



中国科学院大学
University of Chinese Academy of Sciences

博士学位论文

面向粘性不可压缩流体仿真的 SPH 法数值稳定性研究

作者姓名: 刘树森

指导教师: 王文成 研究员

中国科学院软件研究所

学位类别: 工学博士

学科专业: 计算机应用技术

培养单位: 中国科学院软件研究所

2022 年 12 月

Research on Numerical Stability of SPH Method for Simulating
Viscous Incompressible Fluids

A dissertation submitted to
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Doctor of Philosophy
in Computer Application Technology
By
LIU Shusen
Supervisor: Professor WANG Wencheng

Institute of Software, Chinese Academy of Sciences

December, 2022

中国科学院大学
研究生学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。承诺除文中已经注明引用的内容外，本论文不包含任何其他个人或集体享有著作权的研究成果，未在以往任何学位申请中全部或部分提交。对本论文所涉及的研究工作做出贡献的其他个人或集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日 期：

中国科学院大学
学位论文使用授权声明

本人完全了解并同意遵守中国科学院大学有关收集、保存和使用学位论文的规定，即中国科学院大学有权按照学术研究公开原则和保护知识产权的原则，保留并向国家指定或中国科学院指定机构送交学位论文的电子版和印刷版文件，且电子版与印刷版内容应完全相同，允许该论文被检索、查阅和借阅，公布本学位论文的全部或部分内容，可以采用扫描、影印、缩印等复制手段以及其他法律许可的方式保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：

日 期：

导师签名：

日 期：

摘要

光滑粒子流体动力学（Smoothed Particle Hydrodynamics，缩写为 SPH）是实现流体仿真的主要技术之一。相比其它的流体仿真方法，SPH 法具有许多优点，包括：仿真过程严格地保证质量守恒、易于划分材料边界、易于实现不同材料之间的耦合等。但是 SPH 法存在着数值计算不稳定问题，限制了其对粘性不可压缩流体的仿真应用。具体地，有以下几方面的不足：(1)由于粘性求解与不可压缩性求解之间存在冲突，SPH 法在模拟大粘性流体时稳定性较差；(2)当流体处于拉伸状态时，SPH 法会使粒子产生积块，导致其模拟的流体溅射、细丝、薄层等小尺度运动细节稳定性很差；(3)由于流体的物理场与微分都在同一组粒子进行相关计算，会引入零能问题（Zero Energy Mode），加剧了 SPH 法数值计算的不稳定；(4)SPH 法的相关计算精确性很容易受到粒子分布不均匀情况的影响。本论文针对 SPH 法数值计算稳定性方面的不足开展了研究，取得了如下的进展：

(1) 针对 SPH 法中流体粘性计算与不可压缩性计算之间的冲突问题，本文将 SIMPLE (Semi-Implicit Method For Pressure Linked Equation) 迭代算法引入到了 SPH 法中，并改进以抑制不可压缩性计算与粘性计算之间的干扰。测试表明改进后的方法能够稳定地收敛，而且只需要若干次迭代即可生成视觉真实感很高的大粘性流体仿真结果，并能够产生更锐利、更丰富的表面细节，更稳定地保持流体的形态。

(2) 对于 SPH 流体的拉伸不稳定的问题，本文在 SPH 法中引入对偶粒子，将标准的 SPH 粒子划分为带有速度属性的实粒子和作为压强采样点的虚粒子，并首次基于对偶粒子构建了流体的不可压缩性投影求解算法。由于引入的虚粒子可缩小 SPH 法压强梯度计算的拉伸不稳定区域，并抑制粒子扰动对于流体压强的影响，同时可避免零能问题的产生，因此该方法增强了 SPH 流体的拉伸稳定性，改善了流体小尺度细节仿真的质量。在此，本文研究发现均匀分布的虚粒子对于流体稳定性改善至关重要，因此提出了基于 GPU 的空间自适应虚粒子快速生成方法，以高效高质量地实现稳定的流体仿真。

(3) 本文提出的对偶粒子仿真框架包含的虚粒子，不仅可作为拉格朗日视角的粒子，也可作为欧拉视角的空间网格。因此本文提出的对偶粒子框架能够兼容标准 SPH 法、对偶粒子法、粒子网格混合法等各种粒子仿真方法。由此，本文构建了一个通用的流体仿真引擎。在使用该引擎进行的对比研究中发现，粒子网格混合法中粒子分布非常不均匀。因此，本文提出了使用 SPH 粒子位置修正技术以均匀地分布粒子网格混合法中的粒子，改善了此类方法的仿真稳定性。

关键词：计算机图形学，物理仿真，光滑粒子流体动力学，粘性不可压缩流

体，拉伸不稳定

Abstract

Smoothed Particle Hydrodynamics (SPH) is one of the main techniques to simulate fluids. Compared with other fluid simulation methods, the SPH method has many advantages: easy to guarantee the mass conservations, easy to capture the boundary, and easy to simulate the coupling of different materials, etc. Therefore, the method has important applications in visual-reality, computer games, animation, special-effect in film and so on. However, despite years of development, the SPH method still suffers from serious numerical instability, which limits its use in application: (1)The interference between pressure and viscosity solvers leads to the poor stability of the SPH method in simulating high viscous fluids; (2)The SPH method suffers from the tensile instability problem, in which the particles will tend to clump in pairs under tensions and result in a bad visual-quality of small-scaled features, including droplets, streamlines and sheets; (3)The physical fields and their differentiation of fluids are all defined on the same particles, which will introduce the zero-Energy mode and exacerbate the tensile instability further; (4) The accuracy of the SPH method is easily affected by the non-uniform distribution of particles, which leads to the poor stability of fluid simulations. In this thesis, relevant targeted research has been carried out to address the numerical instability of the SPH fluid, and the following progress has been achieved.

For the interference between the viscosity and incompressibility of the SPH fluids, we introduce the SIMPLE (Semi-Implicit Method for Pressure Linked Equation) iterative algorithm to the particle methods, which can effectively suppress the interferences. Tests show that our method converges stably and only requires a few iterations to generate high visual quality results for large viscous fluids. Our method is able to model remarkable viscous behaviors with a good property of preserving sharp surface details.

For the stability issue of SPH fluid simulation, we propose a dual particle framework in the SPH method, which divides the standard SPH particles into real particles that carry velocity, and virtual particles that store the pressure property. Based on the dual particle framework, we devise the projection algorithm to enforce incompressibility. Since the introduced virtual particles can narrow down the unstable region for particles under a tensile stress state, avoid the zero-energy mode, and suppress the effect of particle perturbation on fluid pressures. Our method can weaken the tensile instability of the SPH fluid, and improve the visual quality of the small-scaled features of fluids. Besides, it is found that uniform virtual particles are essential for fluid stability improvement, so a GPU-based spatially adaptive virtual particle generation method is proposed to efficiently obtain well-distributed virtual particles.

