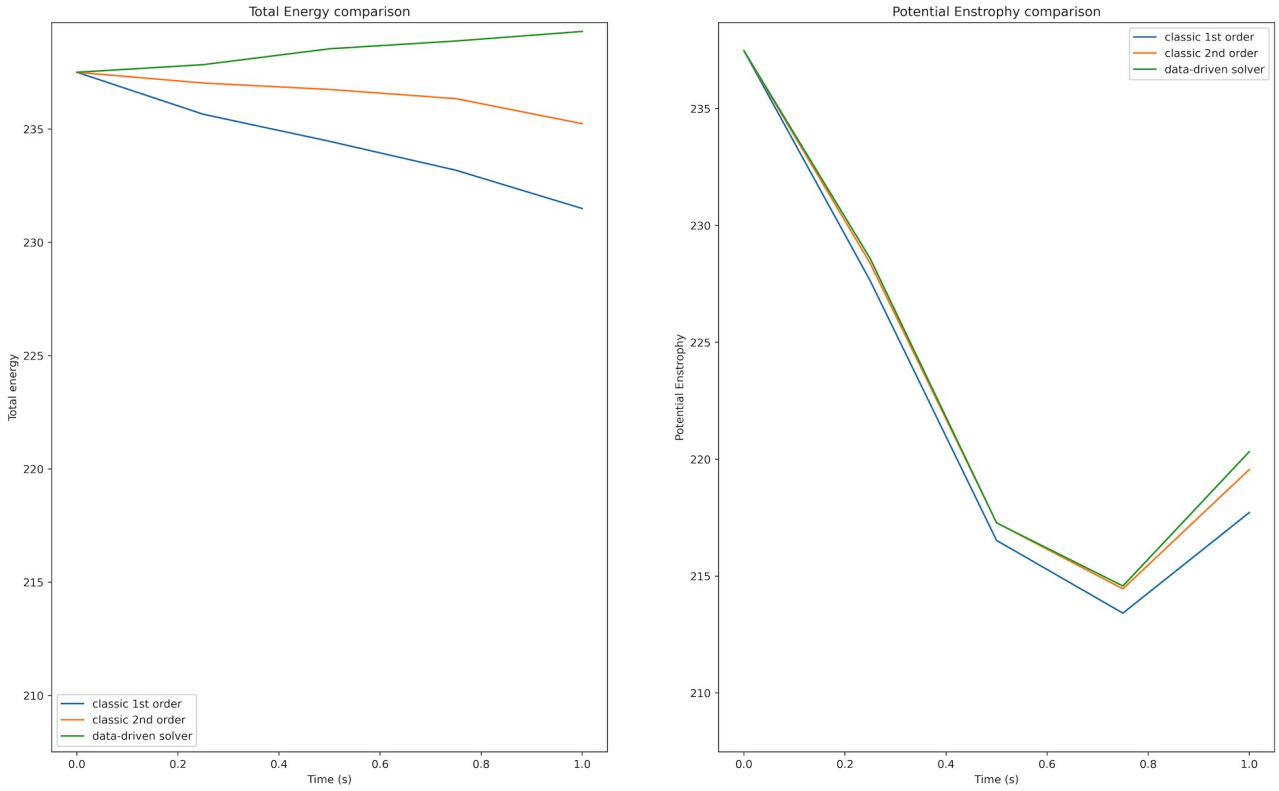


图3：使用第三种神经网络方法时，在 128×128 分辨率网格上总能量守恒（左）和位涡（右）的演化。

上图表明，使用卷积神经网络直接输出边界值，比第一种机器学习方法合理得多。总涡量平方和位涡的变化接近经典的一阶格式和二阶格式。不过，一些噪声可能导致结果略有不准。在增加训练轮次后，或许可以消除这种噪声，从而获得更准确的解。总体而言，这种神经网络方法是可行的。

