arXiv:2301.06816v1 [cs.GR] 17 Jan 2023

一种用于液体模拟的有限元法与有限体积法相结合的方法

Tatsuya Koike早稻田大学
wassan2356@moegi.waseda.jp

Shigeo Morishima早稻田大学
shigeo@waseda.jp

Ryoichi Ando无所属
ryich.ando@gmail.com

摘要

我们提出了一种用于液体动画的新型欧拉模拟框架，同时利用有限元法和有限体积法。与以往将整个模拟域完全采用有限体积法或有限元法进行离散化的方法不同，我们的方法在空间上将两者结合起来，使两种离散化方式在交界处紧密耦合，同时在自由表面施加二阶精度边界条件。我们通过一种变分形式实现这一方法，并为此专门设计了新的形函数。将两种方法结合使用，我们可以兼取各自的优势。例如，有限体积法（FVM）能够产生稀疏线性系统；然而，使用四面体或沃罗诺伊单元等非结构化网格时，计算会变得复杂。另一方面，有限元法（FEM）产生的线性系统相对更稠密，但即使选用非结构化单元，复杂度也保持不变，从而便于实现空间自适应。本文提出在大部分区域使用有限体积法，以保留线性系统的稀疏性；在允许网格单元自由变形的区域则使用有限元法。局部移动网格就是这种方法的一个应用实例。我们表明，使局部网格移动与底层近似刚体运动相适应，可显著减少数值扩散，从而更好地保留锐边、薄片和液体细丝等结构细节。

1 引言

长期以来，空间自适应一直是模拟大尺度物理现象的有力工具，包括水、烟雾和固体。总体而言，实现这一目标的关键在于设计适用于非结构化网格的偏微分方程（PDE）离散化方法。在过去几十年中，人们提出了多种离散化技术，例如有限体积法、有限元法、边界元法和谱方法。

这些方法各有优缺点。例如，谱方法由于已知解析解，能够实现精确的时间积分，但往往受模拟边界形状的限制，例如矩形或球体。边界元法将自由度从体积 $O(n^3)$ 缩减为表面积 $O(n^2)$，但只有在能够快速求和空间散布的格林函数时才实用。尽管可以通过快速多极子法（FMM）加速求和，但线性系统无法预先组装，这意味着许多已证明有效的预条件器无法使用。

在流体领域，有限体积法（FVM）历来是最受青睐的选择，因为有限体积法能够自然地将不可压缩性编码为控制体内动量通量的净值为零。此外，所得线性系统是稀疏的（例如，在均匀网格上，拉普拉斯算子对应的每行只有六个非零项）。另一方面，当网格结构偏离均匀单元时，有限体积法的复杂度会迅速增加。具体而言，需要显式计算沃罗诺伊单元（或幂图粒子）及其对应的面。相比之下，有限元法（FEM）能够优雅地处理这类情况。这是因为有限元法将偏微分方程编码为局部支撑域上的体积分（又称弱形式）；为此，需要通过求积规则计算不规则形状单元的精确体积分。求积规则只需要知道从空间坐标到单位坐标的映射，

对于四面体或六面体网格，这种映射很容易确定。作为灵活性的代价，有限元法通常比有限体积法计算开销更大，因为节点（本质上即自由度）会与对角网格元素相互作用。例如，在均匀网格上，有限元法中的线性化拉普拉斯算子每行会产生 27 个非零项。

本文旨在根据需要在空间上选择使用有限体积法或有限元法。为此面临的一项主要技术挑战，是如何设计这两种方法的强耦合离散化，同时尽可能减少伪影。作为应用示例，我们提出布置局部移动网格，以减少数值扩散。总之，我们的贡献总结如下。

- 有限元法与有限体积法强耦合压力求解：我们提出一种新方法，结合使用有限元法和有限体积法计算压力，并在自由表面处采用最适用的二阶精度边界条件。
- 基于移动最小二乘法的空间插值：对于空间上彼此相邻且采用不同离散化方法的区域，我们使用移动最小二乘法近似来插值速度和水平集。
- 移动网格：我们提出沿速度场移动局部网格，并证明此举可以减轻平流造成的数值扩散。

我们通过各种测试验证了有限元法与有限体积法组合方案的数值精度。此外，我们还比较了使用和不使用移动网格时的模拟结果。

2 相关工作

计算流体力学的研究历史悠久。在图形学领域，流体动画可追溯至 Stam [1],，他首次将半拉格朗日平流格式引入图形学。后来，Fedkiw 等人 [2] 提出了一种烟雾模拟框架，其中引入了涡量限制。迄今为止，一种标准的基于网格的液体模拟方法结合了标记点-单元（MAC）网格与水平集方法 [3, 4]。许多研究人员对这些方法进行了扩展，例如通过考虑单元中心到液体表面的距离来施加二阶精度的狄利克雷边界条件 [5, 6] 。关于这些流体模拟发展历史的全面综述，可参见 Bridson [7]。

八叉树网格是最常见的空间自适应方法之一。在计算力学领域，一些此类研究采用非对称的泊松方程离散化来求解压力 [8]。后来，人们开发了可动态改变八叉树结构的烟雾和液体模拟方法 [9, 10], 。由于这些方法能够大幅缩短计算运行时间，因此得到了广泛应用。上述方法后来得到进一步改进，能够在 T 形交界处以二阶精度处理压力 [11]。借助切割单元法或幽灵流体方法，这些方法还可扩展为在固体 [12, 13]和液体表面 [6]上实现二阶精度。然而，由于这些方法基于有限体积法，在 T 形交界单元附近实现精确离散化并非易事。一些研究人员提出了可行的解决方案，例如使用幂图 [14]或修改泊松方程离散化 [15]。

非结构化网格中的四面体网格和六面体网格是实现自适应性的最佳选择之一。Klingner 等人[16] 最先将空间自适应四面体网格用于烟雾模拟，Chentanez 等人 [17] 则将其扩展至液体模拟。虽然这些方法存在 T 形交叉问题，但 Batty 等人 [18] 提出了一种不存在该问题的四面体方法，而沃罗诺伊分解 [19] 则是另一种能够处理边界的方法。不过，这些离散化方法本身较为复杂，尤其是用于压力求解的泊松方程；此外，计算成本和内存消耗仍令人担忧。