Since our dual particle framework contains virtual particles that can be used as Lagrangian particles or Eulerian grids in different simulation strategies. Therefore, our dual-particle framework is compatible with various particle simulation methods, such as the original SPH method, the dual particle method and the hybrid particle-grid method. As a result, a general fluid simulation engine is constructed in this paper. In a comparative study using this engine, it is found that the particle distribution is very uneven in the hybrid particle-grid method. Therefore, we propose to use the SPH particle shifting technique to uniformly distribute the particles in the hybrid particle-grid method to improve its simulation stability.

Key Words: Computer Graphics, Physical Simulation, SPH, Viscous Incompressible Fluid, Tensile Instability

目 录

第 1 章 绪论.....	1
1.1 研究意义.....	1
1.2 研究现状.....	2
1.2.1 流体的快速仿真方法.....	2
1.2.2 SPH 法不可压缩性求解	4
1.2.3 SPH 法粘性求解	5
1.2.4 SPH 流体的数值计算稳定性研究	6
1.3 研究内容.....	7
1.4 本文组织结构.....	9
第 2 章 粘性不可压缩流体与 SPH 法.....	11
2.1 引言.....	11
2.2 物理场、散度与梯度.....	11
2.3 流体运动的控制方程、不可压缩性与粘性.....	12
2.3.1 连续性方程.....	12
2.3.2 动量方程.....	13
2.3.3 不可压缩性条件.....	14
2.4 流体的粘性.....	14
2.5 流体模拟方法概述.....	16
2.5.1 算子分裂法.....	16
2.5.2 SPH 流体模拟流程	17
2.6 SPH 法的离散化方案	18
2.6.1 光滑核函数.....	18
2.6.2 微分算子的离散化.....	20
2.7 本章小结.....	21
第 3 章 适用于粘性不可压缩 SPH 流体的 SIMPLE 算法.....	23
3.1 引言.....	23
3.2 粘性不可压缩性流体 SIMPLE 迭代算法概述	24
3.3 将 SIMPLE 算法扩展到 SPH 法中	26
3.3.1 离散化方案选择.....	27
3.3.2 动量方程与粘性求解.....	27
3.3.3 不可压缩性求解.....	29
3.3.4 迭代与验证.....	33
3.4 测试与讨论.....	37

3.4.1 与其它方法的对比.....	38
3.4.2 牛顿流体的模拟.....	39
3.4.3 非牛顿流体的模拟.....	41
3.4.4 果酱、黄油、奶油的模拟.....	43
3.5 本章总结.....	45
第 4 章 对偶粒子投影法	47
4.1 引言.....	47
4.2 SPH 流体的拉伸不稳定	48
4.3 对偶粒子框架.....	50
4.3.1 SPH 流体与其拉伸不稳定问题	50
4.3.2 使用对偶粒子缩小拉伸状态下的核函数不稳定区间.....	52
4.3.3 对偶粒子的离散化方法.....	52
4.3.4 对偶粒子投影方法实现概述.....	53
4.4 使用对偶粒子的近似投影方法.....	55
4.4.1 压强的泊松方程离散化.....	55
4.4.2 自由表面边界条件.....	56
4.4.3 实粒子速度的更新.....	57
4.5 虚粒子的生成方法.....	58
4.5.1 粒子微移策略（策略 B）虚粒生成方法	59
4.5.2 空间自适应策略（策略 C）虚粒子生成方法	60
4.6 验证与讨论.....	61
4.6.1 拉伸稳定性验证.....	61
4.6.2 稳定性的综合测试.....	62
4.6.3 收敛性测试.....	65
4.6.4 与应力点 SPH 法的对比研究	67
4.7 综合性测试.....	68
4.7.1 与其它粒子法的对比测试.....	68
4.7.2 更大规模的测试.....	73
4.8 本章总结.....	74
第 5 章 基于对偶粒子法的通用流体仿真引擎	77
5.1 引言.....	77
5.2 流体仿真引擎的框架设计.....	78
5.2.1 仿真场景	79
5.2.2 场景节点图与管线	80
5.2.3 数据结构	82
5.2.4 邻域查找	83
5.3 粒子网格混合法求解器.....	83

5.3.1 粒子网格混合法概述.....	84
5.3.2 粒子网格混合法中的对流计算.....	84
5.3.3 粒子网格混合法中的不可压缩性.....	85
5.4 虚粒子采样方法与稀疏空间网格.....	87
5.5 粒子网格混合法的研究.....	88
5.5.1 对偶粒子法与粒子网格混合法的对比研究.....	88
5.5.2 粒子-网格混合法中粒子杂乱问题的改进方法	90
5.6 本章总结.....	91
第 6 章 总结与展望	93
6.1 本文总结.....	93
6.2 未来展望.....	94
参考文献.....	95
附录一 第三章部分测试场景的单帧平均计算时间统计	107
附录二 第四章部分测试场景的单帧平均计算时间统计	109
致谢.....	111
作者简历及攻读学位期间发表的学术论文与其他相关学术成果	117

图目录

图 1-1 流体的仿真方法。	2
图 2-1 梯度与散度。	12
图 2-2 牛顿剪切流动试验。	15
图 2-3 形变速率张量中各元素的物理意义。	16
图 3-1 本文 SIMPLE 算法的基本实现方式。	24
图 3-2 SPH 粒子的邻域缺失问题。	27
图 3-3 半解析式边界条件。	30
图 3-4 不可压缩性求解算法的测试。	32
图 3-5 面向粘性不可压缩 SPH 流体模拟的 SIMPLE 算法。	32
图 3-6 本文方法的内迭代与 SIMPLE 迭代。	33
图 3-7 SIMPLE 算法中内迭代收敛性分析。	33
图 3-8 粘性系数为 $500 Pa \cdot s$ 流体的卷绳效应模拟。	34
图 3-9 使用不同 SIMPLE 迭代次数模拟的粘性流体卷绳效应。 ..	34
图 3-10 SIMPLE 迭代的收敛分析。	35
图 3-11 时间步长与粒子采样距离对于 SIMPLE 迭代收敛速率的影响。	36
图 3-12 SIMPLE 迭代次数对粘性流体行为的影响。	37
图 3-13 流体压曲效应的模拟。	38
图 3-14 流体卷绳效应的模拟。	39
图 3-15 牛顿流体。	40
图 3-16 不同粘性的牛顿流体材料间的碰撞。	40
图 3-17 包含有 100 万个粒子的大粘性流体。	40
图 3-18 Cross 模型 ^[149] 中剪切变稀、剪切增稠、牛顿流体材料的粘性变化曲线。	41
图 3-19 牛顿流体与非牛顿流体的对比。由左至右分别为低粘性牛顿流体、剪切增稠流体、高粘性流体、剪切变稀流体。	42
图 3-20 固体方块掉落到牛顿流体与非牛顿流体中。	42