5.4 使用卷积神经网络生成的线性系数计算重构斜率

本节使用与前文相同的训练集。

一维

经过20轮训练，我们分别得到了用于粗化 $8\times$、 $16\times$和 $32\times$的神经网络。结果如图35所示，

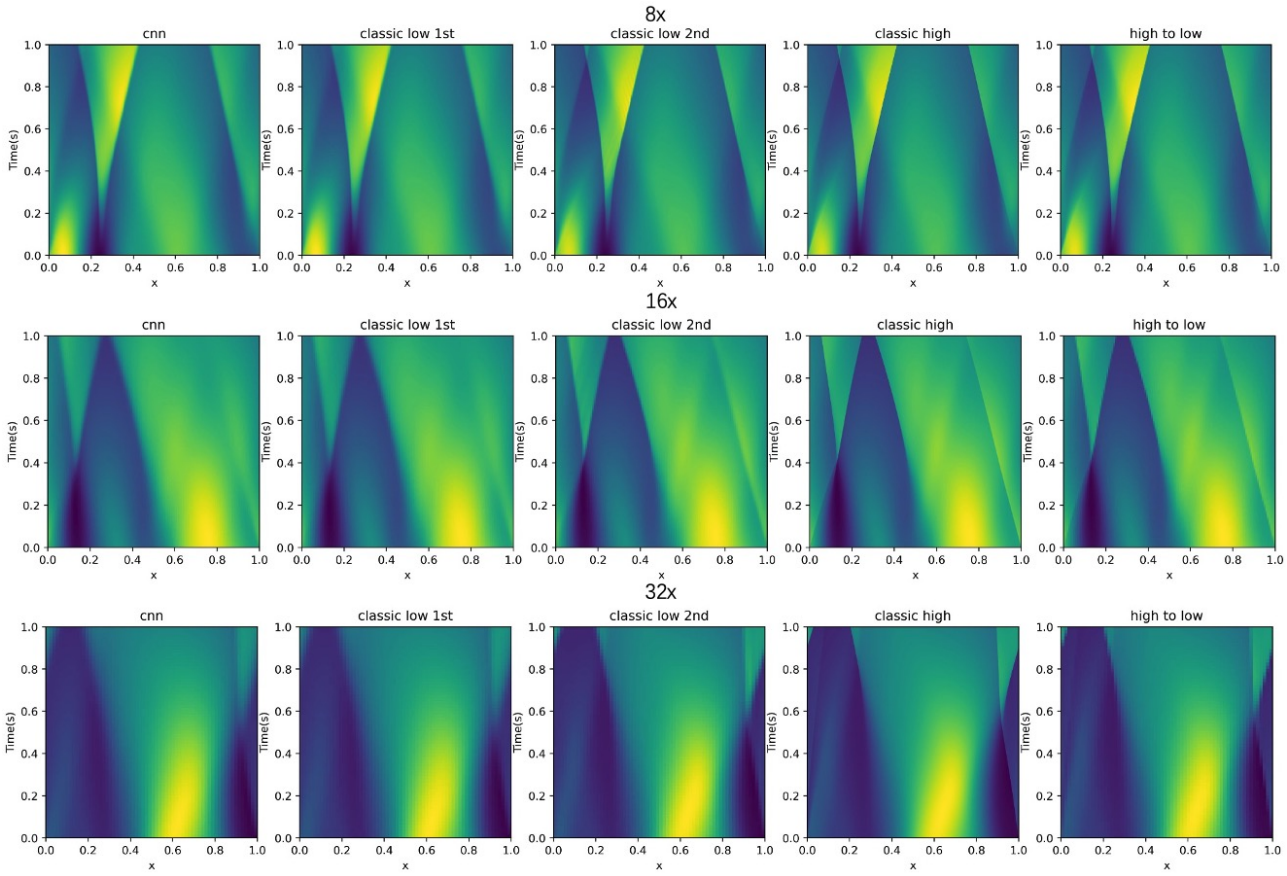


图35：使用第四种神经网络方法的一维机器学习求解器的解。

这表明该神经网络不会产生噪声，且结果与最右侧的目标解非常接近。

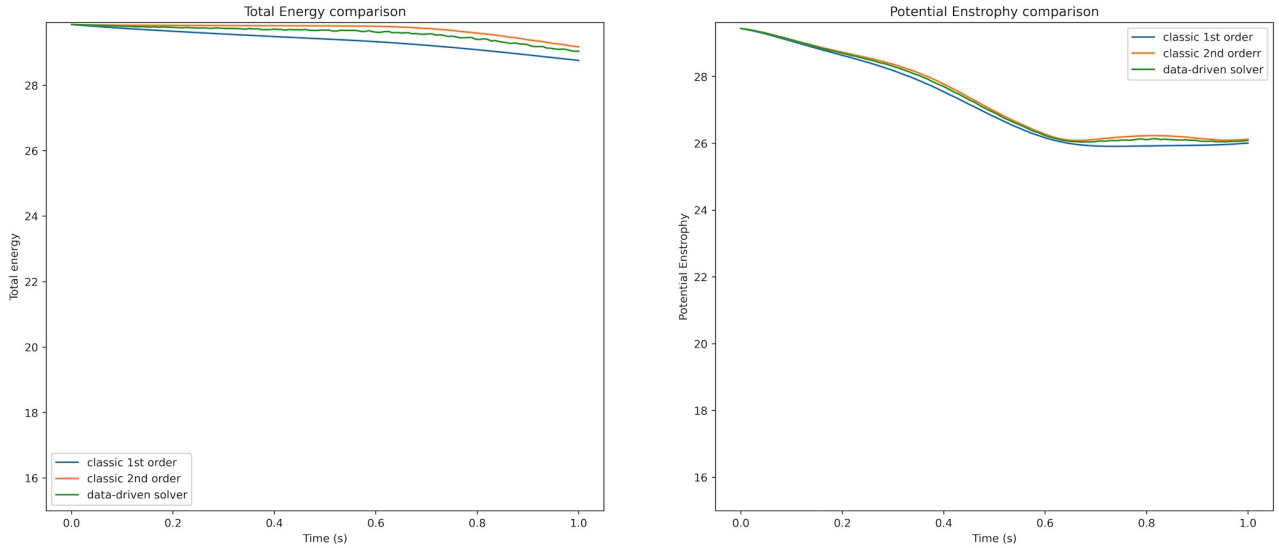


