arXiv:2601.17693v1 [physics.flu-dyn] 25 Jan 2026

DDFK：动态无散核流体模拟

Jingrui Xing[1], YIZAO TANG[2], Mengyu Chu[3]*，以及 BAOQUAN CHEN[3]*

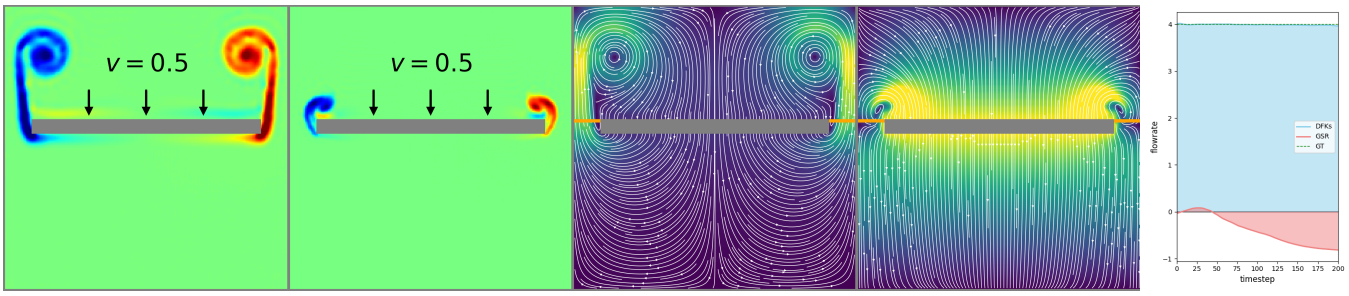


图 1. 活塞在封闭容器内以恒定速度向下运动，分别使用无散核（我们的方法）和 GSR 进行模拟。左侧两图：分别为我们的方法和 GSR 模拟在第 50 帧的涡量场。第三和第四图：分别为我们的方法和 GSR 在第 50 帧的速度场可视化，图中沿速度场绘制了流线，背景颜色表示速度大小。右侧：每个时间步在第三和第四图的橙色线上测得的流量，向上为正。流量通过沿橙色线对速度的 y 分量积分计算得出。

基于内存高效空间表示的流体模拟，例如隐式神经空间表示（INSRs）和高斯空间表示（GSR），其中速度场由神经网络或加权高斯函数参数化，已成为一个新兴研究领域。流体求解器利用了这些空间表示相较于传统离散化的优势，例如空间自适应性和连续可微性，但求解支配流体动力学的时变偏微分方程仍然具有挑战性，尤其是在需要施加无散约束的不可压缩流体中。本文提出一种基于无散核（DFKs）的无网格求解器——动态无散核（DDFKs），用于不可压缩流。每个 DFK 都结合了一个矩阵值径向基函数和一个向量值权重，从而产生无散向量场。我们将连续流速建模为多个 DFK 的总和，从而在保持不可压缩性的同时保留不同层次的细节。定量和定性结果表明，与采用内存高效空间表示的先进方法相比，我们的方法在精度、鲁棒性、涡旋保留能力、时间和内存效率以及对多种现象的适用性方面表现相当，同时在保持不可压缩性方面更为出色。尽管我们的一阶求解器比采用传统离散化的流体求解器更慢，但由于离散化误差降低，我们的方法表现出显著更低的数值耗散。我们通过具有丰富涡旋和多种固体边界条件的各类不可压缩流示例展示了该方法。

1 引言

流体模拟始终是计算机图形学中的一项难题，需要采用表现力丰富的表示方法来捕捉丰富的空间细节，同时支持高效、准确的时间演化，以建模非线性和混沌动力学。传统流体求解器采用基于网格或基于粒子的空间离散化，分别对应欧拉方法和拉格朗日框架；混合方法也得到了广泛提出。这类直接的空间离散化有助于高效计算，从而求解时变流体动力学。

尽管这些方法取得了显著进展，但在表示流场的精细结构时，传统空间离散化很容易遭遇维数灾难。为从根本上解决这一问题，研究者提出了内存高效的空间表示，作为参数化流场的新方法，并由此在流体模拟领域形成一股新兴趋势。Chen 等人 [6] 和 Jain 等人 [16] 使用神经网络作为隐式神经空间表示（INSRs），以取代网格或粒子等显式数据结构。这些方法利用神经网络的连续可微性求解偏微分方程，从而实现具有空间自适应性的流体模拟。此外，Xing 等人 [41] 提出了高斯流体，其采用高斯空间表示（GSR），通过多个加权高斯函数表示速度场。与 INSRs 相比，该方法实现了相当的空间自适应性和内存紧凑性，同时显著提升了空间微分的计算效率和涡量保真度，并具有很高的稳定性。

随着新型内存高效空间表示的提出，在整个模拟过程中维持无散速度场变得颇具挑战。基于隐式神经空间表示的流体求解器采用了传统欧拉方法中常用的算子分裂技术，将纳维-斯托克斯方程分解为平流步骤和投影步骤。在这两种方法中，平流步骤都是通过学习一个拟合平流后速度场的神经网络来求解的，但 [6] 和 [16]采用了不同的投影方法。前一种方法直接优化由投影步骤推导出的 PDE 损失，以学习一个表示压力场的神经网络；这需要计算神经网络的二阶导数，导致计算效率低下。后一种方法利用星上行走法（WoSt）估计每个查询点处的压力梯度，但蒙特卡洛方法的高方差会导致模拟不稳定

* 表示通讯作者。所属机构： [1]北京大学智能科学与技术学院，[2]北京大学电子工程与计算机科学学院， [3]北京大学

在样本有限的情况下。Xing 等人 [41]通过求解纳维-斯托克斯方程的涡量形式，避免了压力投影步骤。然而，为了使流体保持不可压缩，仍需使用物理信息散度损失，这会在优化过程中引入梯度冲突，降低收敛率。更重要的是，即使经过足够多次迭代，散度损失也无法完全降至零，导致流体在某些极端情况下发生严重压缩，如图1所示。

本文提出了一种基于动态无散核（DDFK）的无网格流体求解器，完美解决了在整个模拟过程中保持速度场无散这一挑战，同时保留了内存高效的空间表示方法的所有优势。我们的方法有以下贡献：

- 将无散核（DFK）表示引入流体正向模拟；
- 一种使流体始终不可压缩的新方法；
- 一种采用 DDFK 的无网格求解器，可模拟具有多种固体边界的复杂涡流现象。

2 相关工作

传统流体模拟。自 Stable Fluids [33] 通过引入 Chorin 投影法 [8] 和半拉格朗日平流格式 [28, 29], 奠定了计算机图形学中流体模拟方法的基础以来，研究人员在欧拉方法框架下开展了大量工作，以提升效率和精度。他们通过采用稀疏自适应网格结构 [1, 23, 32]加速压力投影步骤，并开发更精确的平流方法，以减少数值耗散 [18, 30, 42]。还有一些工作将速度场以外的量置于网格上进行平流，例如冲量 [11], 、协向量 [25], 和涡量 [21, 35]。这些方法有助于更好地保持涡量，并在捕捉精细尺度特征时展现出更高的稳定性。

光滑粒子流体动力学（SPH）方法被引入计算机图形学后，Müller 等人 [24], 的工作推动了大量研究，致力于增强不可压缩性和稳定性[4, 14, 15], 、提高空间自适应性[2] ，并减轻稀疏采样区域中因插值不一致而产生的误差[3, 39]。Koschier 等人 [19]对 SPH 技术进行了全面综述。基于位置的流体（PBF）方法[10, 22] 将 SPH 转化为约束形式，使其即便采用较大的时间步长也能保持出色的稳定性。涡方法通常采用拉格朗日表示，例如粒子[31], 、涡丝[37], 和涡片[27]。这类方法将流体方程改写为涡量-速度形式，但在几何处理和边界处理方面存在困难。