English [20] 和 Fan [21]提出了使用移动网格的方法。前者会为每个移动网格准备不同的网格，并在构建压力求解线性系统时使用沃罗诺伊单元将它们拼接起来。后者的研究针对固体模拟，但尚未应用于流体。除了移动网格之外，网格-粒子混合方法也是避免数值扩散的可行替代方案，例如流体隐式粒子法（FLIP） [22] 和扩展窄带流体隐式粒子法（EXNBFLIP） [23]。这些方法有望提升对弹道式液体运动等细节的表现。将其与移动网格结合，预计即使采用较大的时间步长也能实现稳定计算，而我们的方法或许可以与这类方法集成。

3 方法

我们的流体求解器旨在数值求解不可压缩欧拉方程

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \boldsymbol{g}, \quad \nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

其中 $\boldsymbol{u}$，t，ρ，p和 $\boldsymbol{g}$分别表示速度、时间、密度、压力和重力。在离散化过程中，我们将$\rho = 1$设为液体，将 $\rho = 0$ 设为空气。对于边界条件，我们在液体-空气边界处施加狄利克雷边界条件，在液体-固体边界处施加诺伊曼边界条件，使得 $\boldsymbol{n} \cdot \nabla p = 0$ 其中 $\boldsymbol{n}$表示固体法向量。我们采用基于CFL条件的自适应时间步长方法，CFL数约为 2。关于这些设置的理由，请参阅Bridson的书 [7]。
我们在算法1中概述了算法。本节首先讨论如何通过有限体积法和有限元法相结合来离散化模拟域。接着，在第3.2节中，我们详细说明如何构建用于压力求解的线性系统，以及如何处理边界条件。最后，我们讨论如何利用移动网格处理平流，以及如何对定义在网格上的量进行插值。

算法 1方法概述

1：计算时间步长 2：外推速度 3：更新移动网格的位置 4：对水平集或密度进行平流 $\boldsymbol{u}_t$5：对速度进行平流 $\boldsymbol{u}_t$6：重新初始化水平集 7：求解压力并投影速度 8：更新移动网格的速度

3.1 离散化

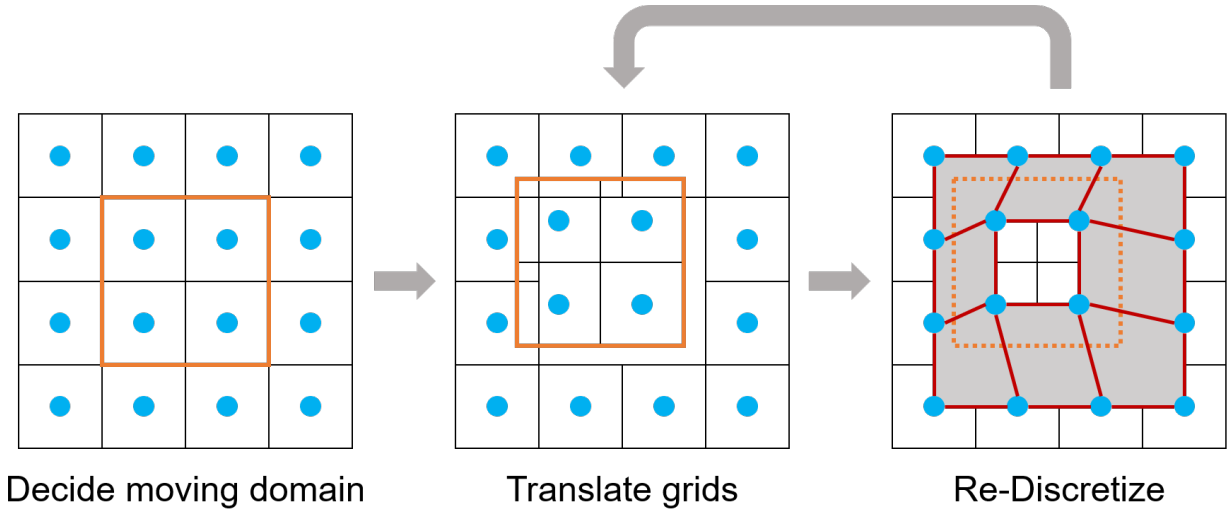

图1：移动网格的离散化概览。我们预先指定移动网格的区域（左）。模拟过程中，根据网格速度 $\boldsymbol{u}_g$ 更新移动网格的位置（中）。我们仅在移动网格的边缘使用有限元法（灰色区域）（右）。

本文大部分采用有限体积法，仅在带有移动网格的边界附近采用有限元法。图 1 展示了离散化过程的概览。为简化压力求解，我们将单元中心点作为节点来构建有限元。设置移动网格时，我们会指定其移动轴。移动网格的速度 $\boldsymbol{u}_g$ 设为单元内的最大速度。

3.2 压力求解

流体模拟中的压力问题可以转化为求解以下动能最小化问题 [12],

$$\underset{p}{\text{minimize}} \int_{\Omega} \frac{1}{2}\rho \left\| \boldsymbol{u}^* - \frac{\Delta t}{\rho}\nabla p \right\|^2 dV, \tag{2}$$

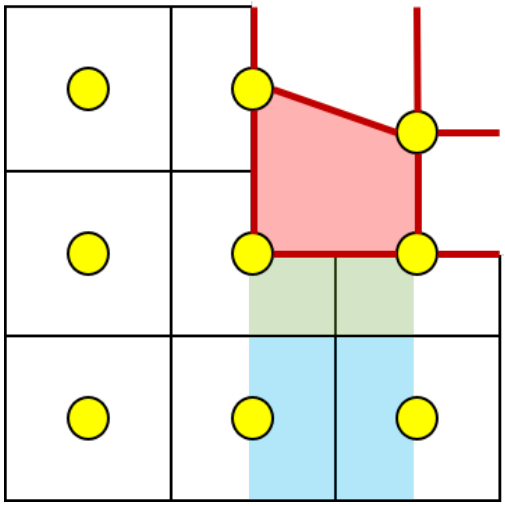

图2：体积积分与控制体积调整。有限体积法中的控制体以蓝色标示，为 Δx^2。红色区域通过高斯求积进行积分。在移动网格附近，控制体会调整为绿色区域（$0.5\Delta x^2$）。