图36：使用第四种神经网络方法在128 分辨率网格上总能量守恒（左）和位涡（右）的演化。

图 35 和图 36 表明，使用卷积神经网络输出边界值的重构斜率是最稳定的方法。其熵变化曲线与经典求解器相似。此外，能量损失优于一阶经典求解器。

平面浅水方程

使用十个初始条件训练四个训练轮次后，将神经网络嵌入以 128×128 分辨率运行 **1** 秒的经典求解器中。与经典求解器相比，我们可以获得，

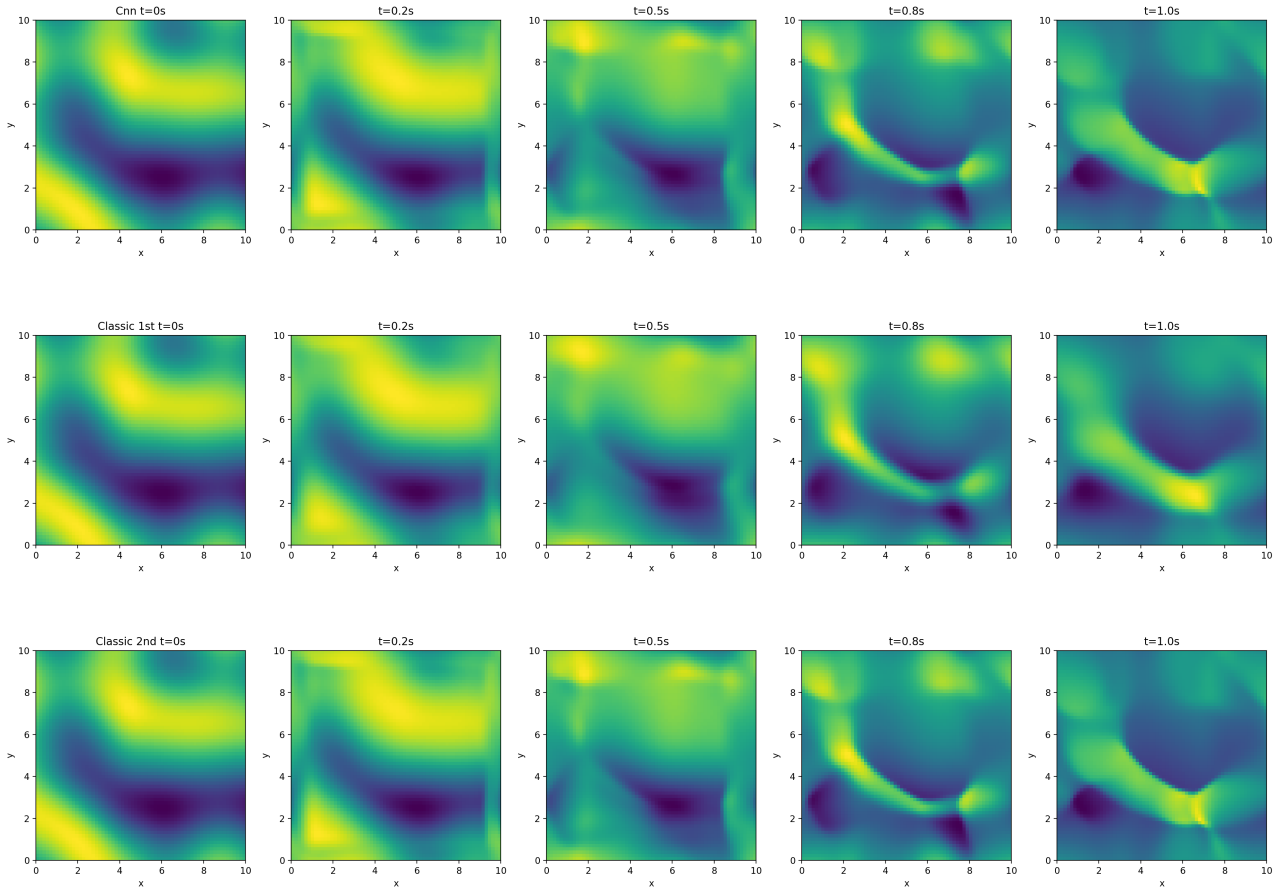


图37：使用第四种神经网络方法的二维机器学习求解器的解（16×）。