除了纯欧拉方法或拉格朗日方法之外，还存在结合这两种框架的混合方法[5, 13, 17, 34],, 以及整合涡量和速度形式的技术 [20, 36] 。传统流体模拟方法通常受其空间离散格式固有局限性的制约，多年来促使人们不断进行改进。我们的研究遵循采用连续表示这一理念，从而提供更强的表达能力。尽管由优化驱动的方法在某些方面可能无法匹敌依赖显式表示的当前最先进的技术，但它在保留精细细节和适应空间变化方面展现出更优越的能力。我们相信，这为未来开展流体模拟研究提供了一条有价值且充满前景的方向。

流体模拟的内存高效空间表示。另一个新兴方向是在模拟框架中融合隐式神经表示（INR）与高斯函数。许多研究将这些表示用作质量分布函数，并将其与物质点法等成熟的数值求解器相结合，从而实现基于物理的场景操控与动画 [12, 40]。例如，Deng 等人 [9] 采用 INR 存储流图，并通过具备扩展平流能力的基于网格的流体求解器，获得了无耗散的模拟结果。然而，将这些创新表示用作求解时变偏微分方程（偏微分方程）的空间函数这一应用方向，相关研究仍相对有限。与本研究最相关的工作包括偏微分方程的隐式神经空间表示（INSR） [6] 和神经蒙特卡罗流体（NMC） [16]。INSR 框架将 INR 用作流体和软体模拟的空间表示，并通过优化技术求解时变偏微分方程。相比之下，NMC 方法利用 INR 固有的连续性，通过采用球面漫步法进行压力计算，并改进隐式神经表示架构以有效适应边界条件。最近，Xing 等人 [41] 提出了一种基于高斯空间表示（GSR）的无网格流体求解器，将连续的流体速度场建模为多个高斯函数的加权和。该方法为传统的欧拉/拉格朗日方法提供了一种替代方案，在精度和细节保留方面表现出色，但无法保证严格满足无散条件。

3 背景

流体的动力学通常由其速度 $\boldsymbol{u}(\boldsymbol{x},\ t)$ 描述，这是定义在流体空间域 Ω 与所关注的时间区间 $[0, T]$的笛卡尔积上的向量场。对于不可压缩流， $\boldsymbol{u}$ 受带有无散约束的纳维-斯托克斯方程支配：

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{g}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

其中 ρ 为流体密度， p 为压力， ν 为运动黏度，且 $\boldsymbol{g}$ 为外力产生的加速度。如果流体为无黏（即 $\nu = 0$），假设 ρ 为常数，对式1 取旋度，可得涡量 $\boldsymbol{\omega} = \nabla\times\boldsymbol{u}$ 的控制方程

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\nabla)\boldsymbol{u} + \nabla\times\boldsymbol{g}. \tag{3}$$

给定流体的初始状态

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}^0(\boldsymbol{x}) \quad \forall \boldsymbol{x}\in\Omega \tag{4}$$

以及边界条件

$$\boldsymbol{u}(\boldsymbol{x},t)=\boldsymbol{u}_{\mathrm{b}}(\boldsymbol{x},t)\quad\forall\boldsymbol{x}\in\Gamma_1, \tag{5}$$

$$\boldsymbol{u}(\boldsymbol{x},t)\cdot\boldsymbol{n}_{\Gamma_2}(\boldsymbol{x})=f_{\mathrm{b}}(\boldsymbol{x},t)\quad\forall\boldsymbol{x}\in\Gamma_2, \tag{6}$$

其中 $\boldsymbol{u}^0$，$\boldsymbol{u}_{\mathrm{b}}$，$f_{\mathrm{b}}$ 是已知函数，Γ_1，$\Gamma_2\subseteq\partial\Omega$ 分别是应用无滑移和自由滑移边界条件的区域，$\boldsymbol{n}_{\Gamma_2}$ 是边界几何的法向量。数值求解器通常会将感兴趣的时间区间离散为多个时间步，并根据 $\boldsymbol{u}^0$ 以下公式 1或公式 3计算每个时间步中的流速演化，同时确保每个时间步都满足无散约束和边界条件。

4 无散核

我们的求解器利用无散核（DFK）对不可压缩流的速度场进行参数化，同时保持经典的时间离散方式不变。本节将介绍 DFK 的数学公式，以及求解器演化 DFK 所需的一些基本要素。

数学公式。给定空间维度 d，第 i 个 DFK 是一个具有多个属性的粒子：其位置为 $\boldsymbol{p}_i\in\mathbb{R}^d$、核半径为 $h_i\in\mathbb{R}$，权重为 $\boldsymbol{w}_i\in\mathbb{R}^d$。我们首先利用径向基函数 $\phi(r)$ 构造一个标量值核：

$$\phi_i(\boldsymbol{x})=\phi\left(\frac{\|\boldsymbol{x}-\boldsymbol{p}_i\|}{h_i}\right). \tag{7}$$

然后，可以从 ϕ_i进一步推导出矩阵值核 $\boldsymbol{\psi}_i$：$\mathbb{R}^d\to\mathbb{R}^{d\times d}$

$$\boldsymbol{\psi}_i(\boldsymbol{x})=(-\mathbf{I}\nabla\cdot\nabla+\nabla\nabla^\top)\phi_i(\boldsymbol{x}), \tag{8}$$

其中 $\mathbf{I}$ 是单位矩阵，$\nabla\nabla^\top$ 是海森算子。由无散核诱导的向量场定义为

$$\tilde{\boldsymbol{u}}_i(\boldsymbol{x})=\boldsymbol{\psi}_i(\boldsymbol{x})\boldsymbol{w}_i. \tag{9}$$

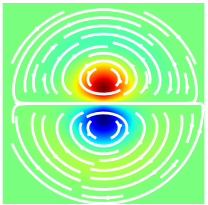
可以证明，对于任意给定的权重 $\boldsymbol{w}_i$，只要径向基函数 $\phi(r)$ 是 C^2 连续的 [26]，就有 $\nabla\cdot\tilde{\boldsymbol{u}}_i=0$。因此，结合 N 个无散核，我们可以得到无散向量场的表示

$$\tilde{\boldsymbol{u}}(\boldsymbol{x})=\sum_{i=1}^{N}\tilde{\boldsymbol{u}}_i(\boldsymbol{x})=\sum_{i=1}^{N}\boldsymbol{\psi}_i(\boldsymbol{x})\boldsymbol{w}_i \tag{10}$$

其可学习参数为 $\Theta=\{\boldsymbol{p}_i,h_i,\boldsymbol{w}_i:\ i=1,,N\}$。我们选择 [38] 中提出的 C^4连续分段多项式函数作为径向基函数：

$$\phi(r)=\begin{cases}(1-r)^6(35r^2+18r+3), & r\in[0,1]\\ 0, & \text{otherwise}\end{cases}, \tag{11}$$

因为已证明这是 [26] 中无散核的最优选择。插图展示了采用该公式时，二维中单个无散核产生的速度场流线图，背景颜色反映其涡量。将式 11 代入式 10 即可得到无散核的最终公式：



$$\tilde{\boldsymbol{u}}(\boldsymbol{x})=\sum_{i=1}^{N}f(r)\boldsymbol{w}_i+g(r)(\boldsymbol{w}_i\cdot\boldsymbol{y})\boldsymbol{y}, \tag{12}$$

where $\boldsymbol{y}=\frac{\boldsymbol{x}-\boldsymbol{p}_i}{h_i}$, $r=\|\boldsymbol{y}\|$,

$$\begin{aligned}f(r)&=56(1-r)^4[-5(d+5)r^2+(d-1)(4r+1)],\\ g(r)&=1680(1-r)^4.\end{aligned} \tag{13}$$