其中， $\boldsymbol{u}^*$ 表示待投影的速度，Ω 表示整个流体域。由于我们方法中的压力求解基于 FVM 和 FEM，下面先简要介绍这两种方法。

首先，我们介绍有限体积法，它在我们的方法中占据主要部分。为计算压力，我们求解以下线性系统 [15],

$$[\nabla]^T[V][A][F][\nabla]\{p\} = [\nabla]^T[V][A]\{\boldsymbol{u}^*\}, \tag{3}$$

其中 $[\nabla]^T$， $[\nabla]$， $[V]$， $[A]$，和 $[F]$ 分别表示离散散度算子、离散梯度算子、各面的控制体、流体面积分数，以及各面液体分数的倒数，用于实现狄利克雷边界条件下的二阶精度 [6]。

对于有限元法中的压力求解，我们采用 Ibayashi 等人提出的方法 [24]。与各种有限元方法类似，我们首先将空间坐标 $\boldsymbol{x}$ 映射到单位坐标 $\boldsymbol{\xi} = [\xi, \eta, \zeta]$。在这种设定下，三维中任意坐标处的量 q 可改写如下，

$$q(\boldsymbol{\xi}) = \sum_{i=1}^{8} N_i(\boldsymbol{\xi}) q_i, \tag{4}$$

其中， N_i 是三线性形函数，定义在每个单元的节点处。在此假设下，我们构造如下线性系统:

$$\sum_{e=1}^{n} \int_{\Omega_e} [\nabla_e]^T |J| [D] [\nabla_e] \{p_e\} dV_\xi = \sum_{e=1}^{n} \int_{\Omega_e} [\nabla_e]^T |J| \boldsymbol{u}^* dV_\xi, \tag{5}$$

其中， $\{p_e\} \in \mathbb{R}^{8\times 1}$ 是存储在第 e 个单元节点上的压力， J 是雅可比矩阵。 $[\nabla_e]$ 是梯度算子，定义如下:

$$[\nabla_e]_{ij} = [J^{-1}\nabla_\xi N(\boldsymbol{\xi})]_i, \qquad \nabla_\xi = \left(\frac{\partial}{\partial\xi}, \frac{\partial}{\partial\eta}, \frac{\partial}{\partial\zeta}\right)^T. \tag{6}$$

$[D]$定义如下:

$$[D] = \frac{1}{\rho} M_\phi, \tag{7}$$

其中， M_ϕ 是一个矩阵，用于使 Ibayashi 等人 [24] 提出的压力求解达到二阶精度。我们使用八点高斯求积积分方案对式（5）进行积分。为简化计算，我们使用在高斯求积点处插值得到的各单元速度。

比较式(3)和式(5)可知，有限体积法中的体积因子 $[V]$ 和二阶算子 $[F]$ 分别对应有限元法中的 $|J|$ 和 $[D]$。由于两种方法使用了相同的压力定义空间点，

因此我们认为每个方程中的压力 p 完全一致。由此，我们可以通过组合对应的矩阵来构建线性系统，从而正确计算压力。需要注意的是，有限元法和有限体积法在交界处的积分域存在重叠。我们根据有限元单元的占用情况调整有限体积法的控制体来处理这一问题，如图2中的绿色区域所示。由于有限元法采用高斯求积插值方案并将其映射到单位坐标，因此难以调整积分范围。

我们对二阶算子进行投影，使线性系统成为数值稳定的对称正定矩阵（SPD），这一点已由先前的研究证实。因此，精度会有所折损，但我们没有观察到这种投影造成任何视觉伪影。关于倾斜水池，与完全二阶求解器的比较将在第5节中讨论。

计算出压力后，我们必须将速度投影到无散场，因此要减去压力梯度

$$\{\boldsymbol{u}\} = \{\boldsymbol{u}^*\} - [F][\nabla]\{p\}. \tag{8}$$

对于有限元法，速度定义在单元中心，因此减法项是根据高斯求积插值方案的计算点求平均得到的。

3.3 插值

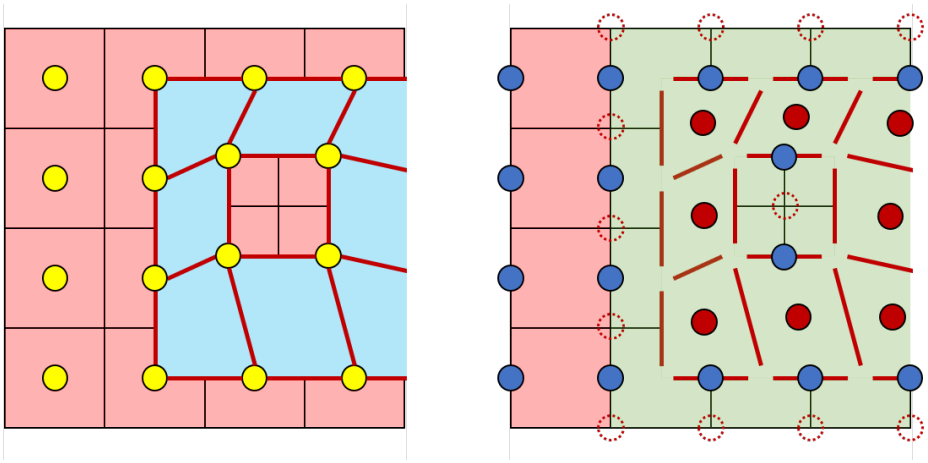

图3：针对单元中心值（左）和面中心值（右）的插值方案。黄色、深蓝色、深红色圆点以及深红色空心圆分别表示单元中心值、有限体积法的面中心值、有限元法的面中心值，以及不存在该单元的节点位置。

对于规则分布的点，在有限体积法和有限元法区域中均使用三线性插值来插值这些值。对于两种方法交界的区域，则根据物理量的位置采用不同的策略。

首先，考虑单元中心值。由于单元中心值的位置是共享的，我们只需根据待插值的位置属于有限体积法区域还是有限元法区域，切换插值方法。对于有限体积法区域（图 3 中的红色区域），我们采用三线性插值对数值进行插值；对于有限元法区域（蓝色区域），则使用逐单元插值，并通过牛顿法计算单位坐标。