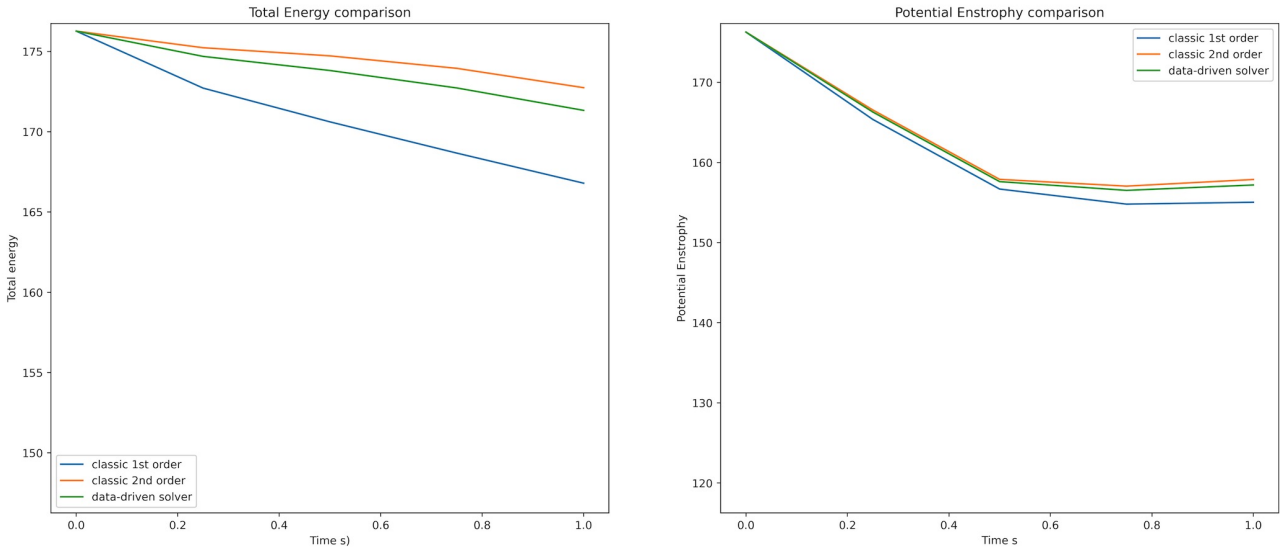


图38：使用第四种神经网络方法在 128×128 分辨率网格上的总能量守恒（左）和位涡（右）的演变。

图37显示，机器学习求解器的结果比经典的一阶求解器更锐利。此外，基于机器学习的求解器的位涡已非常接近二阶求解器的结果。

球面上的浅水方程

在球面情形下，我们使用神经网络为 hw 多输出一组 alpha。训练四个训练轮次后，将得到的神经网络嵌入经典求解器，并在 250×125 分辨率下运行 3 秒，得到以下结果。