计算开销节省。通过式 12计算向量场需要 $O(N)$ 次浮点运算，对于需要频繁查询的任务而言，计算开销过大。为降低时间复杂度，我们利用每个无散核的非零域具有紧凑性的特点，采用通常用于基于粒子的流体模拟中以加速邻域搜索的哈希网格，将单次查询的浮点运算次数降至 $O(1)$。

空间微分方面的优势。无散核的梯度可通过对等式 $\boldsymbol{x}$ 两边关于其求导得到，见12：

$$\nabla\tilde{\boldsymbol{u}}(\boldsymbol{x})=\sum_{i=1}^{N}\frac{1}{h_i}\left(\frac{f'(r)}{r}\boldsymbol{w}_i\boldsymbol{y}^\top+\frac{g'(r)}{r}(\boldsymbol{w}_i\cdot\boldsymbol{y})\boldsymbol{y}\boldsymbol{y}^\top+g(r)\boldsymbol{y}\boldsymbol{w}_i^\top+g(r)(\boldsymbol{w}_i\cdot\boldsymbol{y})\mathbf{I}\right), \tag{14}$$

由此还可以进一步推导散度和旋度算子。与依赖自动微分来计算微分量的隐式神经表示不同，无散核支持直接且高效的求导，其时间复杂度与计算 DFK 场本身相同。此外，$\phi(r)$ 的 C^4连续性保证了 $\nabla\tilde{\boldsymbol{u}}$具有 C^1连续性，不会像 GSR 中的截断高斯函数那样因不连续而引入数值误差。这些固有优势使得在施加基于物理的约束时能够更快、更准确地进行优化，因此，无散核是用于动态模拟的计算效率更高的选择。

5 算法

本节将介绍我们的动态无散核（DDFK）求解器，该求解器使用无散核作为表示来模拟无外力的无黏不可压缩流。我们的方法主要由三个部分组成：初始化、基于物理的优化和重新初始化。算法 1 简要概述了我们的方法。

算法 1DDFK 流体求解器

$\tilde{\boldsymbol{u}}^0\leftarrow$ 初始化$(\boldsymbol{u}^0)$
对于 $n\leftarrow 1,\cdots\ T$**执行**
  //每隔 N_{reinit} 步重新初始化无散核：
  如果 n 模 $N_{\text{重新初始化}}=0$ **然后**
    $\tilde{\boldsymbol{u}}^\star\leftarrow$ 重新初始化$(\tilde{\boldsymbol{u}}^{n-1})$
  else
    $\tilde{\boldsymbol{u}}^\star\leftarrow\tilde{\boldsymbol{u}}^{n-1}$
  结束 if
  //基于物理的优化的初始猜测：
    $\tilde{\boldsymbol{u}}^*\leftarrow$ 平流输运位置$(\tilde{\boldsymbol{u}}^\star,\ \tilde{\boldsymbol{u}}^{n-1})$
  //基于物理的优化：
  $\tilde{\boldsymbol{u}}^n\leftarrow$ 优化损失$(\tilde{\boldsymbol{u}}^*,\tilde{\boldsymbol{u}}^{n-1})$
结束循环

5.1 初始化

我们首先通过初始化 DDFK $\tilde{\boldsymbol{u}}^0$，使其拟合给定的初始流速条件 $\boldsymbol{u}^0$。与 [26], 类似，我们将这一过程表述为一个优化问题：

$$\arg\min_{\Theta} \frac{1}{d|\Omega|}\int_{\Omega} \|\tilde{\boldsymbol{u}}^0(\boldsymbol{x}) - \boldsymbol{u}^0(\boldsymbol{x})\|_1 \mathrm{d}V. \tag{15}$$

然而，直接计算该积分十分困难，因此我们用其蒙特卡洛估计替代目标函数：

$$\mathcal{L}_{\text{val}} = \frac{1}{Qd}\sum_{j=1}^{Q} \|\tilde{\boldsymbol{u}}^0(\boldsymbol{x}_j) - \boldsymbol{u}^0(\boldsymbol{x}_j)\|_1, \tag{16}$$

其中，$\boldsymbol{x}_1$、$\cdots \boldsymbol{x}_Q$ 在每次优化迭代中从 Ω 均匀随机采样。为了进一步提高拟合质量，以利于后续的时间积分过程，我们还采用了额外的梯度损失：

$$\mathcal{L}_{\text{grad}} = \frac{1}{Qd^2}\sum_{j=1}^{Q} \|\nabla\tilde{\boldsymbol{u}}^0(\boldsymbol{x}_j) - \nabla\boldsymbol{u}^0(\boldsymbol{x}_j)\|_{\text{sum}}, \tag{17}$$

其中，$\|\cdot\|_{\text{sum}} = \sum_k \sum_l |[\cdot]_{kl}|$ 表示矩阵所有元素绝对值之和。除了场本身的取值外，梯度损失还对无散核在空间中的变化提供监督，使其在收敛时能更准确地拟合给定场。初始化过程中的总损失为

$$\mathcal{L}_{\text{init}} = \mathcal{L}_{\text{val}} + \mathcal{L}_{\text{grad}}. \tag{18}$$

在实践中，我们发现，在某些示例中，靠近 Ω 外边界处的拟合误差可能极高。这是因为优化过程中采样点只能落在外边界的一侧，而无散核无法直观地外推场（不像 GSR，见第 7 节中的零净动量段落 [26]）。我们通过将流体域稍微扩展至 Ω_ε，并构造一个定义在$\hat{\boldsymbol{u}}^0$ 上的 Ω_ε 的解析延拓 $\boldsymbol{u}^0$，然后将 Ω 改为 Ω_ε，并将 $\boldsymbol{u}^0$ 改为$\hat{\boldsymbol{u}}^0$，即可修正此问题，见式16和式17。

优化前，我们将无散核的位置初始化为覆盖 Ω_ε 的网格顶点坐标。无散核的半径设为

$$h_i = \eta\left[\frac{\Gamma(1+d/2)|\Omega_\varepsilon|}{N\pi^{d/2}}\right]^{1/d} \tag{19}$$

根据 [26],，其中 $\Gamma(\cdot)$ 是伽马函数，η 是一个超参数，而权重设为零。随后，我们使用带有调度器的 Adam 优化器；在连续若干次迭代中损失未下降后，调度器会降低学习率，从而获得可用于模拟的初始无散核。

5.2 基于物理的优化

时间积分通过优化无散核的可学习参数来完成，以最小化一系列基于物理的损失。通过将搜索空间限制为无散核所表示的无散向量场，我们的求解器无需使用其他基于内存高效空间表示（如隐式神经空间表示和 GSR）的方法中常见的物理信息散度损失，从而实现更直接的优化，同时严格满足不可压缩条件。

5.2.1 基于平流的初始猜测。为确保获得更好的收敛结果和时间一致性，我们在优化前将粒子位置向前平流一个时间步，同时保持无散核的其他参数不变。具体而言，我们采用速度场 $\tilde{\boldsymbol{u}}^{n-1}$ 和四阶龙格–库塔法来移动粒子位置。请注意，这一步不同于传统粒子方法（如光滑粒子流体动力学或粒子在单元格（PIC）与流体隐式粒子（FLIP））中的典型平流操作：在那些方法中，粒子沿自身速度移动；而移动无散核所用的速度则由邻近粒子计算得出。