与单元中心值不同，有限体积法和有限元法的面中心值位置并不重合；因此，对于边界附近（绿色区域）的面中心值，我们采用移动最小二乘法（MLS）插值方法。正如

sAndo 等人所建议的， [15] 我们根据以下公式进行 MLS 插值，

$$q(\boldsymbol{p}) = [p_x \quad p_y \quad p_z \quad 1]\,(Z^T D Z)^{-1} Z^T D \begin{bmatrix} \{q\}_1 \\ \vdots \\ \{q\}_n \end{bmatrix}, \tag{9}$$

$$Z = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \\ x_n & y_n & z_n & 1 \end{bmatrix}, \quad D = \operatorname{diag}\left(\begin{bmatrix} N(\boldsymbol{p} - \boldsymbol{x}_1, \Delta x) \\ \vdots \\ N(\boldsymbol{p} - \boldsymbol{x}_n, \Delta x) \end{bmatrix} \right), \tag{10}$$

其中， $\boldsymbol{p} = [p_x, p_y, p_z]$， $\boldsymbol{x}_i = [x_i, y_i, z_i]$， n，以及 q 分别表示查询点、第 i 个采样点的位置、采样点的数量，以及离散点处的变量。我们通过不同的方式确定权重 N，分别用于有限体积法和有限元法的值。对于有限体积法的权重，我们采用

$$N_{FVM}(\boldsymbol{r}, h) = \prod_{j=1}^{3} \max(1 - \frac{||\mathrm{r_j}||}{\mathrm{h}}, \epsilon), \tag{11}$$

其中， ϵ 为安全常数值。该权重与简单的三线性插值权重一致。对于有限元法，我们首先计算单位坐标 $\boldsymbol{\xi}$，并将单元插值的系数用作有限元法的权重。为了计算$\boldsymbol{\xi}$，我们考虑由连接速度定义点（即每个单元中心）构成的虚拟单元。如图 3 所示，在未定义单元的区域，我们假设单元中心位于网格节点处，并基于这些虚拟单元计算 $\boldsymbol{\xi}$。如果采集到的采样点不足，可以通过从周围区域采集采样点，并赋予权重 ϵ，来稳定移动最小二乘插值。

3.4 平流

我们对速度和水平集平流都采用半拉格朗日平流格式 [1] 。在我们的方法中，物理量的定义点会在每个仿真步更新。因此，我们需要注意平流项中所用物理量的位置。采用半拉格朗日方法更新数值的方程如下:

$$q^*(\boldsymbol{x}, t) = q(\boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x}, t)\Delta t, t) \tag{12}$$

其中， q^* 表示平流后的中间物理量， q 表示将要进行平流的物理量。在我们的方法中，可以认为 q 是更新前定义在网格上的物理量。幸运的是，我们可以将式（12）视为使用相对速度进行的平流，因为它与基于以下平流方程的推导一致:

$$q^*(\boldsymbol{x}, t) = q(\boldsymbol{x} - \boldsymbol{u}_g\Delta t - (\boldsymbol{u}(\boldsymbol{x}, t) - \boldsymbol{u}_g)\Delta t, t), \tag{13}$$

其中， q 的自变量中的第二项表示移动网格位移，第三项表示由相对速度引起的平流。这种策略可以视为 Chimera Grids [20] 所提出平流方法的简化版本。我们的方法与其方法的区别在于，由于我们的插值方法具有第 3.3 节所述的连续性，因此无需准备多层虚单元。

4 结果

我们在一台搭载 AMD Ryzen 9 5950X 的 Linux 机器上进行了一些实验。我们使用 Eigen 库 [25] 求解线性系统，并将压力求解的相对残差容差设为 10^{-4} 。我们使用基于偏微分方程的方法 [26] 进行水平集重初始化。我们使用 marchingcubes [27] 提取液面，并通过 Mitsuba [28]进行可视化。本节将评估有限元法与有限体积法相结合的精度，并展示使用移动网格进行模拟的优势。

4.1 精度评估

我们使用线性场或二次场评估插值方法，结果如图4和图5所示。我们设置了一个位移网格，并在其周围使用有限元法以及移动网格的离散化。我们将范围设为 1×1 无量纲场，每个场由以下方程表示。

$$\text{Linear}: f(x, y) = 0.5(x + y) \tag{14}$$

$$\text{Quadratic}: f(x, y) = 2\left\{(x - 0.5)^2 + (y - 0.5)^2\right\}. \tag{15}$$

分辨率	单元中心	面中心
16^2	5.332e-3（N / A）	4.349e-3（N / A）
32^2	1.333e-3(2.00)	1。086e-3(2。00)
64^2	3。324e-4(2。00)	2.716e-4(2.00)
128^2	8.325e-5(2.00)	6.726e-4(2.01)
256^2	2.047e-5(2.02)	1.669e-5(2.01)

表1：我们在二次场上的插值方案的 L_∞ 误差和精度阶数。

我们将误差转换为以下对数尺度，

$$f(x, y) = 1.0 + 0.1 \log_{10} |\text{error}(x, y)|, \tag{16}$$

并将其以热图形式可视化。我们的插值方法能够以极小的误差重现线性场。由于该方法基于三线性插值，因此在二次场上会产生一些误差，但随着分辨率加倍，误差以二阶速度减小。（表1）。

我们还通过水平及倾斜水池测试检验了压力求解的准确性。如图2所示，采用一阶精度的压力求解无法将速度投影为无散场；但采用二阶压力求解则可以获得无散速度。

4.2 使用移动网格的模拟结果

	图7		图7的半分辨率	
	上一页	Ours	上一个	Ours
平流	92.01	102.05	4.27	6.20
投影	1080.55	1097.69	35.48	37.21
更新移动网格	N/A	32.32	N/A	1.69
总计	1188.76	1249.69	40.67	46.07

表2：每帧耗时（秒）在图7所示场景及其分辨率减半时，前一种方法和我们的方法 .