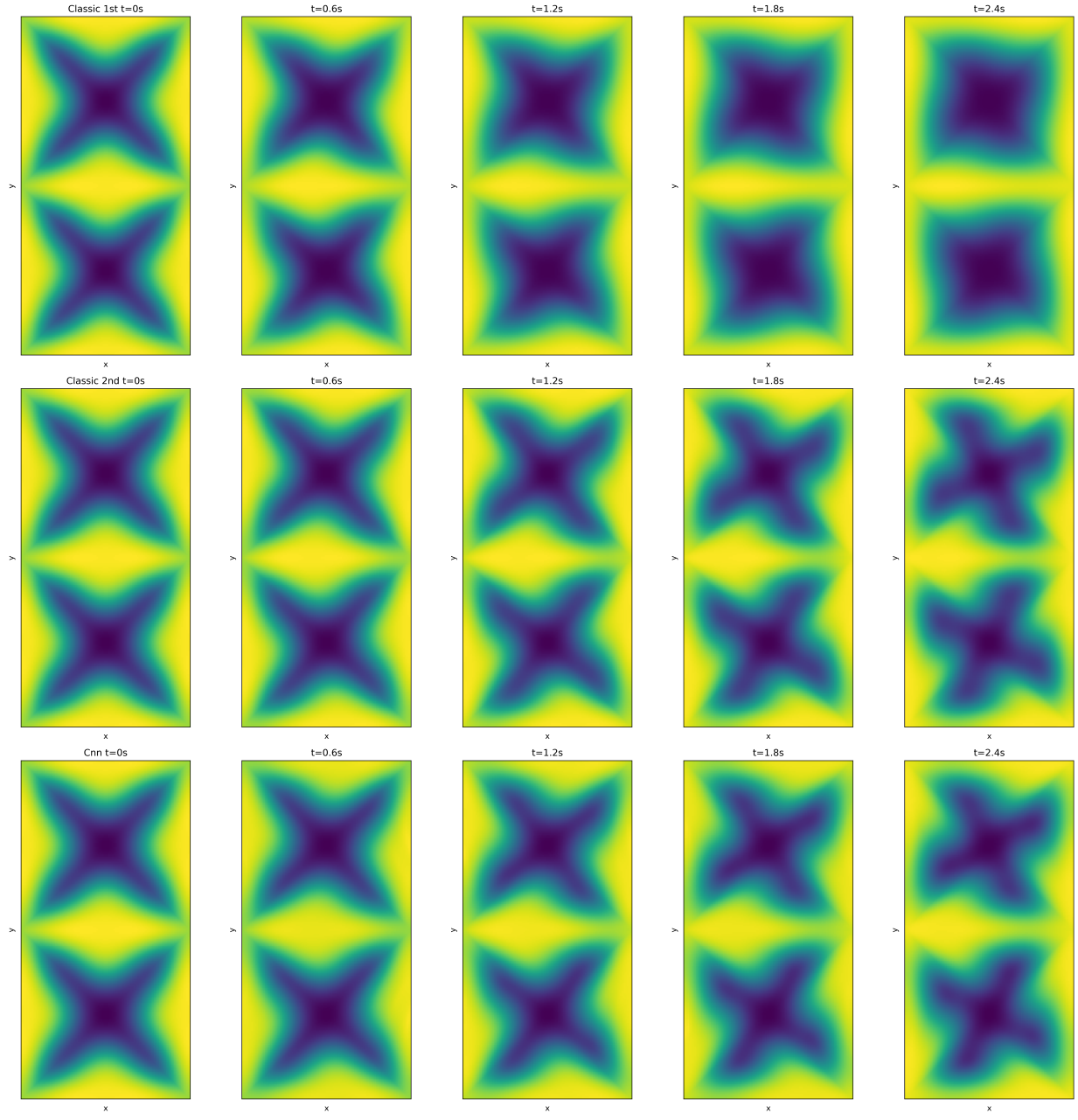


图39：在计算网格上使用第四种神经网络方法（10×）得到的球面机器学习求解器的解。

运行三天后，我们发现，由于精度不足，经典的一阶结果变得模糊，但二阶求解器和基于机器学习的求解器表现良好。此外，机器学习求解器性能稳定，且没有噪声。

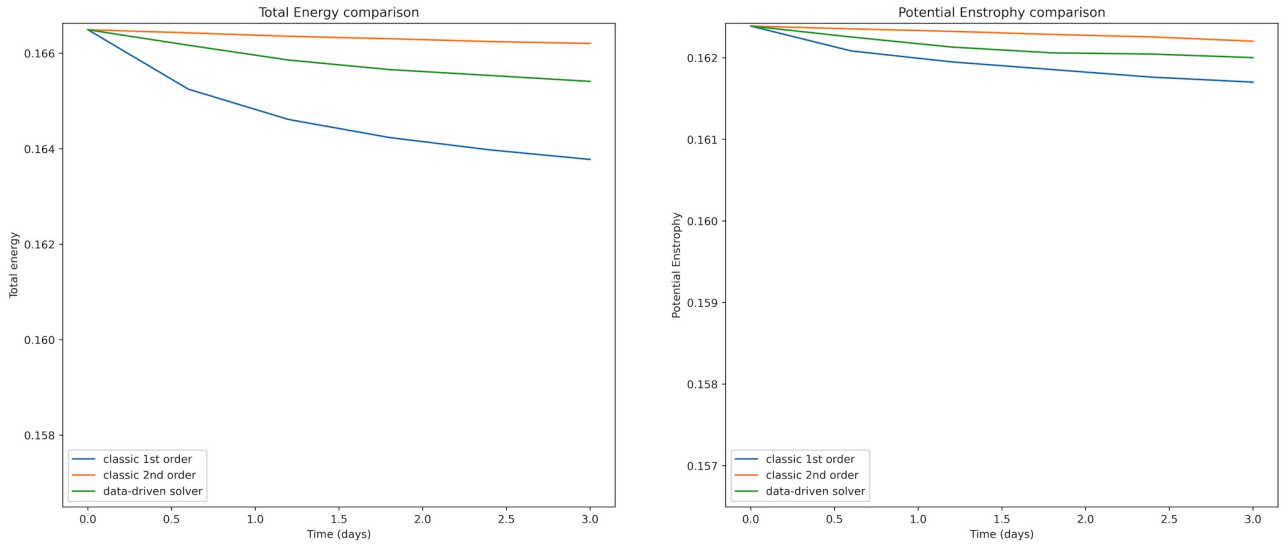


图40：在 250×125 分辨率网格上使用第四种神经网络方法时，总能量守恒（左）和位涡（右）的演化。

图40中的能量损失与图39中的结果一致。经典的一阶求解器的能量损失最严重，导致其解模糊不清。机器学习求解器的能量损失介于一阶和二阶之间，结果准确。在四种方法中，这种神经网络方法表现最佳，结果稳定且准确。如果增加训练时间和训练集数量，结果可能会优于二阶精度求解器。