尽管此步骤只能提供一个非常粗略的平流方程近似解 $\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \boldsymbol{0}$，但它是后续优化的一种可行初始猜测。利用当前速度场更新粒子位置，有助于后续优化搜索更接近真实平流状态的局部最优解，从而使模拟得到的整体流速更符合物理现实。第 6.4 节通过消融测试验证了这一平流步骤的有效性。

5.2.2 涡量损失。由于无散核无法表示散度非零的向量场，因此，某些传统求解器中常用的、将方程1 拆分为平流步骤和投影步骤的算子分裂方法不再适用于我们的情况。因此，我们借鉴涡方法的思路，改为通过优化无散核来最小化以下涡量损失，从而求解方程3：

$$\mathcal{L}_{\text{vor}} = \frac{1}{Q\hat{d}}\sum_{j=1}^{Q} \|\nabla\times\tilde{\boldsymbol{u}}^n(\boldsymbol{x}_j) - \omega(\boldsymbol{x}_j)\|_1, \tag{20}$$

^^ 其中，在二维中 $d=1$，在三维中 $d=3$，因为涡量在二维中是标量，在三维中是向量；每次迭代时，$\boldsymbol{x}_1, \cdots \boldsymbol{x}_Q$ 均从 Ω 中均匀随机采样；$\omega(\boldsymbol{x})$ 是从 $\nabla\times\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x})$ 平流而来的涡量场。记 $\boldsymbol{\Phi}^{n-1}$：$\mathbb{R}^d \to \mathbb{R}^d$ 为一个映射，它将时间步 $n-1$ 时刻的点映射到经过速度场 $\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x})$ 的四阶龙格–库塔时间积分向前输运后的位置；$\boldsymbol{\Psi}^{n-1}$：$\mathbb{R}^d \to \mathbb{R}^d$ 则为 $\boldsymbol{\Phi}^{n-1}$的逆映射。在二维中，$(\boldsymbol{\omega}\cdot\nabla)\boldsymbol{u} = \boldsymbol{0}$，因此方程3 可写为$\frac{\mathrm{D}\omega}{\mathrm{D}t} = 0$，这意味着涡量仅沿速度场输运，不随时间演变：

$$\omega(\boldsymbol{x}) = \nabla\times\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{\Psi}^{n-1}(\boldsymbol{x})). \tag{21}$$

在三维中，涡量场的演化遵循 $\frac{\mathrm{D}\omega}{\mathrm{D}t} = (\nabla\boldsymbol{u})\omega$，其中右端是一个矩阵乘积。因此，如 [35], 中所述，涡量是一个线元，可以在流映射$\boldsymbol{\Phi}^{n-1}$ 下以

$$\omega(\boldsymbol{x}) = [\mathrm{d}\boldsymbol{\Psi}^{n-1}(\boldsymbol{x})]^{-1}\nabla\times\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{\Psi}^{n-1}(\boldsymbol{x})), \tag{22}$$

其中 $\mathrm{d}\boldsymbol{\Psi}^{n-1}$ 是 $\boldsymbol{\Psi}^{n-1}$的雅可比矩阵。

5.2.3 边界处理。我们通过向优化系统引入边界损失来处理边界条件。无滑移边界由式 5 描述，并通过类型 1 边界损失加以约束

$$\mathcal{L}_{\text{b1}} = \frac{1}{Q_{\text{b1}}d}\sum_{j=1}^{Q_{\text{b1}}} \|\tilde{\boldsymbol{u}}^n(\boldsymbol{z}_j) - \boldsymbol{u}_{\text{b}}(\boldsymbol{z}_j, t_n)\|_1, \tag{23}$$

而由式 6 描述的自由滑移边界则通过类型 2 边界损失加以约束

$$\mathcal{L}_{\mathrm{b2}} = \frac{1}{Q_{\mathrm{b2}}} \sum_{j=Q_{\mathrm{b1}}+1}^{Q_{\mathrm{b1}}+Q_{\mathrm{b2}}} |\tilde{\boldsymbol{u}}^n(\boldsymbol{z}_j) \cdot \boldsymbol{n}_{\Gamma_2}(\boldsymbol{z}_j) - f_{\mathrm{b}}(\boldsymbol{z}_j, t_n)|, \tag{24}$$

其中， $\boldsymbol{z}_1$、$\cdots$ $\boldsymbol{z}_{Q_{\mathrm{b1}}}$ 是在适用无滑移条件的边界几何 Γ_1 上均匀随机采样得到的，$\boldsymbol{z}_{Q_{\mathrm{b1}}+1}$、$\cdots$ $\boldsymbol{z}_{Q_{\mathrm{b1}}+Q_{\mathrm{b2}}}$是在适用自由滑移条件的 Γ_2 上采样得到的， t_n 表示第 n步的时间。

时间积分的总损失是涡量损失和边界损失的线性组合：

$$\mathcal{L}_{\mathrm{adv}} = \mathcal{L}_{\mathrm{vor}} + \lambda_{\mathrm{b1}}\mathcal{L}_{\mathrm{b1}} + \lambda_{\mathrm{b2}}\mathcal{L}_{\mathrm{b2}}. \tag{25}$$

粒子平流后，我们使用与初始化步骤相同的优化器来最小化 $\mathrm{L}_{\mathrm{adv}}$ ，从而得到下一步$\tilde{\boldsymbol{u}}^n$的无散核。

5.3 重新初始化

随着模拟经过多个时间步，优化过程可能会驱使 DFK 粒子聚集在某些局部区域，进而形成粒子分布稀疏的空洞。为缓解这一问题，我们每隔 N_{reinit} 步在时间步开始时重新初始化无散核，这包括从头开始重新学习当前速度场，并在特定区域引入额外粒子以保留流动细节。

在重新学习阶段，我们按照第 5.1 节所述，进行相同的拟合优化，并将目标速度场替换为最后一个时间步的无散核 $\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x})$。由于速度场通常会随时间变得越来越复杂，简单的重新学习过程可能不足以解析 $\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x})$ 中的精细尺度结构。因此，我们从前一状态 $\tilde{\boldsymbol{u}}^{n-1}$ 借用一些粒子，以补充拟合不足的区域。具体而言，我们在每个前序无散核的位置处计算梯度误差：

$$\boldsymbol{G}_i^{\mathrm{err}} = \nabla\tilde{\boldsymbol{u}}^{\star}(\boldsymbol{p}_i^{n-1}) - \nabla\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{p}_i^{n-1}), \tag{26}$$

其中， $\boldsymbol{p}_i^k$表示无散核状态$\tilde{\boldsymbol{u}}^k$中第 i个粒子的位置。接下来，我们将满足 $\|G_i^{\mathrm{err}}\|_{\mathrm{sum}} > \alpha \cdot \mathrm{mean}\{\|G_j^{\mathrm{err}}\|_{\mathrm{sum}}\colon\ j = 1, \cdots\ N\}$ 的粒子追加到 $\tilde{\boldsymbol{u}}^{\star}$。最后，我们再次进行优化，使 $\tilde{\boldsymbol{u}}^{\star}$ 拟合 $\tilde{\boldsymbol{u}}^{n-1}(\boldsymbol{x})$。本文所有示例中均选用 $\alpha = 2$。

6 结果

我们在具有不同初始条件、多种边界形状和类型的各类示例上评估了我们的方法。与大多数依赖一阶优化的求解器相比，我们的方法展现出与 GSR 相当的学习率鲁棒性；在不同场景下，两者都无需进行超参数调优。为适应不同尺度的测试案例，我们将整个流体域和边界几何归一化到标准尺寸，并相应调整初始速度场。通过这一缩放策略，一组参数配置便足以应对所有二维测试案例，所有三维测试案例则使用另一组配置。