图 3-21 曲奇饼干上的奶油。	43
图 3-22 黄油的压曲效应模拟。	44
图 3-23 果酱。	44
图 3-24 不同形态奶油的模拟。	45
图 4-1 对偶粒子方法。	48
图 4-2 一维空间中的粒子拉伸不稳定体展示。	48
图 4-3 零能问题。	49
图 4-4 SPH 法中的核函数及其一阶微分。	51
图 4-5 标准 SPH 法与本文对偶粒子方法的拉伸不稳定问题。	52
图 4-6 对偶粒子方法的计算步骤。	54
图 4-7 虚粒子的实粒子邻域缺失会导致其散度计算错误。	56
图 4-8 自由表面边界条件。	57
图 4-9 对偶粒子法研究中选用的 3 种虚粒子生成策略。	58
图 4-10 粒子微移策略的（策略 B）虚粒子生成方法。	59
图 4-11 空间自适应策略（策略 C）虚粒子生成方法。	60
图 4-12 旋转的流体方块。	61
图 4-13 策略 C 模拟的二维溃坝场景。	63
图 4-14 使用了不同虚粒子分辨率的策略 C 模拟的溃坝场景。	63
图 4-15 无粘性流体的溃坝测试。	64
图 4-16 策略 C 能够稳定模拟无粘性流体。	64
图 4-17 低粘性流体的溃坝测试。	65
图 4-18 不同的时间步长下的平均收敛次数对比。	66
图 4-19 采用 0.001s 时间步长三种策略迭代次数对比。	67
图 4-20 策略 C 与应力点 SPH 法的对比。	68
图 4-21 Chalk 等人提出的应力点 SPH 法 ^[169] 。	68
图 4-22 无黏性流体拉伸状态的对比。	69
图 4-23 使用了 XSPH 粘性的流体拉伸状态的对比。	69
图 4-24 施加了人工粘性（XSPH）与表面张力作用 ^[104] 的流体拉伸状态的对比。	70

图 4-25 两个流体在无重力空间中的碰撞。	71
图 4-26 喷泉。	71
图 4-27 大粘性不可压缩性流体。	72
图 4-28 重构了表面网格后的大粘性流体材料。	72
图 4-29 溃坝场景。	73
图 4-30 坠落的水杯。	73
图 4-31 多个流体的坠落。	74
图 4-32 流体与固体石像鬼相撞的过程。	75
图 5-1 基于对偶粒子的流体仿真引擎框架。	77
图 5-2 流体仿真引擎系统架构图。	78
图 5-3 仿真场景。	79
图 5-4 使用本文 SPH 流体的 SIMPLE 迭代算法模拟冰淇淋。	80
图 5-5 使用本文流体仿真引擎模拟的冰淇淋场景节点图。	81
图 5-6 使用本文流体仿真引擎模拟的冰淇淋场景（图 6.3 场景）时序图。	81
图 5-7 流体仿真引擎的应用示例。	82
图 5-8 基于 GPU 的粒子质量密度计算。	82
图 5-9 粒子邻域的存储方式。	83
图 5-10 粒子网格混合法。	84
图 5-11 粒子网格混合法的实现方式。	85
图 5-12 二维网格的索引。	86
图 5-13 策略 C 与粒子网格混合法的对比。	89
图 5-14 标准 SPH 投影法、本文的对偶粒子法与粒子网格混合法的管线图（Pipeline）对比。	90
图 5-15 粒子网格混合法测试场景（图 5-16）的节点图。	91
图 5-16 粒子网格混合法模拟的流体在使用粒子修正技术之前与之后的对比。	92

第1章 绪论

1.1 研究意义

流体在现实世界中很常见，比如：跌宕奔腾、气势磅礴的瀑布，盛开花朵般四溅的水花，缓缓滴落的蜂蜜，如艺术品般美丽的奶油花等。如何在计算机上快速逼真地模拟流体的运动，是计算机图形学领域的重点研究内容之一。流体仿真技术是游戏引擎、影视特效制作行业的核心技术之一^[1-6]，而且在虚拟现实、医疗^[7]、专业人员培训^[8]、科学可视化^[9]、数字孪生等应用场景中都有重要应用。近些年，随着“元宇宙”概念的快速兴起，流体仿真技术也再次成为了研究的热点^[10-12]。经过了几十年的发展，目前已有多种流体仿真方法被提出，比如欧拉网格法^[13-15]、拉格朗日粒子法^[16-19]、粒子网格混合法^[20-22]等。其中，光滑粒子流体动力学法（Smoothed Particle Hydrodynamics，缩写 SPH）是最常被使用的拉格朗日粒子法。相比于其它方法，SPH 法具有如下的优势，因此备受关注：

- 相比于欧拉网格法，SPH 法易于准确划分材料的边界。这不仅可以生动模拟流体复杂多变的运动细节，也使得 SPH 法非常适于模拟不同材料之间的耦合^[10, 23, 24]。
- 相比于欧拉网格法与粒子网格混合法，SPH 法的理论推演与编程实现较为简单直观，因此便于推广和使用，也易于与其它方法结合作为改善其它模拟方法的技术手段^[16, 25-33]。
- SPH 法作为无网格拉格朗日方法，能够有效地保证材料的质量守恒特性，理论上不会发生类似于欧拉网格法中的质量丢失问题^[34]，而且 SPH 法对流计算过程也不会引入类似于欧拉网格法与粒子网格混合法中的数值粘性^[35]。
- 相比于其它方法，SPH 法的适用性很广。它不仅可以模拟流体，也能够模拟弹塑性固体、沙子、雪、海绵、磁流体等材料，以及模拟不同材料之间的耦合、热传导、相变、溶解扩散、腐蚀等过程^[36-45]，因此使用 SPH 法易于构建多材料、多过程的统一物理仿真引擎。

尽管 SPH 流体仿真方法有上述优点，但它也存在着数值计算不稳定问题，妨碍了其应用推广。这些问题包括如下几个方面：

- **SPH 法的拉伸不稳定问题。**在 SPH 法中，随着粒子间距的减小，粒子间的吸附力会错误地增大，这一问题会导致处于拉伸状态下的流体发生粒子结块的问题，使 SPH 法难以逼真地模拟流体薄层、拉丝、溅射等小尺度的运动细节^[46]。