6 结论与未来工作

6.1 结论

为总结本学位论文，我们在torch和Dace框架下实现了一阶和二阶的一维/二维欧拉方程与浅水方程，以及球面浅水方程，其中包括若干数值通量格式和通量限制器。经测试，我们的经典求解器表现远优于Pyclaw求解器。随后，我们尝试将机器学习嵌入经典求解器。尽管已有大量工作提出基于数据驱动的加速计算流体力学求解器，基于浅水方程的模拟仍有待探索。我们已为浅水方程开发了四种基于数据驱动的求解器。

实验表明，第一种直接使用卷积神经网络输出通量的方法不可行。此外，第二种神经网络方法采用学习系数法计算边界值，由于插值得到的边界值存在间歇性不连续，导致结果不稳定，表现不佳。在第三种方法中，我们使用卷积神经网络直接输出边界值。由于训练轮次不足或训练集有限，可能存在一些噪声，但该方法总体上可行。在第四种方法中，我们用卷积神经网络计算重构斜率以生成线性系数，其性能最佳，并且至少能保证一阶精度。

6.2 未来工作

在输入和模型架构等方面仍有进一步改进的空间。未来，首先要将训练集扩大，并将训练周期延长至第二个

方法。我们的方法基于用于非线性插值的 WENO方法，以获得边界值；理论上可以达到任意精度。

此外，我们还需要找到最佳的模板数量，从理论上讲，这可以提高精度。在此前的实验中，我们使用了三个模板。我们还可以采用学习校正方法 [9], ，其理念更接近大涡模拟，用于实现机器学习求解器，该求解器使用卷积神经网络直接输出通量或单元状态。我们尝试过这种方法，但尚未解决稳定性问题。

此外，我们可以继续将数据驱动方法嵌入欧拉方程中。欧拉方程要求密度和能量项为正，因此还应考虑更多约束。

数据可用性

我们模型的源代码（包括学习组件）以及训练集生成器可在 GitHub 上获取（ https://github.com/baaiy0610/master-thesis）。

插图目录

1 展示单元平均值和通量的典型有限体积网格.10 2 单元平均值在每个界面处定义黎曼问题.12 3 左图为计算网格，右图为球面网格.17 4 第一种神经网络结构：卷积神经网络直接输出边界数值通量。 . . . . . 18 5 第二种神经网络结构：利用卷积神经网络生成的线性系数计算边界状态.19 6 第三种神经网络结构：利用卷积神经网络直接计算边界状态。 . . . . . 20 7 第四种神经网络结构：利用卷积神经网络生成的线性系数计算重构斜率 .21 8 使用不同数值通量求解测试案例的一阶一维浅水方程黎曼问题，分辨率为 128×128 .22 9 使用不同数值通量求解测试案例的二阶一维浅水方程黎曼问题，分辨率为 128×128 .23 10 使用不同数值通量求解测试案例的二维浅水方程，分辨率为 400×400 .24 11 由二维Perlin噪声生成的水床.25 12 分辨率为 400×400时带水床的二维浅水方程解.25 13 以罗斯贝-豪威茨波为初始数据的球面浅水方程。图中展示了流体高度 h 。上方两幅图（时间分别为t=0 和1天）从上方观察；下方两幅图（时间分别为t=2 天和4天）从下方观察. 27 14 比较不同框架下二维浅水方程求解器的性能。分辨率范围为 100×100 至 1000×1000.28 15 比较浅水方程球面求解器在使用Rossby–Haurwitz波运行1.5秒时的性能。分辨率范围为 100×50 至 1000×500.28 16 使用不同数值通量求解测试算例的一阶一维欧拉方程，分辨率为 $128 \times$128.29 17 使用不同数值通量求解测试案例的二阶一维欧拉方程，分辨率为128×128 .30 18 使用不同数值通量求解测试案例的一阶二维欧拉黎曼问题，分辨率为 400×400.32 19 使用不同数值通量求解测试案例的二阶二维欧拉黎曼问题，分辨率为 400×400.32 20 由球面上的柏林噪声生成的球面训练集.33 21 在计算网格上将单元通量从高分辨率网格插值到低分辨率网格。 34 22 上方四幅图为初始值的守恒向量（h、hu、hv、hw）^{T}。下方四幅图为使用第一种神经网络方法运行一天后的结果.34 23 使用第一种神经网络方法进行罗斯贝–豪威茨波测试时， 100×50 网格上总能量守恒（左）和位涡（右）的演变.