我们使用自己提出的方法进行了一些实验，并将结果与先前的方法 [5] 进行了比较。表2显示了图7场景中的计算时间。

图7展示了液体物体下落实验的结果。在这项测试中，我们设置移动网格，使其速度与下落物体的速度相同。这可以避免物体细节缺失。从结果可以看出，采用移动网格的方法能够抑制数值扩散的影响。与不使用我们技术的标准液体求解相比，我们的方法耗时长 1.05 倍，尤其是平流部分耗时长 1.11 倍。分辨率减半时，我们的方法计算耗时长 1.13 倍，平流耗时长 1.45 倍。我们将在第5节讨论这一点。

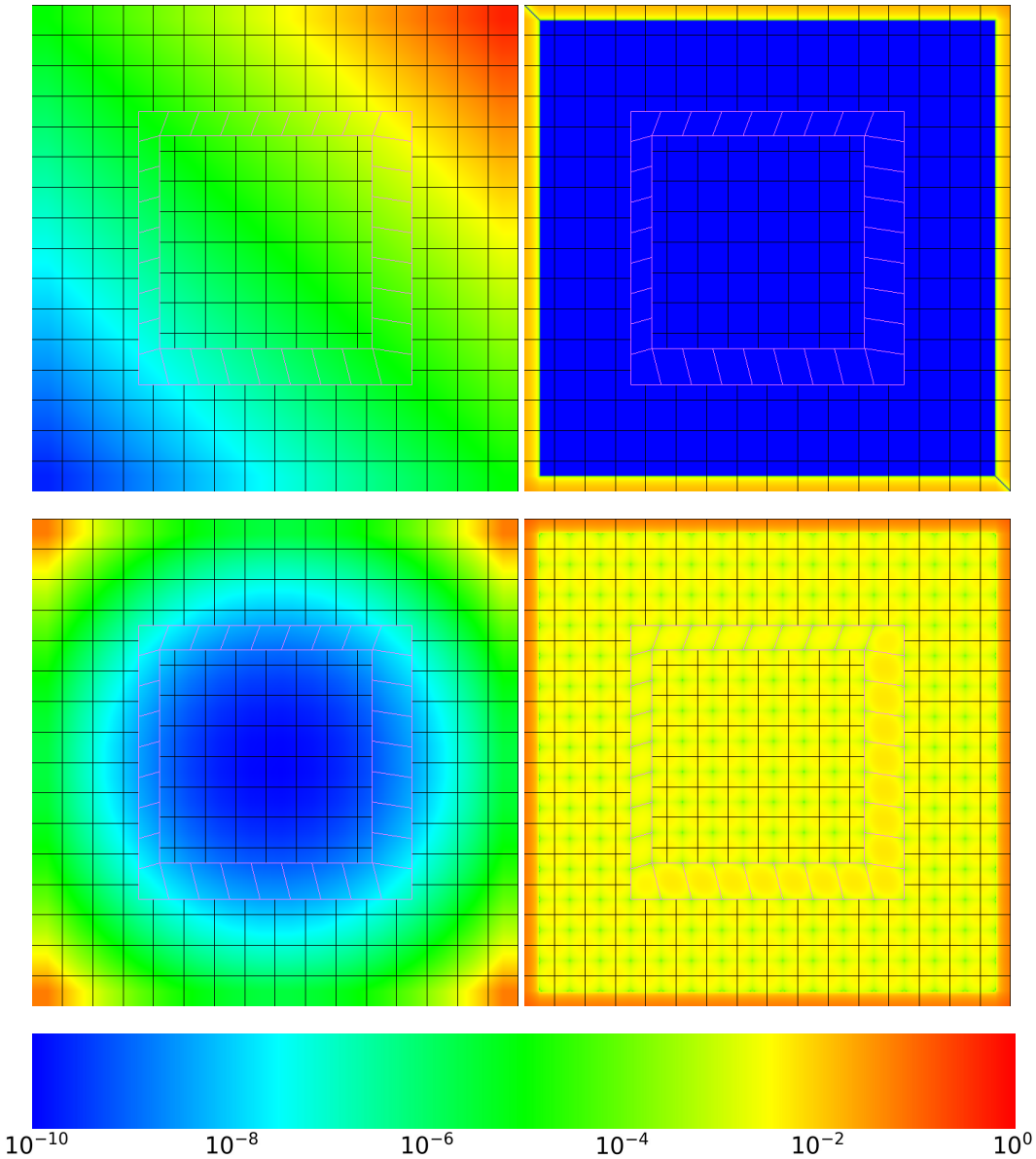