- **SPH 法的零能问题 (Zero-Energy Mode)**。由于 SPH 法中流体的物理场及其微分场均被定义在了同一组粒子上，这会导致 SPH 法的微分计算不准确，进一步恶化 SPH 的拉伸不稳定问题^[47]。
- **SPH 法对于粒子的分布较为敏感**。SPH 法需要使用粒子来代表流体，但流体运动的情况复杂难以预料，相应地粒子分布也不可避免地会出现不均匀、不充足的情况，这会影响 SPH 法的插值精确性，降低流体模拟的稳定性^[48]。
- **SPH 流体的粘性与不可压缩性求解步骤之间的干扰问题**。现阶段面向计算机图形学应用的 SPH 法普遍采用算子分裂法来构建流体的仿真框架，因此流体的粘性求解和不可压缩性求解需要分为两个独立的步骤进行，这会引起两求解过程的相互干扰，严重影响大粘性不可压缩流体的仿真质量^[49]。

为改善 SPH 流体仿真的稳定性，拓展 SPH 法的应用潜力，推动物理仿真技术的发展，本文着重探讨了 SPH 法数值稳定性的改进，并主要从以下两方面展开研究：一方面侧重于抑制大粘性流体中粘性、不可压缩性求解过程间的冲突，增强大粘性流体的仿真稳定性；另一方面侧重于研究抑制 SPH 法拉伸不稳定问题的方法，以改善 SPH 流体仿真的视觉真实感。

1.2 研究现状

流体具有两种重要的性质，即：不可压缩性与粘性。由于离散化方案的不同，不同技术方法对于这两种性质的计算过程与实现效果均有较大的差别。接下来本节将对近些年面向计算机图形学应用的流体快速仿真方法，以及 SPH 流体的不可压缩性、粘性、数值稳定性方面的研究成果和发展趋势进行简要的讨论。

1.2.1 流体的快速仿真方法

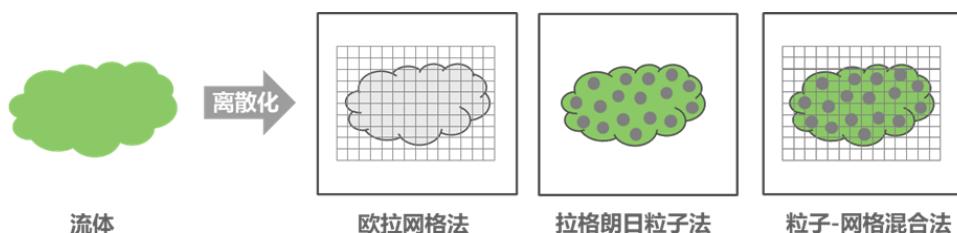


图 1-1 流体的仿真方法。

Figure 1-1 Fluid simulation methods.

现阶段，面向计算机图形学领域应用的流体快速仿真方法可划分为三大类（图 1-1），分别是：计算流体物质微团物理量变化的拉格朗日法（Lagrangian Method）^[16-19]、计算流体所占据空间中各固定点物理量变化的欧拉法（Eulerian

Method) [13-15]、将拉格朗日法与欧拉法结合的粒子-网混合格法 (Hybrid Particle-grid Method) [20-22]。

计算机图形学领域用于模拟流体运动的各类拉格朗日法中，最常被使用的方法即是 SPH 法。SPH 法是一种完全不依赖网格 (Meshless) 的仿真方法^[34]。在该方法中，流体被视为可自由运动、携带有物理量的离散粒子系统，使用该粒子系统可实现流体物理场的离散化，并可完成与流体运动相关微分方程的求解。该方法最早由 Lucy、Gingold 与 Monagan 等人提出，用于模拟星际物质的运动等天文现象^[50, 51]，后逐渐被拓展到计算流体力学领域中用于解决复杂的工程问题^[52-56]。1996 年，Desbrun 将其引入到了计算机图形学领域^[57]，用于生成游戏、影视动画、虚拟现实中的流体特效^[1, 56-66]。自此，SPH 法成为了计算机图形学领域的研究热点之一^[67-69]。

欧拉法是使用空间网格实现流体物理场离散化的一类方法^[70-72]。与 SPH 法不同，此类方法中不存在可运动的节点，只有固定的空间网格 (Grid)，各物理量以及流体对流运动等相关计算都在固定的空间网格上进行。目前，欧拉粘性不可压缩性流体的研究已有较多成果，其中代表性的成果如下：Batty 等人基于粘性流体的能量耗散函数模型，推导出隐式粘性计算公式^[73]，实现了大粘性流体材料的稳定仿真。为消除欧拉网格流体中不可压缩性与粘性的相互干扰问题，Larionov 等人建立了流体的粘性与不可压缩性的统一求解器^[49]。由于该方法不需要使用算子分裂法，可避免粘性与不可压缩性间的冲突，因此可以显著改善大粘性流体仿真的稳定性。为进一步改善这种粘性不可压缩流体统一求解方法的计算效率，Goldade 等人设计了基于八叉树空间网格的隐式粘性计算方案^[15]，降低了粘性不可压缩流体模拟过程的内存消耗。受到 Larionov 等人研究^[49]的启发，Takahashi 等人采用了类似的方法建立了求解固体运动、流体不可压缩性与粘性的统一求解器^[74]，改善了粘性流体与固体耦合场景的稳定性。相比 SPH 法，欧拉网格法的计算精确性更高，但该类方法也同时存在着难以保证质量守恒、流体边界划分不准确、对流计算易引入数值耗散等问题^[75]。

为抑制欧拉法中存在的问题，有研究者提出了在欧拉法中引入拉格朗日粒子的方法，这一方法即是粒子网格混合法^[5, 76-81]。该方法使用固定空间网格来求解运动相关的微分方程，并使用粒子来计算流体的对流运动^[82]。由于需要在粒子与网格之间进行直接的插值，标准的粒子网格混合法——PIC (Particle In Cell) 法^[83-85]存在着严重的数值耗散，因此 Brackbill 等人提出了 FLIP (Fluid Implicit Particle) 法^[86]，通过在粒子上保留一部分高频速度，来降低插值过程的数值粘性。但 FLIP 法会在粒子的运动中引入较为严重的噪声，因此 Jiang 等人提出在粒子与网格间插值仿射速度场，以保留流体的旋转和剪切运动，进而减弱数值耗散，

这一方法被称为 APIC 法 (Affine PIC) [84]。在 APIC 法工作之上, Fu 等人提出了 Poly-PIC 法^[25], 降低了插值过程中流体能量与涡量的耗散。Hu 等人结合伽辽金法与移动最小二乘法, 设计了具有高阶精确性的广义插值计算方法, 新的方法被称为 MLS-MPM 法 (Moving Least Square MPM) [20]。Fei 等人提出了新的插值策略^[35], 即: 当检测粒子之间有相互远离的趋势时, 就减少该组粒子与网格间的插值量。这一策略可显著降低粒子网格混合法的数值粘性。近些年粒子网格混合法发展得很快。然而, 这类方法依然存在着粒子分布杂乱、数值耗散大^[35]、难以保证质量守恒^[25]等问题, 有待于进一步完善。