24 在计算网格上对高分辨率至低分辨率网格插值中心值。 .35

25 上面两幅图是未使用通量限制器的Roe格式结果。下面两幅图是使用通量限制器的Roe格式结果。 36

26 使用第二种神经网络方法进行RossbyâĂŞ豪威茨波测试时， 100×50 网格上总能量守恒（左）和位涡（右）的演化.36 27 使用不同重构方法得到的单元状态比较.37 28 由一维高斯过程生成的一维训练集.38 29 使用第三种神经网络方法的一维机器学习求解器的解.38 30 使用第三种神经网络方法时，128 分辨率网格上总能量守恒（左）和位涡（右）的演化.39 31 由二维Perlin噪声生成的二维训练集.40 32 使用第三种神经网络方法的二维机器学习求解器的解（$8\times$）。. . . . . 40 33 使用第三种神经网络方法的二维机器学习求解器的解（$16\times$）。. . . . . 41 34 使用第三种神经网络方法时， 128×128 分辨率网格上总能量守恒（左）和位涡（右）的演化.42 35 使用第四种神经网络方法的一维机器学习求解器的解.43 36 使用第四种神经网络方法时，128 分辨率网格上总能量守恒（左）和位涡（右）的演化.43 37 使用第四种神经网络方法的二维机器学习求解器的解（$16\times$）。. . . . . 44 38 使用第四种神经网络方法时， 128×128 分辨率网格上总能量守恒（左）和位涡（右）的演化.44 39 在计算网格上使用第四种神经网络方法的球面机器学习求解器的解（$10\times$）.45 40 使用第四种神经网络方法时， 250×125 分辨率网格上总能量守恒（左）和位涡（右）的演化.46

表格目录

1 一维浅水方程 h 采用不同数值通量的一阶格式及逐渐加密网格的相对误差。.222 一维浅水方程 h 采用不同数值通量的二阶格式及逐渐加密网格的相对误差。.233 二维浅水方程 h 采用不同数值通量的二阶格式及逐渐加密网格的相对误差。.244 采用三个网格的罗斯贝-豪威茨波，网格单元数分别为 100×50、 200×100 和 400×200，并采用不同的数值通量格式。表中列出了 h 第1天的相对误差。.275 一维欧拉方程 ρ 采用不同数值通量的一阶格式及逐渐加密网格的相对误差。.306 一维欧拉方程 ρ 采用不同数值通量的二阶格式及逐渐加密网格的相对误差。.317 二维欧拉方程 ρ 采用不同数值通量的一阶格式及逐渐加密网格的相对误差。.328 二维欧拉方程 ρ 采用不同数值通量的二阶格式及逐渐加密网格的相对误差。.33

参考文献

[1] KÃďmpf Jochen（2009）《初学者海洋模拟：使用开源软件》。柏林：Springer。[2] LeVeque, R.J.（2011）《双曲型问题的有限体积法》。剑桥：剑桥大学出版社。[3] Mishra, S.（2018）《用于加速微分方程计算的数据驱动机器学习框架》，《工程数学》，1(1)，第118–146页。网址：https://doi.org/10.3934/mine.2018.1.118。[4]Bar-Sinai, Y. 等（2019）《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)，第15344–15349页。网址：https://doi.org/10.1073/pnas.1814058116。[5] Magiera, J. 等（2020）《用于黎曼问题的约束感知神经网络》，《计算物理学杂志》，409，文章编号109345。网址：https://doi.org/10.1016/j.jcp.2020.109345。[6] Maddu, Suryanarayana 等。“STENCIL-NET：偏微分方程的数据驱动、自适应解离散化。” arXiv预印本 arXiv:2101.06182（2021）。[7] Shu, C.-W.（1998）《双曲守恒律的本质无振荡格式与加权本质无振荡格式》，《数学讲义》，第325–432页。网址：https://doi.org/10.1007/bfb0096355。[8]Weyn, J.A.、Durran, D.R. 和 Caruana, R.（2020）《在立方球面上使用深度卷积神经网络改进数据驱动的全球天气预报》。网址：https://doi.org/10.1002/essoar.10502543.1。[9] Kochkov, D. 等（2021）《机器学习加速的计算流体力学》，《美国国家科学院院刊》，118(21)。网址：https://doi.org/10.1073/pnas.2101784118。[10] Vreugdenhil, C.B.（2011）《浅水流动的数值方法》。多德雷赫特：Springer。[11] Tec-Science（2021）《欧拉运动方程（动量守恒）的推导》—tec-science，tec。网址：
h
t
t
p://
w
w
w.
t
e
c-
s
c
i
e
n
c
e.
c
o
m/
mechanics/gases-and-liquids/derivation-of-the-euler-equation-of-motion-conservation-of-momentum/（访问日期：2022年11月22日）。[12] Courant, R.、Friedrichs, K. 和 Lewy, H.（1986）《论数学物理中的偏差分方程》，Kurt Otto Friedrichs，第53–95页，10.1007/978-1-4612-5385-3 7。_[13] Press, WH；Teukolsky, SA；Vetterling, WT；Flannery, BP（2007），“第10.1.2节：Lax方法”，《Numerical Recipes：科学计算的艺术》（第3版），纽约：剑桥大学出版社，ISBN 978-0-521-88068-8[14] Abdoul Kader, M.Y.、Rabé, B. 和 Bisso, S.（2022）《Rusanov格式的一种新变体：β–用于浅水流数值求解的Rusanov格式》，《应用数学国际期刊》，35(4)。网址：https://doi.org/10.12732/ijam.v35i4.7。[15] Ambrosi, D.（1995）《采用Roe黎曼求解器近似浅水方程》，《流体数值方法国际期刊》，20(2)，第157–168页。网址：https://doi.org/10.1002/fld.1650200205。