图4：线性（上）和二次（下）场的单元中心值插值结果。左：重构场。右：误差可视化。

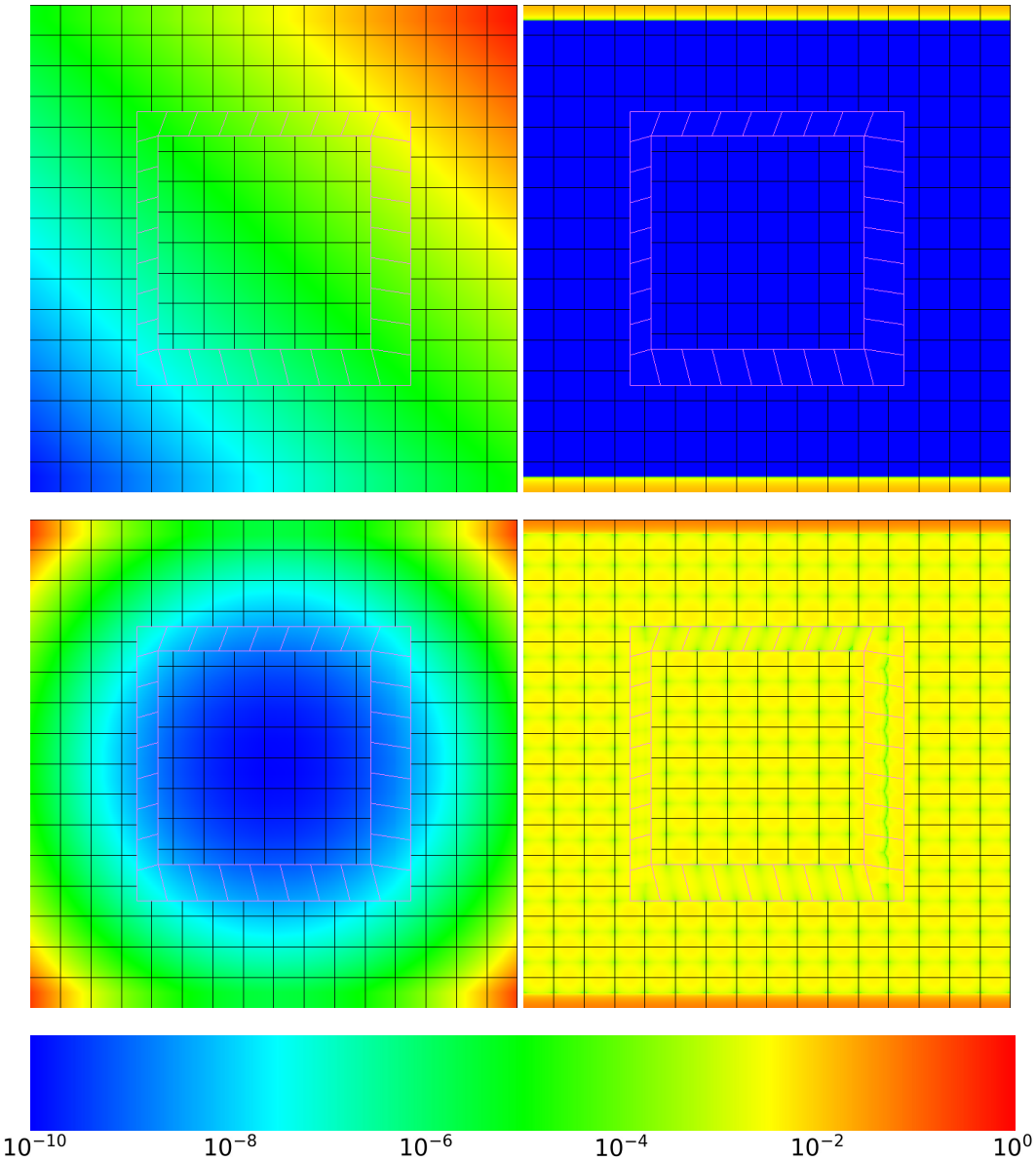

图5：线性（上）和二次（下）场的面中心值插值结果。左：重构场。右：误差可视化。

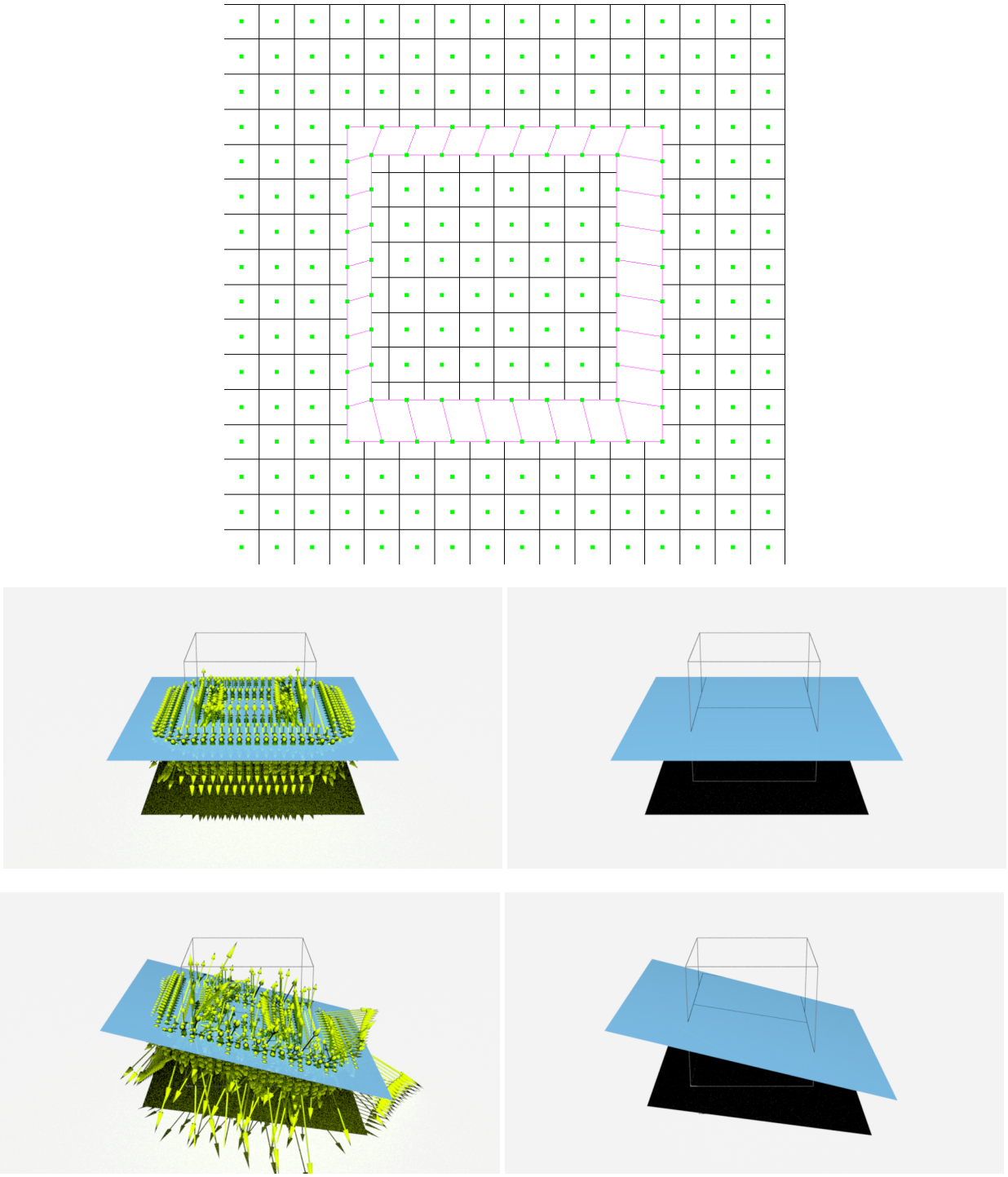

图 6：我们的压力求解器在水平水池和倾斜水池上的测试结果。上图展示了二维网格几何的简单可视化。我们沿垂直于液面的方向施加强重力。黄色箭头表示投影后的速度。左：朴素的一阶投影。右：二阶投影（SPD）。