1.2.2 SPH 法不可压缩性求解

不可压缩性是指流体密度保持恒定的特性, 该性质会强烈地影响流体的运动^[87]。因此, 实现流体不可压缩性准确、稳定、快速地求解是实现 SPH 流体高质量仿真的前提^[1, 88, 89]。标准的 SPH 不可压缩求解方法包含有两大类, 即: 基于状态方程的方法 (Equation of state-based methods) [90] 和投影法(Projection-based methods)^[91]。其中, 基于状态方程的方法根据流体局部的密度或速度散度误差来计算获取压强^[92]。而投影法则需要使用压强泊松方程来求解压强^[93]。除上述两种标准方法外, 也有 SPH 法与其它方法结合的衍生方法, 如 PBF (Position based fluid) 法^[16]等。本小节将对这些方法的研究进展进行简要的讨论。

1) 基于状态方程的不可压缩性实现方法。早期的 SPH 法通过计算线性状态方程来获得压强, 这类压强求解器允许流体发生轻微的压缩, 并根据流体压缩的程度来计算压强, 进而对流体施加不可压缩性^[94]。为进一步提高计算精确性, Markus 等人将非线性形式的压强状态方程引入到计算机图形学领域中^[95], 但由于该方法是基于显式微分方程的求解方法, 因此稳定性很差, 导致该方法对于时间步长的取值有严格的限制。为此, Solenthaler 等人提出了一种更稳定的流体压强计算方法——PCISPH (Predictive-Corrective Incompressible SPH) 法^[92]。该方法通过“预测-修正”方式迭代消除密度误差, 最终可稳定地保证流体的不可压缩性。为进一步提高 PCISPH 法这一计算策略的计算效率, He 等研究者由压强泊松方程得到了积分化的局部压强求解状态方程, 该方法可以提高“预测-修正”迭代策略的收敛速度^[96]。为了进一步增强算法的稳定性, Bender 等人在恒定密度约束之外, 又对流体施加速度无散度约束, 新的方法被称为 DFSPH 法 (Divergence-Free SPH) [97]。相比之前的方法, DFSPH 法能够获得更均匀的粒子分布, 更快的收敛速度, 并可以使用更大的时间步长用于仿真。

2) 投影法。与基于状态方程的方法不同, 投影法是通过求解压强泊松方程来实现流体不可压缩性的方法。相比于基于状态方程的方法, 投影法能够获得更为精确的计算结果^[93]。投影法的求解策略最早由 Chorin 提出^[98], Cummins 与

Rudman 在 1999 年将这种方法引入到了 SPH 法中并被广泛使用^[99]；此后，Raveendran 等人提出将 PCISPH 法与这种投影法结合，用以提高 PCISPH 法的收敛速度^[100]。为进一步提高投影法的收敛速度与稳定性，Markus 等人则提出了一种新的离散压强泊松方程的方法^[101]，并将其称为 IISPH 法（Implicit ISPH），该方法无需设置表面边界条件也可快速地求解。为了更准确地实现流体与固体之间的耦合，并改善流体模拟算法的稳定性，He 等人在投影法中引入了隐式的“错列粒子”^[102]。除微分项的离散化方案外，流体的自由表面边界条件施加方式也会对投影法的计算稳定性和收敛速度有明显的影响^[91, 103]，因此 Yang 等人借鉴了 Nair 等人的半解析式表面条件^[104-106]，并将其引入到了计算机图形学领域的研究中。这种不需要布设气态粒子的方法不仅简单易于实现，而且可以显著提高投影法求解器的收敛速度。

3) SPH 法的其它衍生方法。除去改进标准的 SPH 方法的研究之外，也有研究者引入其它领域的技术来改进 SPH 流体模拟方法。为了简化流体模拟的计算过程，Macklin 等人将 SPH 与用于仿真固体的 PBD（Position based dynamics）法^[107]结合，得到了 PBF（Position based fluid）法^[16]。PBF 法不需要求解压强场，可直接根据密度误差修正粒子位置。相比于标准 SPH 法，PBF 法虽然不是严格基于物理的方法，但计算速度快、计算过程稳定、实时性好，因此在 Realflow 等商业软件被广泛的应用^[1]。在此之后，Kang 等人进一步改进了 PBF 法^[30]，使得此类方法在保证密度恒定条件的同时，也能够满足速度场的无散度条件，改善了原 PBF 法的粒子分布。Weiler 等人提出将 SPH 法与 Projective Dynamics 法^[108]融合用于仿真无粘性流体，得到了 Projective Fluid 法^[109]。与标准的 SPH 法相比较，Projective Fluid 法有更好的稳定性。为提高仿真的精确性，Sin 等人将 Voronoi 图引入到 SPH 投影法中^[110]，这种方法将流体粒子作为节点来建立 Voronoi 图，并在 Voronoi 图上离散物理场的微分算子，显著改善了投影法的计算精度与稳定性。Goes 等人在 Sin 等人的研究基础上，使用更广义的 Power 图来离散化压强泊松方程的微分项，提高了此类仿真方法的精确性^[111]。为提高计算的效率，Zhai 等人则提出了自适应采样以及 GPU 加速的策略优化了基于 Power 图的流体仿真方法^[18]。赵建旺等人提出了将 SPH 粒子作为原 FLIP 法背景网格的策略^[112]，以在大规模流体仿真场景中，生成更多流体的运动细节。

1.2.3 SPH 法粘性求解

粘性是流体普遍具有的特性，当流体出现形变的趋势时（形变速率），流体内部就会出现剪切应力来抵抗流体的形变。由于该特性强烈地影响着流体的行为，因此粘性的求解方法是 SPH 法的研究重点之一。