我们首先在一个具有解析解的示例上应用该方法，以量化其数值精度和收敛率。随后，我们通过与传统求解器及其他内存高效空间表示进行对比实验，验证该方法的有效性；结果凸显了其数值耗散较低、满足不可压缩条件的能力更强。此外，我们还展示了无散核能够处理复杂动力学，胜任高度复杂的流体模拟。最后，我们进行了一系列消融研究，以评估该方法中各个关键组件的作用。本节末尾给出了所有示例的性能表现。

6.1 定量研究

我们使用泰勒-格林涡流实验进行定量研究，分析我们的方法在初始化和时间积分期间的数值精度及优化的收敛速率。在此示例中，初始速度场设为

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{y}) = \begin{bmatrix} \sin x \cos y \\ -\cos x \sin y \end{bmatrix} \tag{27}$$

定义在流体域 $\mathrm{D} = [0, 2\pi] \times [0, 2\pi]$ 上。随后，我们以 0.001 秒的时间步长运行模拟，共 100 帧。

由于该速度场在无黏不可压缩流体中将保持恒定，因此我们可以通过计算某一帧的模拟速度场与初始速度场 $\boldsymbol{u}(\boldsymbol{x},\ \boldsymbol{y})$ 在 60×60 均匀网格上的均方误差（MSE），来衡量数值误差。

表 1 中的结果表明，我们的方法的数值误差远低于基于 INR 的方法和 GSR。

方法	第0帧	第50帧	第100帧
欧拉方法	2.432×10^{-4}	9.757×10^{-3}	2.019×10^{-2}
INSR	8.998×10^{-7}	1.715×10^{-5}	1.992×10^{-5}
NMC	1.829×10^{-4}	6.492×10^{-4}	1.725×10^{-3}
GSR	9.957×10^{-8}	2.510×10^{-7}	2.181×10^{-7}
Ours	2.343×10^{-8}	2.484×10^{-8}	3.305×10^{-8}

表1. 使用泰勒-格林涡旋示例进行的定量比较。不同方法模拟得到的速度场与第0、50和100帧真实值之间的均方误差。

6.2 验证

DDFK 的有效性。我们在泰勒涡旋示例上，将本方法与传统欧拉求解器、INSR 和 GSR 进行比较，并以涡量粒子法的模拟结果作为真实值，从而验证 DDFK 的有效性。完整结果见图 6。本方法在保持涡旋和细节方面的能力与 GSR 相当，且背景噪声显著低于 INSR，表明本方法具有更好的稳定性。如图所示，本方法在涡旋保持和细节呈现方面的能力与 GSR 同样出色；两种方法都能准确捕捉泰勒涡旋的内在结构和演化动力学，且不会出现明显退化。尽管本方法在细节保持方面与 INSR 相当，但其背景

噪声更少，同时能更精确地保持涡旋强度，这体现了我们的 DDFK 求解器具有更优的稳定性。此外，本方法以少得多的自由度捕捉到与欧拉求解器相同的流动细节，同时实现了更好的涡旋保持效果，充分展现了其强大的空间自适应性。

零散度保持验证。通过两个实验验证 DDFK 在整个模拟过程中保持速度场零散度的能力。在第一个示例中，活塞在封闭容器内以恒定速度向下移动，如图 1 所示；活塞上施加无滑移条件，容器壁上施加自由滑移条件。容器是一个正方形，中心位于 (0, 0)，边长为 10。活塞是一个宽 8、高 0.5 的矩形，初始中心位于 (0, 2)，并以 0.5 的速度向下移动。将活塞上边缘所在直线位于容器内部的线段记为集合 $\mathrm{S}^{\mathrm{upper}}$，将活塞的上边缘记为 $\mathrm{E}^{\mathrm{upper}}$；定义曲线 C 上的流量为速度场的 y 分量沿 C的积分，记作 F_C。如果流体不可压缩，

$$F_{\mathcal{E}^{\mathrm{upper}}} + F_{\mathcal{S}^{\mathrm{upper}}\setminus\mathcal{E}^{\mathrm{upper}}} = F_{\mathcal{S}^{\mathrm{upper}}} = 0 \tag{28}$$

在任意给定时刻都成立。请注意，$\mathcal{S}^{\mathrm{upper}}\setminus\mathcal{E}^{\mathrm{upper}}$ 在图 1 中以橙色线条标出。根据活塞的速度和宽度，我们可以计算出 $F_{\mathcal{E}^{\mathrm{upper}}} = -0.5 \times 8 = -4$，因此 $F_{\mathcal{S}^{\mathrm{upper}}\setminus\mathcal{E}^{\mathrm{upper}}}$ 的真实值为 4。我们通过计算均匀采样的 $F_{\mathcal{S}^{\mathrm{upper}}\setminus\mathcal{E}^{\mathrm{upper}}}$ 个点处速度的 y 分量的平均值，来估计使用无散核和 GSR 进行模拟时的 400 个点在 $\mathcal{S}^{\mathrm{upper}}\setminus\mathcal{E}^{\mathrm{upper}}$ 上的流量乘以长度 $|\mathcal{S}^{\mathrm{upper}}\setminus\mathcal{E}^{\mathrm{upper}}|$。如图 1 所示，在整个模拟过程中，我们的 DDFK 求解器始终使流量接近真实值，而 GSR 的流量则低得多。这也使得采用我们的方法时，活塞两侧的涡旋上升得更高。此外，从图 1 中的流线和图 中的散度场可以看出，GSR 在基于物理信息的散度损失进行优化时未能维持无散场，而我们的方法通过无散核完美地解决了这一问题。

在第二个示例中，我们比较了两种方法在如下情形中的表现：一对涡旋穿过两个带无滑移边界的球形障碍物之间的间隙，如图 5 所示。在这一过程中，正确模拟调和分量是成功通过间隙的关键。我们测试了两种障碍物间距不同的情形。两种方法都能成功让涡旋穿过障碍物，但 GSR 中的涡旋穿过狭窄间隙的速度比穿过宽阔间隙更快，而且涡旋尺寸也出现了不正确的增大。这是因为障碍物使 GSR 的速度场难以维持无散条件，而我们的方法能够自然地保持无散特性，因此能正确捕捉障碍物对涡旋前进的阻碍作用——穿过较窄间隙时，涡旋会停留更久。

低数值耗散。我们通过跳蛙实验验证了方法的数值精度：两个从底部释放的涡对进行类似跳蛙的旋转并向上移动。如图所示，2与在第四次半周旋转前涡对便合并的高斯流体相比，我们的方法在第五次半周旋转后仍能保持涡的间距和结构。这表明我们的方法相比高斯流体具有更低的数值耗散。

6.3 更多示例

卡门涡街。 在此示例中，流体从计算域左侧流入，绕过一个圆柱形固体障碍物，并在其后方形成交替排列的涡街。如图 3 所示，计算域的上、下边界采用自由滑移条件，而圆柱形障碍物上采用无滑移条件。同时，计算域的左、右边界采用自由滑移条件，且 f_b 设置为流入速度，以保持流体持续流动。在初始化和重新初始化步骤中，我们在计算域左边界外放置了额外的粒子，以防止平流导致粒子不足。图 3 中的结果表明，我们的方法和高斯流体都比基于 INSR 的神经蒙特卡洛方法具有更好的稳定性；而我们的模拟结果则展现出更高的雷诺数，且障碍物后方形成了不规则涡街。