[16]Toro, E.F.（1997）。HLL与HLLC黎曼求解器。载于：《黎曼求解器与流体动力学数值方法》。Springer，柏林，海德堡。可获取于：https://doi.org/10.1007/978-3-662-03490-3_10

[17] I. Delis, A.（2002）“HLLE黎曼求解器在含源项浅水方程中的改进应用”，《工程数值方法通讯》，19(1)，第59–83页。可获取于：https://doi.org/10.1002/cnm.570。

[18] Kim, D.-H.和Cho, Y.-S.（2004）“基于HLLC近似黎曼求解器的浅水方程分析”，《韩国水资源协会学报》，37(10)，第845–855页。可获取于：https://doi.org/10.3741/jkwra.2004.37.10.845。

[19]Balsara, D.S.（2012）“守恒律的二维 HLLC 黎曼求解器：在欧拉方程和磁流体动力学流动中的应用”，《计算物理学杂志》，231(22)，第 7476–7503 页。可访问：https://doi.org/10.1016/j.jcp.2011.12.025。[20] Hu, L. 等（2022）“不同斜率限制器对非静力模型中分层剪切流模拟的影响”，《海洋科学与工程杂志》，10(4)，第 489 页。可访问：https://doi.org/10.3390/jmse10040489。[21]Heun 方法（2021）维基百科。维基媒体基金会。可访问：https://en.wikipedia.org/wiki/Heun%27s_method（访问日期：2022年11月23日）。[22]Gottlieb, S. 和 Shu, C.-W.（1998）“总变差递减龙格–库塔格式”，《计算数学》，67(221)，第 73–85 页。可访问：https://doi.org/10.1090/s0025-5718-98-00913-2。[23] Bale, D.S. 和 LeVeque, R.J.（2001）“曲流形上双曲系统的波传播算法”，《双曲问题：理论、数值与应用》，第 129–138 页。可访问：https://doi.org/10.1007/978-3-0348-8370-2_14。[24] Calhoun, D.A.、Helzel, C. 和 LeVeque, R.J.（2008）“圆形和球形区域中偏微分方程的逻辑矩形网格与有限体积法”，《SIAM评论》，50(4)，第 723–752 页。可访问：https://doi.org/10.1137/060664094。[25]Williamson, D.L. 等（1991）“球面几何下浅水方程数值近似的标准测试集”。可访问：https://doi.org/10.2172/5232139。[26] Ben-Nun, T. 等（2022）《天气与气候建模的高效性能工程》，Python，arXiv.org。可访问：https://arxiv.org/abs/2205.04148v1（访问日期：2022年11月23日）。[27] Liska, R. 和 Wendroff, B.（2003）“用于欧拉方程一维和二维测试问题的若干差分格式比较”，《SIAM科学计算杂志》，25(3)，第 995–1017 页。可访问：https://doi.org/10.1137/s1064827502402120。[28]“相关高斯过程”（2006），《马尔可夫过程、高斯过程与局部时》，第 551–579 页。可访问：https://doi.org/10.1017/cbo9780511617997.013。[29]Night（2021）《如何在球面上使用柏林噪声？》，DevForum。可访问：https://devforum.roblox.com/t/how-to-use-perlin-noise-on-a-sphere/1545154（访问日期：2022年11月23日）。

[30]柏林噪声（2022）维基百科。维基媒体基金会。可访问：
https://en.wikipedia.org/wiki/Perlin_noise（访问日期：2022年11月23日）。[31] 插值（scipy.interpolate）#（无日期）《插值（scipy.interpolate）— SciPy v1.9.3 手册》。可访问：https://docs.scipy.org/doc/scipy/reference/interpolate.html（访问日期：2022年11月23日）。[32]地理空间数据通用重网格器（无日期）xESMF。可访问：
https://xesmf.readthedocs.io/en/latest/（访问日期：2022年11月23日）。[33] Chen, C. 和 Xiao, F.（2008）“基于多矩有限体积法的立方球浅水模型”，《计算物理学杂志》，227(10)，第 5019–5044 页。可访问：https://doi.org/10.1016/j.jcp.2008.01.033。