1) SPH 法中的粘性求解方法研究。SPH 法中有较多种类的粘性求解方法，

其中, XSPH 法是一种被广泛使用的人工粘性求解方法。该方法最早由 Monaghan 提出^[113], 它通过对粒子速度场进行局域加权平均计算, 使流体获得了类似于粘性的运动特征。该方案得到的“粘性”虽然与真实流体的粘性相差较大, 也无法模拟大粘性流体, 但它在模拟低粘性流体时, 可以平滑掉流体粒子的高频速度, 改善流体运动的稳定性, 因此被广泛使用^[16, 114]。后有研究者基于流体粘性的动力学方程, 使用 SPH 粒子离散方程中的高阶微分项, 建立了粘性的显式方程计算方法^[95, 115-118]。此类粘性求解器虽然有较高的仿真精度, 能够模拟粘性流体的诸多运动特征, 但此类方法的稳定性很差, 当时间步长和粘性系数同时取较大值时仿真会发生崩溃。为了解决此问题, 有研究者提出了使用隐式粘性约束方程替代原显式方程的方案^[119]。这种基于隐式粘性方程的方法稳定性更好, 不仅能够准确模拟粘性流体卷绳效应与压曲效应^[120, 121], 而且在时间步长与粘性系数同时取较大值时, 该方法依然能够稳定地实现仿真。为了改善粘性流体角动量守恒特性, Bender 等人提出了通过消除特定比例的速度梯度来获得流体粘性的方法^[90]。由于该方法计算过程中角动量损失较小, 因此可以较好地模拟粘性流体卷绳效应与压曲效应。然而该方法不存在具有物理意义的粘性系数, 这会在使用中造成不便。目前计算机图形学领域中最先进的隐式粘性算法是 Weiler 等人提出的, 该方法简化并改进了流体动量方程中粘性项的离散化方案^[122], 使得 SPH 的粘性计算不仅易于实现, 而且能够得到更真实、更精确的仿真结果。

2) 粘性求解与不可压缩求解之间的耦合。在大多数不可压缩性流体的仿真方法中, 粘性求解器与不可压缩求解器间普遍使用算子分裂法实现耦合^[73, 96, 116, 119]。然而, 当仿真时间步长和流体的粘性系数取较大值时, 两种求解器之间则会产生较严重的冲突, 影响了仿真的稳定性。针对此问题, Peer 等人提出通过重建流体速度场方法来实现流体不可压缩性与粘性的同时求解, 可有效避免两者间的冲突^[119, 123]。但由于标准的 SPH 法很难使用速度微分量精确重建速度场^[11, 124], 导致此类粘性约束施加过程中不可避免地引入了严重的数值耗散, 反而降低了大粘性流体仿真的稳定性。在欧拉网格法流体模拟的研究中, Larionov 提出了粘性不可压缩性的统一求解方法, 该方法在同一个线性系统中求解粘性和不可压缩性的作用^[49], 由于避免了不可压缩性与粘性之间的冲突, 因此可以模拟具有更丰富表面细节的大粘性流体。目前, 这种使用统一求解器避免求解步骤间冲突的策略, 在欧拉网格法中已有较多的研究^[74, 125-127]。但受限于 SPH 法中粒子邻域关系的不确定性, 这些方法很难引入到 SPH 法流体中。

1.2.4 SPH 流体的数值计算稳定性研究

SPH 流体仿真方法中存在着较多的数值不稳定问题^[48, 124], 会导致流体仿真过程出现错误, 影响仿真的视觉真实感。其中较为严重的问题分别是 SPH 法的

拉伸不稳定问题以及 SPH 法不易实现一阶精度的问题。

拉伸不稳定是 SPH 法中较难处理的一类数值计算问题^[16]。当被仿真的材料处于拉伸状态下，构成材料的粒子会出现结块、断裂等问题，导致材料的形态无法维持并错误地碎裂。在计算机图形学领域的研究中，常通过消去负压强或设置正的背景压强来避免粒子间出现吸附力，进而降低拉伸不稳定问题的影响^[73, 85, 140]。但这会破坏流体很多小尺度的运动细节，导致其很难较为稳定、逼真地模拟流体溅射、拉丝和薄层等形态。

标准的 SPH 法插值计算精度较低，导致流体的运动不稳定，视觉真实感不高。这一问题产生的根本原因是 SPH 核函数归一化条件很容易被破坏：当流体的粒子分布不均匀时，SPH 核函数就不再满足归一化条件；另外由于 SPH 法在边界附近不具有克罗内克函数（Kronecker delta）属性^[124]，因此边界附近的粒子邻域缺失问题也会破坏归一化条件。在计算机图形学领域中，这一类数值不稳定问题的研究可被划分为流体边界处理方法以及保证插值精确性的方法两方面，以下是这两方面问题的讨论。

1) 边界处理方法。由于 SPH 法的粒子邻域的缺失问题会导致自由表面附近的流体粒子密度计算值过低。针对这一问题，Schechter 提出了在流体表面附近均匀布设气态虚粒子的方法^[128]；Si 与袁志勇等人提出根据流体表面粒子的邻域平均数量和内部粒子邻域平均数量之比修正流体密度计算^[129]，以及使用水平集（level set）函数修正表面附近的粒子密度^[130, 131]的方法。这些方法能够改善因粒子邻域缺失对稳定性造成的影响。在固壁边界附近，邻域缺失问题会表现得更为明显，会导致固壁边界附近粒子分布非常的不均匀。避免该问题的常规方法是在固体内部设置固态幽灵粒子（Ghost-solid particle）弥补边界附近流体粒子的邻域缺失问题^[102, 122, 132-135]，但这一方法需要耗费大量的计算时间和内存。半解析式流固边界方法可以有效避免上述的问题^[17, 136, 137]。该类方法使用面片（mesh）或有向距离场来表示固体，粒子会根据其与固体边界的距离及边界形态修正计算，实现对邻域缺失问题的抑制。

2) 保证插值精确性。保证 SPH 流体插值精确性的研究可以分为两类，一类是小幅度修正粒子位置，迫使粒子分布均匀的方法^[91, 138]；另一类是引入更先进的插值计算方法，比如：通过构建隐式方程的方式来修正 SPH 核函数的权重，保证 SPH 核函数的归一化条件^[139]，以及引入移动最小二乘法保证插值精确性^[140]的技术。这些方法都能够提高 SPH 法的仿真稳定性。

1.3 研究内容

本文工作主要围绕着改善 SPH 流体仿真的数值稳定性方面展开，并重点研究了抑制 SPH 法中流体粘性与不可压缩性间冲突，改善流体在拉伸状态下的稳

定性的方法。具体的研究工作如下：

(一) 改善 SPH 粘性不可压缩流体仿真稳定性的研究：(1) 本文提出了适用于粘性不可压缩 SPH 流体的 SIMPLE 迭代算法。SPH 法在模拟流体时，一般会基于算子分裂法来分别对流体施加粘性与不可压缩性的作用。然而，这种模拟策略会导致流体的粘性计算与不可压缩性计算之间产生冲突，在求解过程中发生相互的干扰，严重影响大粘性流体的仿真稳定性。因此，本文引入并改进了 SIMPLE (Semi-Implicit Method For Pressure Linked Equation) 迭代算法，通过对流体不可压缩性与粘性进行总体迭代的方式，逐渐减轻两者之间的干扰。对比测试证明本文方法能够稳定地收敛，而且只需要若干次迭代即可显著提高大粘性流体仿真的稳定性，使流体能够产生更锐利、更丰富的表面细节，更稳定地保持形态。(2) 本文提出了基于 Cross 模型的非牛顿流体高效仿真方法。由于非牛顿流体的粘性会发生大幅度的变化，容易导致其仿真的不稳定。本文则结合 Cross 模型，改进了 SIMPLE 算法中的隐式粘性仿真方法，实现了包含剪切变稀流体、剪切增稠流体在大时间步长下的稳定仿真。