6.4 消融测试

基于平流的初始猜测。 初始猜测提高了我们方法的准确性。如图 7所示，我们在二维跳蛙示例中比较了使用和不使用初始猜测的方法。经过长时间平流后，不使用初始猜测的方法生成的结果变得不对称，而完整方法的结果仍保持对称，并能更长时间地维持涡旋对的旋转。从数值结果来看，我们还可以观察到，尽管使用基于平流的初始猜测之前涡量场的损失较低，但优化后，使用初始猜测的涡量场损失更低。这表明初始猜测使我们的方法能够执行更准确的平流。

重新初始化。 图 4 比较了不进行重新初始化的方法与完整方法。前者在涡旋尾部附近出现了噪声伪影，如图 4a中的红框所示，这是由于粒子不足，如图 4c所示。重新初始化过程有效地向这些拟合不足的区域补充粒子，从而得到更清晰的模拟。

7 结论

本文提出了一种基于动态无散核的不可压缩流无网格求解器。我们的方法能够完美保持零散度，解决了使用 INSRs 和 GSR 等其他内存高效的空间表示优化物理信息散度损失时面临的主要问题。同时，DDFK 还具备 INSRs 和 GSR 相较于传统离散化方法的优势：空间自适应性、连续可微性和涡量保真度。此外，我们的方法能够在长时间范围内稳定模拟流体现象，并能泛化到具有复杂涡旋和多样边界的场景。

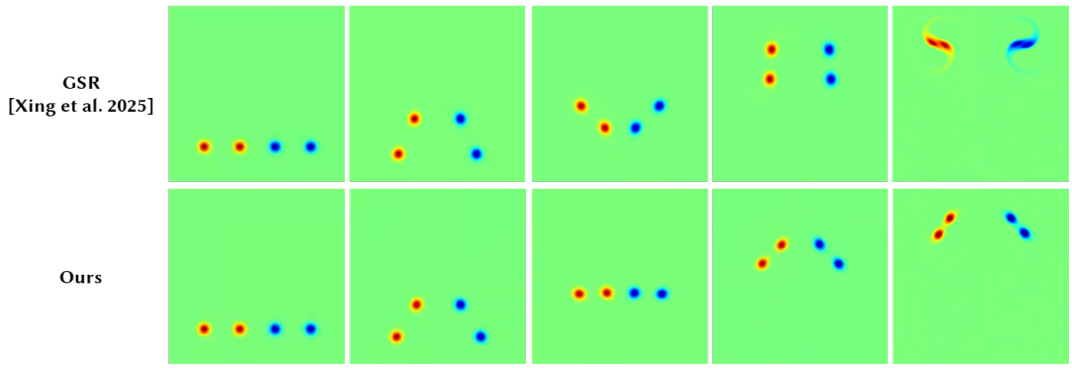


图 2 [41] 2 0, 165, 456, 。我们的方法与高斯流体（GSR）的比较 。
1050 1500.
从左到右的图像分别是 Leapfrog D 在帧 和 的模拟结果。

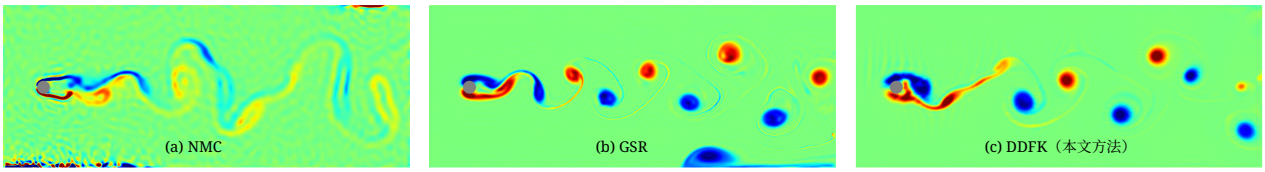


图 3。神经蒙特卡洛（NMC）、高斯流体（GSR）和我们的方法（DDFK）生成的卡门涡街示例。子图分别展示了模拟结果中第 152、199 和 199 帧的涡量场。

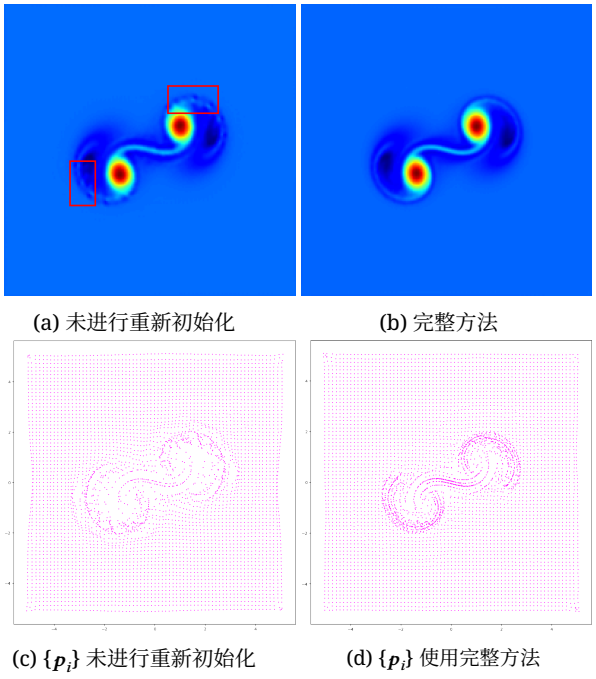


图 4. 重新初始化的消融研究。我们展示了帧 180 时的模拟结果和粒子位置。

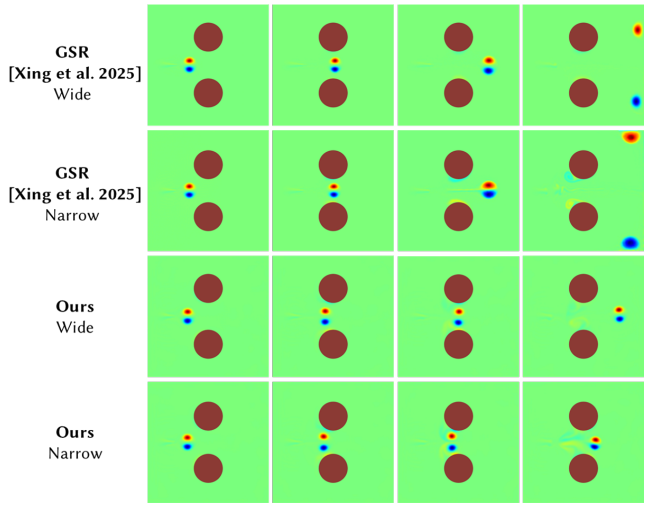


图 5. 我们使用宽间隙和窄间隙的涡旋穿越示例，将我们的方法与高斯流体（GSR） [41] 进行了比较。宽间隙设置下，宽间隙和窄间隙设置中两个球形障碍物的位置分别为 (0.5,0.27), (0.5, 0.73) 和 (0.5,0.285), (0.5, 0.715)。所有障碍物的半径均为 0.1174。图中从左到右依次展示帧 200、336、520 和 879。