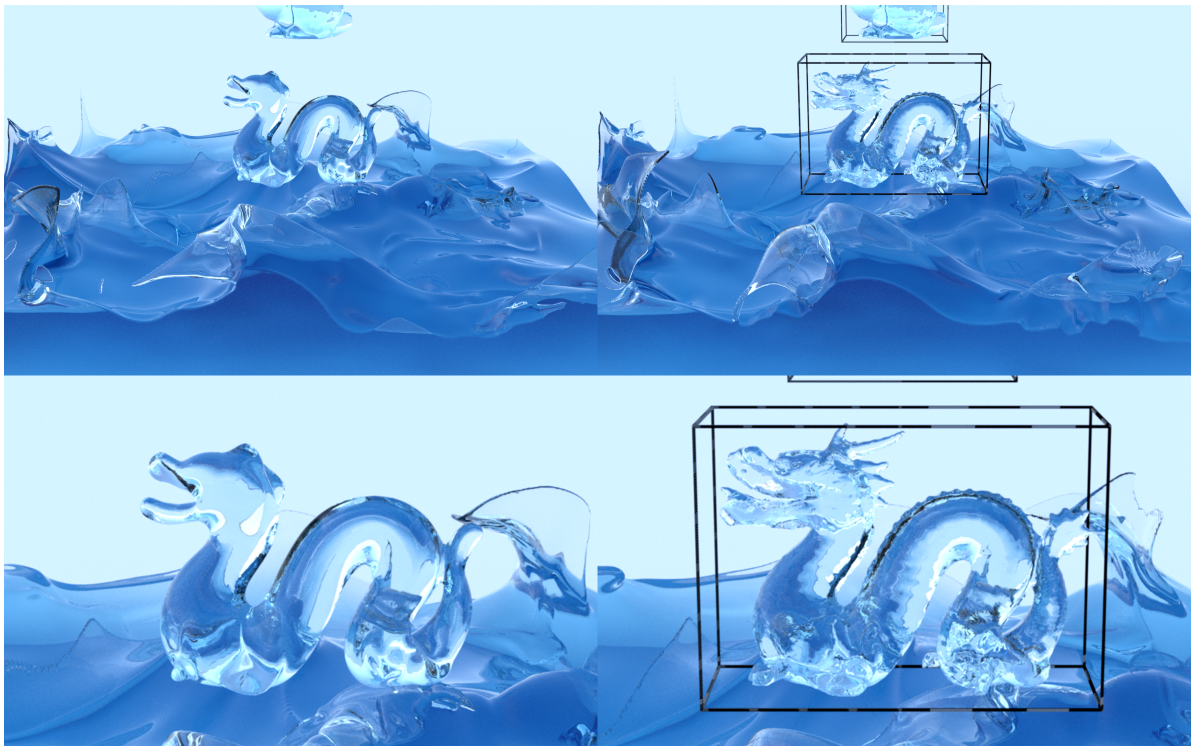

图7：比较 以不同分辨率网格模拟液体物体落入水池时，先前方法与所提出方法的效果 f 600^3分辨率。左： 先前的方法。右：我们的方法。

5 讨论与结论

本文提出了一种用于液体模拟的新型欧拉模拟框架，该框架结合了有限元法和有限体积法。我们展示了基于移动最小二乘法的插值方案以及新型压力求解的数值稳健性。此外，我们展示了该方法在移动网格中的适用性，并表明该方法能够抑制数值扩散。

如图4和图5中的二次场插值测试所示，即使采用三线性插值，随着评估点远离定义点，误差也会变得更大。这表明，使用与液体速度相同的移动网格覆盖网格区域，可以使插值在定义点附近进行，从而减小误差。

至于压力求解器，我们发现，如图8所示，我们投影得到的SPD产生了一些误差。然而，正如先前的工作 [15], 所述，使用SPD矩阵是一种令人满意的替代方案，因为它具有稳定且收敛速度快的特点。

尽管我们的方法目前能够有效抑制数值扩散，但如果在CFL条件下计算时间步长时使用相对速度进行优化，也可能有效减少计算运行时间。

在性能方面，我们目前仍处于实验阶段，因此尚未针对提高运行性能对代码进行优化。特别是在平流方面，移动网格边界中MLS插值采样点的搜索尚未优化，导致进行了许多不必要的计算。然而，由于我们的方法仅限于在移动网格边界使用有限元法，随着分辨率提高，有限元法部分所占比例降低，因此计算开销相对较小。当然，优化十分重要，但使用有限元法所带来的低开销正是我们方法的优势之一。
在我们的实验中，我们采用了六面体单元；不过，我们认为也可以使用四面体单元，但代价是线性插值的精度会有所降低。
未来的研究中，如果能够自适应地确定移动网格区域，并考虑旋转运动，将更有意义。我们也在考虑采用有限元处理八叉树网格中由 T形交叉引起的复杂问题的可能性 [15]。