(二) SPH 流体拉伸不稳定问题的研究：(1) 在 SPH 法中引入对偶粒子框架。SPH 流体仿真方法自被发明以来，就一直受到拉伸不稳定问题的困扰。这一数值计算不稳定问题常会导致流体粒子产生积块，使流体难以稳定地维持拉伸的形态。抑制该问题常需要设置背景压强、人工斥力等非物理的手段，但这些方法依然难以稳定地模拟流体的薄层、拉丝、飞溅等小尺度的细节。为抑制拉伸不稳定问题，提升流体小尺度细节模拟的稳定性，本文在 SPH 法中引入了对偶粒子的框架，将标准的 SPH 粒子划分为具有速度属性的实粒子和具有压强属性的虚粒子。由于额外引入的虚粒子可以减小 SPH 法压强梯度计算的拉伸不稳定区域、避免零能问题的产生，因此可以有效地提高流体的拉伸稳定性，进而提高 SPH 法流体小尺度细节的仿真稳定性。(2) 虚粒子采样策略对于对偶粒子法稳定性影响的研究。在进行对偶粒子方法的研究中发现，虚粒子的采样方式会对流体行为产生非常明显的影响，而使用类似于欧拉网格的虚粒子分布方式能够显著地提高流体的稳定性。因此，本文提出了基于 GPU 的空间自适应虚粒子快速生成方法，以高效高质量地实现稳定的流体仿真。

(三) 提出基于对偶粒子法的通用粒子法流体仿真引擎。本文提出的对偶粒子仿真框架中包含的虚粒子，不仅可作为拉格朗日视角的粒子，也可作为欧拉视角的空间网格。因此本文提出的对偶粒子框架能够兼容标准 SPH 法、对偶粒子法、粒子网格混合法等多种粒子仿真方法。基于此，本文构建了一个通用的流体仿真引擎。在使用该引擎进行的对比研究中发现，粒子网格混合法中粒子分布非常不均匀。因此，本文提出使用 SPH 粒子位置修正技术 (Particle Shifting) 以均匀地分布粒子网格混合法中的粒子，改善了此类方法的稳定性。

1.4 本文组织结构

本文的组织结构如下：

第一章是本论文的绪论部分。该部分讨论了本文课题的研究背景与研究意义、该领域目前的研究内容与成果，并简述了本课题的主要研究工作。

第二章是相关基础理论的讨论与介绍。具体包括连续介质力学与流体力学的部分基础理论、使用 SPH 法实现流体模拟的基本策略、SPH 法中的离散化方法三部分。

第三章讨论了抑制 SPH 流体中粘性求解与不可压缩性求解之间冲突的研究工作。本章首次在 SPH 法体系下构建了 SIMPLE 迭代算法，并测试了这一算法的收敛性，在对比测试中证明该方法可显著提高大粘性流体的仿真稳定性。此外，本章的研究也进一步改进了隐式粘性求解算法，提出了基于 Cross 模型的非牛顿流体高效仿真方法。

第四章提出了抑制 SPH 流体拉伸不稳定性方法。本章讨论了 SPH 流体中拉伸不稳定产生的原因，首次提出了可抑制拉伸不稳定的对偶粒子框架，基于对偶粒子框架构建了投影法求解器，并验证了新方法在抑制拉伸不稳定问题、改善流体仿真质量方面的有效性。此外，本章设计了能够在 GPU 上快速生成均匀分布虚粒子的方法，并在算法的验证阶段发现虚粒子分布的均匀性会对流体仿真的稳定性有较大的影响。

第五章基于对偶粒子法提出了一种通用的流体仿真引擎。该引擎可兼容标准 SPH 法、对偶粒子法与粒子网格混合法等各类粒子仿真方法。文章介绍了引擎的实现架构、仿真场景的搭建方式与仿真管线的设计，并探讨了 PIC、FLIP、NFLIP 等粒子网格混合法求解器的实现方式。最后，本章在完全相同的外部条件下对比研究了对偶粒子与粒子网格混合法的区别，对比发现粒子网格混合法中粒子的分布非常不均匀。因此，本文提出了使用 SPH 粒子位置修正技术以均匀地分布粒子网格混合法中的粒子，改善了此类方法的仿真稳定性。

第六章对本文的研究成果进行了总结，并讨论了未来的研究方向与计划。

第 2 章 粘性不可压缩流体与 SPH 法

2.1 引言

流体材料是由分子等微观粒子构成的，但受限于从微观粒子尺度上分析宏观流体运动的复杂性，通常会将宏观的物质材料视为连续的介质而非大量离散的微观粒子对其进行研究，这一假设即为“连续介质假设”，基于该假设所建立的力学理论体系也被称为连续介质力学 (Continuum Mechanics)^[87]。连续介质假设忽略了材料的微观结构，认为材料是由连续、均匀、无间隙的质点构成的，这些质点可被认为是在宏观上无限小，但在微观上远大于量子尺度的物质微团，其携带有诸如密度、速度、应力等宏观物理量，且满足经典牛顿力学体系中的物理定律，例如质量守恒定律、动量守恒定律、能量守恒定律等。借助于连续介质假设，也便于描述材料宏观视角下的应力与应变函数对应关系，即本构方程。结合材料的本构方程和经典力学的基本定律，可对材料的运动进行分析、计算和预测。

然而，连续介质假设中的流体本构方程是很难直接求解的，一般需要将其中的物理场与微分算子进行离散化，使连续的微分方程转化为代数方程组之后才能够求解。SPH 法中包含有一套简单、直观且有效的离散化方法，可以将流体的动量方程、连续性方程转化为离散形式，进而计算流体的不可压缩性与粘性作用下的运动，最终实现流体运动的模拟。

本章将引入流体力学的相关理论，对粘性不可压缩流体的物理学模型及其相关特性进行简要介绍和讨论。随后，本章将介绍使用 SPH 法模拟流体的基本流程及其离散化方案。

2.2 物理场、散度与梯度

在连续介质力学中，场一般指某物理量 f 与空间位置 \mathbf{x} 的对应关系，即： $f(\mathbf{x})$ ，常用于表征某物理量的空间分布。由于“连续介质假设”认为材料是由连续的质点构成，因此材料的物理场取值在一定空间内通常是一阶连续的，相应地该物理场也就存在其一阶微分。本文常用的两种一阶微分算子是梯度和散度，以下是这两种微分计算的介绍。