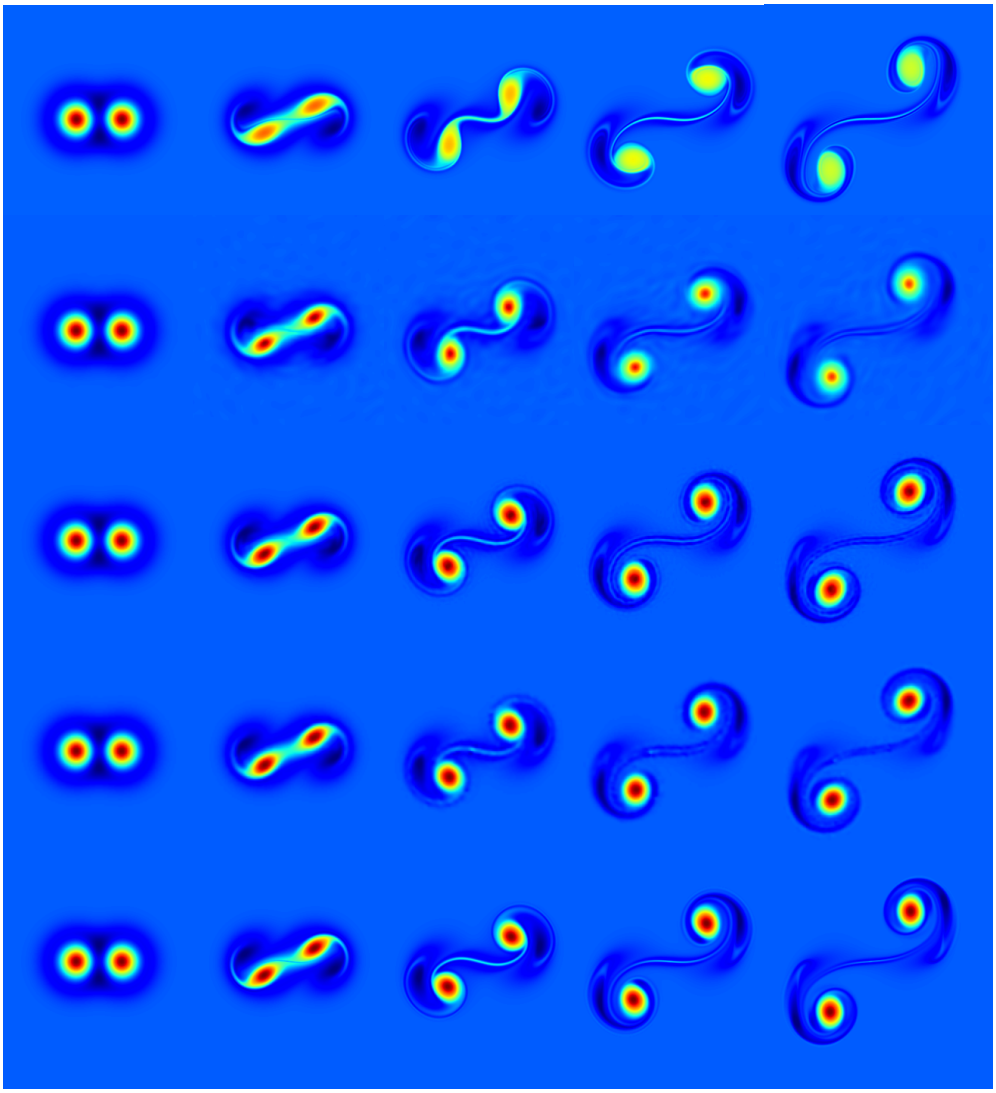


图 6. 在泰勒涡旋示例中，我们将本方法与传统欧拉网格以及不同的内存高效空间表示 INSR 和 GSR 进行比较。第一列括号内为相应速度场表示的平均文件大小。图像分别展示了第 0、100、200、300 和 399 帧的涡量场。

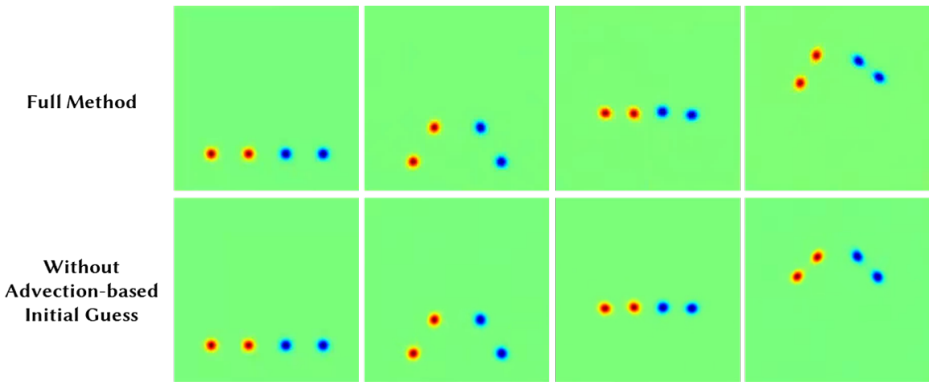


图 7. 基于平流的初始猜测消融测试。从左到右依次为 Leapfrog2D 在第 0、150、533 和 1000 帧的模拟结果。

参考文献

[1] Mridul Aanjaneya、Ming Gao、Haixiang Liu、Christopher Batty 和 Eftychios Sifakis。2017年。用于高分辨率自适应液体的幂图与稀疏分页网格。《ACM 图形学汇刊》(TOG) 36, 4 (2017年), 1–12。[2] Bart Adams、Mark Pauly、Richard Keiser 和 Leonidas J. Guibas。2007年。自适应采样粒子流体。载于《ACM SIGGRAPH 2007 论文集》（美国加利福尼亚州圣迭戈）（SIGGRAPH '07）。计算机协会，美国纽约州纽约，48–es。doi:10.1145/1275808.1276437[3] Stefan Band、Christoph Gissler、Andreas Peer 和 Matthias Teschner。2018年。用于无散黏性光滑粒子流体的移动最小二乘压力边界。《计算机与图形学》76 (2018年), 37–46。[4] J an Bender 和 Dan Koschier。2015年。无散光滑粒子流体动力学。载于第14届 ACM SIGGRAPH/Eurographics 计算机动画研讨会论文集。147–155。[5] Jeremiah U Brackbill 和 Hans M Ruppel。1986年。流体流动的二维自适应分区粒子网格计算方法：FLIP。《计算物理学杂志》65, 2 (1986年), 314–343。[6] Honglin Chen、Rundi Wu、Eitan Grinspun、Changxi Zheng 和 Peter Yichen Chen。2023年。时变偏微分方程的隐式神经空间表示。载于第40届国际机器学习大会论文集（美国夏威夷州檀香山）（ICML'23）。JMLR.org，文章202，共16页。[7] Honglin Chen、Rundi Wu、Eitan Grinspun、Changxi Zheng 和 Peter Yichen Chen。2023年。时变偏微分方程的隐式神经空间表示。载于国际机器学习大会。PMLR，5162–5177。[8] Alexandre Joel Chorin。1968年。纳维-斯托克斯方程的数值求解。《计算数学》22, 104 (1968年), 745–762。[9] Yitong Deng、Hong-Xing Yu、Diyang Zhang、Jiajun Wu 和 Bo Zhu。2023年。神经流映射上的流体模拟。《ACM 图形学汇刊》42, 6，文章248（12月 2023 21 10 1145 3618392），页码。doi: 。/ Elie Diaz、Jerry Hsu、[10] Eisen Montalvo-Ruiz、Chris Giles 和 Cem Yuksel。2025年。隐式基于位置的流体。载于《亚洲 SIGGRAPH 大会 2025 论文集》。1–9。[11] Fan Feng、Jinyuan Liu、Shiying Xiong、Shuqi Yang、Yaorui Zhang 和 Bo Zhu。2022年。冲量流体模拟。《IEEE 可视化与计算机图形学汇刊》29, 6 (2022年), 3081–3092。[12] Yutao Feng、Yintong Shang、Xuan Li、Tianjia Shao、Chenfanfu Jiang 和 Yin Yang。2024年。PIE-NeRF：基于神经辐射场的交互式物理弹性动力学。arXiv: 2311 13099 [ ] 2311 13099。cs.CV [13] https://arxiv.org/abs/。Francis H Harlow。1964。粒子网格流体动力学计算方法。《计算物理方法》3 (1964年), 319–343。[14] Xiaowei He、Shusen Liu、Yuzhong Guo、Jian Shi 和 Ying Qiao。2025年。一种改进收敛性的可压缩与不可压缩流半隐式光滑粒子流体动力学方法。载于《计算机图形学论坛》。Wiley Online Library，e70043。[15] Markus Ihmsen、Jens Cornelis、Barbara Solenthaler、Christopher Horvath 和 Matthias Teschner。2013年。隐式不可压缩光滑粒子流体动力学。《IEEE 可视化与计算机图形学汇刊》20, 3 (2013年), 426–435。[16] Pranav Jain、Ziyin Qu、Peter Yichen Chen 和 Oded Stein。2024年。神经蒙特卡洛流体模拟。载于《ACM SIGGRAPH 2024 会议论文集》（美国科罗拉多州丹佛）（SIGGRAPH '24）。计算机协会，美国纽约州纽约，文章 9, 11 10 1145 3641519 3657438 页。doi: 。/ 。Chenfanfu Jiang、[17] Craig Schroeder、Andrew Selle、Joseph Teran 和 Alexey Stomak-hin。2015年。仿射粒子网格法。《ACM 图形学汇刊》(TOG) 34, 4 (2015年), 1–10。[18] ByungMoon Kim、Yingjie Liu、Ignacio Llamas 和 Jarek Rossignac。2005年。Flow-Fixer：在流体模拟中使用双向误差补偿与校正。载于 NPH。51–56。[19] Dan Koschier、Jan Bender、Barbara Solenthaler 和 Matthias Teschner。2022年。计算机图形学中的光滑粒子流体动力学方法综述。载于《计算机图形学论坛》，第41卷。Wiley Online Library，737–760。[20] Petros D Koumoutsakos、Georges-Henri Cottet 和 Diego Rossinelli。2008年。基于粒子的流动模拟——连接计算机图形学与计算流体力学。载于 SIGGRAPH 2008——第35届计算机图形学与交互技术国际会议。ACM，1–73。[21] Chung Ho Liu。2001年。用速度–涡量法数值求解三维纳维–斯托克斯方程。《流体数值方法国际期刊》35, 5 (2001年), 533–557。[22] Miles Macklin 和 Matthias Müller。2013年。基于位置的流体。《ACM 图形学汇刊》(TOG) 32, 4 (2013年), 1–12。[23] Aleka McAdams、Eftychios Sifakis 和 Joseph Teran。2010年。用于大网格流体模拟的并行多重网格泊松求解器。载于《计算机动画研讨会》，第65卷。74。[24] Matthias Müller、David Charypar 和 Markus Gross。2003年。面向交互式应用的基于粒子的流体模拟。载于2003年 ACM SIGGRAPH/Eurographics 计算机动画研讨会论文集。154–159。[25] Mohammad Sina Nabizadeh、Stephanie Wang、Ravi Ramamoorthi 和 Albert Chern。2022年。余向量流体。《ACM 图形学汇刊》(TOG) 41, 4 (2022年), 1–16。