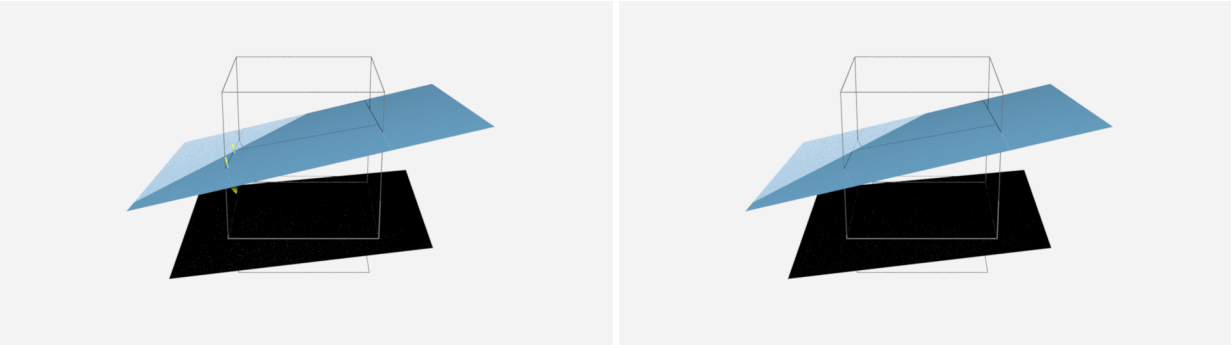

图 8：SPD 与二阶精度的比较。在倾斜水池中，我们观察到 SPD 存在一些误差，但在采用完全二阶精度的情况下则没有。

致谢

本研究得到了日本学术振兴会青年科学家研究资助金（18K18060）的支持。

参考文献

[1] Jos Stam。稳定流体。载于第*26*届计算机图形学与交互技术年度会议论文集，SIGGRAPH '99，第121–128页，美国纽约州纽约市，1999年。ACM Press/Addison-Wesley Publishing Co.[2] Ronald Fedkiw、Jos Stam 和 Henrik Wann Jensen。烟雾视觉模拟。载于第*28*届计算机图形学与交互技术年度会议论文集，SIGGRAPH '01，第15–22页，美国纽约州纽约市，2001年。计算机协会。[3] Francis H. Harlow 和 J. Eddie Welch。自由表面流体的时变黏性不可压缩流动数值计算。《流体物理》，8(12):2182–2189，1965年12月。[4] N. Foster 和 D. Metaxas。流体动画控制。载于国际计算机图形学会议论文集，第178–188页，1997年。[5] Nick Foster 和 Ronald Fedkiw。液体实用动画。SIGGRAPH '01，第23–30页，美国纽约州纽约市，2001年。计算机协会。[6] Doug Enright、Duc Nguyen、Frederic Gibou 和 Ron Fedkiw。将粒子水平集方法与二阶精度压力边界条件用于自由表面流动。载于第*4*届*ASME-JSME*联合流体工程会议论文集，编号*FEDSM2003–45144*。美国机械工程师学会，第2003–45144页，2003年。[7] Robert Bridson。《计算机图形学流体模拟》。A K Peters/CRC Press，2008年9月。[8] Stéphane Popinet。Gerris：一种用于复杂几何形状中不可压缩欧拉方程的基于树的自适应求解器。《计算物理学杂志》，190:572–600，2002年10月。[9] Lin Shi 和 Yizhou Yu。采用自适应八叉树细化的烟雾视觉模拟。2004年8月。[10] Frank Losasso、Frédéric Gibou 和 Ron Fedkiw。利用八叉树数据结构模拟水和烟雾。*ACM*图形学汇刊，23(3):457–462，2004年8月。[11] Frank Losasso、Ronald Fedkiw 和 Stanley Osher。水平集方法和不可压缩流动的空间自适应技术。《计算机与流体》，35(10):995–1010，2006年。[12] Christopher Batty、Florence Bertails 和 Robert Bridson。用于精确固体-流体耦合的快速变分框架。*ACM*图形学汇刊，26(3):100，2007年。[13] Yen Ting Ng、Chohong Min 和 Frédéric Gibou。单相流的高效流体-固体耦合算法。《计算物理学杂志》，228(23):8807–8829，2009年12月。[14] Mridul Aanjaneya、Ming Gao、Haixiang Liu、Christopher Batty 和 Eftychios Sifakis。用于高分辨率自适应液体模拟的幂图与稀疏分页网格。*ACM*图形学汇刊，36(4)，2017年7月。[15] Ryoichi Ando 和 Christopher Batty。具有自适应表面分辨率的实用八叉树液体模拟器。*ACM* 图形学汇刊，39(4)，2020 年7月。

[16] Bryan M. Klingner、Bryan E. Feldman、Nuttapong Chentanez 和 James F. O'Brien。采用动态网格的流体动画。载于*ACM SIGGRAPH 2006* 论文集，SIGGRAPH '06，第820–825页，美国纽约州纽约市，2006年。计算机协会。[17] Nuttapong Chentanez、Bryan E. Feldman、François Labelle、James F. O'Brien 和 Jonathan R. Shewchuk。基于格子的四面体网格上的液体模拟。载于*2007*年*ACM SIGGRAPH/Eurographics*计算机动画研讨会论文集，SCA '07，第219–228页，德国戈斯拉尔，2007年。欧洲图形学协会。[18] Christopher Batty、Stefan Xenos 和 Ben Houston。用于精确灵活自适应流体模拟的四面体嵌入边界方法。载于*Eurographics*论文集，2010年。[19] Tyson Brochu、Christopher Batty 和 Robert Bridson。使流体模拟单元与表面几何和拓扑相匹配。*ACM*图形学汇刊，29(4)，2010年7月。[20] R. Elliot English、Linhai Qiu、Yue Yu 和 Ronald Fedkiw。用于水模拟的Chimera网格。载于第*12*届*ACM SIGGRAPH/Eurographics*计算机动画研讨会论文集，SCA '13，第85–94页，美国纽约州纽约市，2013年。计算机协会。[21] Ye Fan、Joshua Litven、David I. W. Levin 和 Dinesh K. Pai。欧拉-拉格朗日模拟。32(3)，2013年7月。[22] Yongning Zhu 和 Robert Bridson。将沙子作为流体进行动画模拟。*ACM*图形学汇刊，24(3):965–972，2005年7月。[23] Takahiro Sato、Chris Wojtan、Nils Thuerey、Takeo Igarashi 和 Ryoichi Ando。用于液体模拟的扩展窄带FLIP。计算机图形学论坛，2018年。[24] H. Ibayashi、C. Wojtan、N. Thuerey、T. Igarashi 和 R. Ando。在动态扭曲网格上模拟液体。*IEEE*可视化与计算机图形学汇刊，第1–1页，2018年。[25] Gaël Guennebaud、Benoît Jacob 等。Eigen v3。http://eigen.tuxfamily.org，2010年。[26] Giovanni Russo 和 Peter Smereka。关于计算距离函数的说明。《计算物理学杂志》，163:51–67，2000年。[27] William E. Lorensen 和 Harvey E. Cline。行进立方体：一种高分辨率三维表面构造算法。*SIGGRAPH*计算机图形学，21(4):163–169，1987年8月。[28] Wenzel Jakob。Mitsuba渲染器，2010年。http://www.mitsuba-renderer.org。