[26] Xingyu Ni、Jingrui Xing、Xingqiao Li、Bin Wang 和 Baoquan Chen。2025年。使用无散核表示流场以进行重建。《ACM 计算机图形学与交互技术论文集》8，4，文章52（2025年8月），21页。doi:10.1145/3747872[27] Tobias Pfaff、Nils Thuerey 和 Markus Gross。2012年。用于流体动画的拉格朗日涡片。《ACM 图形学汇刊》（TOG）31，4（2012），1–8。[28] André Robert。1981年。原始气象方程的稳定数值积分方案。《大气—海洋》19，1（1981），35–46。[29] John Stanley Sawyer。1963年。一种求解涡量平流方程的半拉格朗日方法。《Tellus》15，4（1963），336–342。[30] Andrew Selle、Ronald Fedkiw、Byungmoon Kim、Yingjie Liu 和 Jarek Rossignac。2008年。一种无条件稳定的 MacCormack方法。《科学计算杂志》35，2（2008），350–371。[31] Andrew Selle、Nick Rasmussen 和 Ronald Fedkiw。2005年。一种用于烟雾、水和爆炸的涡粒子方法。载于《ACM SIGGRAPH 2005论文集》（美国加利福尼亚州洛杉矶）（SIGGRAPH '05）。计算机协会，美国纽约州纽约， 910 914 10 1145 1186822 1073282 – 。doi: . / 。Rajsekhar Setaluri 、[32] Mridul Aanjaneya 、 Sean Bauer 和 Eftychios Sifakis。2014 年。SPGrid：一种用于自适应烟雾模拟的稀疏分页网格结构。《ACM 图形学汇刊》（TOG）33，6（2014），1–12。[33] Jos Stam。1999年。Stable Fluids。载于《第26届年度计算机图形学与交互技术会议论文集》（SIGGRAPH '99）。ACM Press/Addison-Wesley Publishing Co.，美国， 121 128 10 1145 311535 311548 – 。doi: . / 。[34] Yizao Tang、Yuechen Zhu、Xingyu Ni 和 Baoquan Chen。 。用于模拟砂水混合物的颗粒单元方法。《ACM 图形学汇刊》44，6，文章 266 2025 19 10 1145 3763279 （12月），页码。doi: . / Sinan [35] Wang、Yitong Deng、Molin Deng、Hong-Xing Yu、Junwei Zhou、Duowen Chen、Taku Komura、Jiajun Wu 和 Bo Zhu。2024年。基于流图的欧拉涡方法。《ACM 图形学汇刊》（TOG）43，6（2024），1–14。[36] Sinan Wang、Junwei Zhou、Fan Feng、Zhiqi Li、Yuchen Sun、Duowen Chen、Greg Turk 和 Bo Zhu。2025年。基于涡粒子流图的流体模拟。《ACM 图形学汇刊》 44, 4, 91 2025 24 10 1145 3731198 文章（7月），页码。[37] doi: . / Steffen Weißmann 和 Ulrich Pinkall。 2010。基于丝状体的烟雾及其涡脱落与变分重联。载于《ACM SIGGRAPH 2010论文集》（美国加利福尼亚州洛杉矶）（SIGGRAPH '10）。计算机协会，美国纽约州纽约，文章 115, 12 10 1145 1833349 1778852 页码。doi: . / 。Holger [38] Wendland。1995。分片多项式、正定且紧支的最小次数径向函数。《计算与数学进展》4，1（1995），389–396。[39] Lukas Westhofen、Stefan Jeske 和 Jan Bender。2023年。用于一阶SPH导数的线性一致修正方法比较。《ACM计算机图形学与交互技术论文集》6，3（2023），1–20。[40] Tianyi Xie、Zeshun Zong、Yuxing Qiu、Xuan Li、Yutao Feng、Yin Yang 和 Chenfanfu Jiang。2024年。PhysGaussian：用于生成动力学的物理集成三维高斯。arXiv: 2311 12198 [ ] 2311 12198 . cs.GR [41] https://arxiv.org/abs/。Jingrui Xing、Bin Wang、Mengyu Chu 和 Baoquan Chen。2025。。高斯流体：一种基于高斯空间表示的无网格流体求解器。载于《ACM SIGGRAPH 2025会议论文集》（加拿大不列颠哥伦比亚省温哥华）（SIGGRAPH '25）。计算机协会，美国纽约州纽约，文章31，11页。doi:10.1145/3721238.3730620[42] Jonas Zehnder、Rahul Narain 和 Bernhard Thomaszewski。2018年。一种用于细节保持流体模拟的平流—反射求解器。《ACM 图形学汇刊》（TOG）37，4（2018），1–8。