1) 梯度的定义。在处理能量优化等实际问题时，常需要找到某个场取值变化最大的方向，这一方向可通过计算该场的梯度来获取（如图 2-1(a) 所示）。对于三维空间中某个标量场 $f(\mathbf{x})$ ，其梯度定义式为：

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \quad (2-1)$$

其中 ∇ 为梯度算子，上式计算获得的矢量即是 f 场变化最大的方向；如果对三维空间中的某矢量场 $\mathbf{f}(\mathbf{x}) = (f, g, h)$ 求梯度，则可得到一个 2 阶张量：

$$\nabla \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x}, & \frac{\partial g}{\partial x}, & \frac{\partial h}{\partial x} \\ \frac{\partial f}{\partial y}, & \frac{\partial g}{\partial y}, & \frac{\partial h}{\partial y} \\ \frac{\partial f}{\partial z}, & \frac{\partial g}{\partial z}, & \frac{\partial h}{\partial z} \end{pmatrix} \quad (2-2)$$

2) 散度的定义。进行物理模拟的相关计算时，常需要评估某个矢量场的发散程度（如图 2-1(b)所示），这时就需要引入散度的概念。如果在三维空间中有一矢量场为 $\mathbf{f}(\mathbf{x}) = (f, g, h)$ ，场 \mathbf{f} 的散度定义式如下：

$$\nabla \cdot \mathbf{f}(\mathbf{x}) = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} \quad (2-3)$$

由该定义式可知，散度计算的是某个空间位置点邻近物理量内聚或发散的趋势，如果 \mathbf{o} 点邻近区域内场的方向均指向 \mathbf{o} 点，则散度 $\nabla_{\mathbf{o}} \cdot \mathbf{f}$ 的取值为负；若 \mathbf{o} 点邻近区域内的场的方向均远离点 \mathbf{o} ，则散度 $\nabla_{\mathbf{o}} \cdot \mathbf{f}$ 的取值为正；若散度 $\nabla_{\mathbf{o}} \cdot \mathbf{f}$ 的取值为零，则表明在点 \mathbf{o} 的邻近区域内，场的向内和向外趋势互相抵消。

3) 散度与梯度的关系。结合场的散度定义与上文中场的梯度定义，可得到任意场的散度算子与梯度算子的对应关系，即：

$$\text{tr}(\nabla \mathbf{f}(\mathbf{x})) = \nabla \cdot \mathbf{f}(\mathbf{x}) \quad (2-4)$$

其中 $\text{tr}(\nabla \mathbf{f}(\mathbf{x}))$ 为矩阵 $\nabla \mathbf{f}(\mathbf{x})$ 的迹。

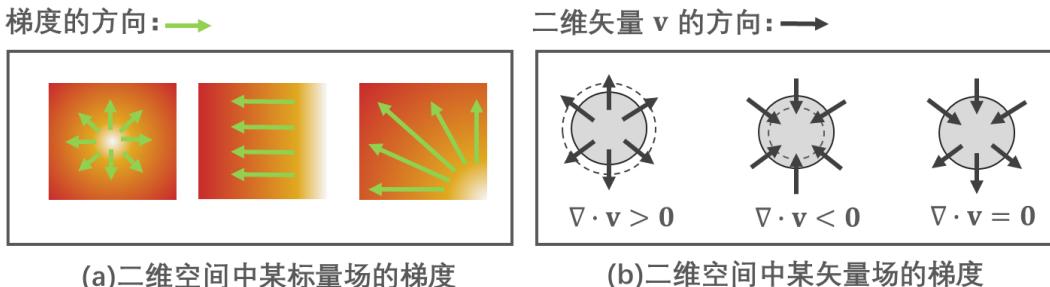


图 2-1 梯度与散度。
Figure 2-1 Gradient and divergence.

2.3 流体运动的控制方程、不可压缩性与粘性

基于上小节介绍的连续性假设与物理场的定义，可以建立流体运动的控制方程。本节将对流体力学的部分基础理论——流体的连续性方程、动量方程、不可压缩性条件三方面进行简要的讨论。

2.3.1 连续性方程

在连续介质力学中，流体被认为是由连续的质点构成的，因此描述流体各特性的方程可基于连续物理场及其空间微分来构建。基于连续介质假设可以推导得

到流体最重要的特性之一：质量守恒定律。以下是这一定律的讨论。

以三维空间为例，假设在材料所占据的空间中存在一个较小的封闭空间 Ω ，该封闭空间 Ω 中的材料质量变化速度可通过计算其内部密度变化率的积分获得：

$$\delta m = \iiint_{\Omega} \frac{\partial \rho(\mathbf{x}, t)}{\partial t} dV \quad (2-5)$$

在经典牛顿力学理论中，质量是不会凭空增大或减小的，因此导致封闭空间 Ω 内质量发生变化的原因只能是物质流入或流出封闭空间 Ω 造成的。基于此，可以得到如下等式。

$$\iiint_{\Omega} \frac{\partial \rho(\mathbf{x}, t)}{\partial t} dV = -\oint_{\partial\Omega} \rho \mathbf{u} \cdot \mathbf{n} dS \quad (2-6)$$

其中的 \mathbf{n} 为曲面 dS 的法线方向， \mathbf{u} 为材料的速度场， ρ 为材料的密度场。由散度定理（ $\oint_{\partial\Omega} \rho \mathbf{u} \cdot \mathbf{n} dS = \iiint_{\Omega} \nabla \cdot (\rho \mathbf{u}) dV$ ），上式右端项可进一步改写为速度场的散度形式：

$$\iiint_{\Omega} \frac{\partial \rho(\mathbf{x}, t)}{\partial t} dV = -\iiint_{\Omega} \nabla \cdot (\rho \mathbf{u}) dV \quad (2-7)$$

其中， Ω 可以为任意体积、形状的封闭空间，因此可以假设封闭空间 Ω 无穷小，小到只包含材料的一个质点，此时上述积分式转化为一个偏微分方程，这个方程是连续介质力学中最重要的公式之一——连续性方程，其形式如下。

$$\frac{d\rho}{dt} = \nabla \cdot (\rho \mathbf{u}) \quad (2-8)$$

2.3.2 动量方程

纳维斯托克斯方程（Navier Stokes Equation），简称 NS 方程，是描述流体运动的控制方程，其形式为：

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla \cdot \tau + \mathbf{f} \quad (2-9)$$

其中， \mathbf{u} 是流体的速度场，本文中 \mathbf{u} 在三维空间直角坐标系下 x、y、z 轴方向的分量分别为 u 、 v 、 w ； p 是流体的压强场； \mathbf{f} 表示流体所受到的重力、离心力等体力场。该等式的左端项为速度场对时间的随体导数，在拉格朗日视角下该随体导数可直接写成： $\frac{D\mathbf{u}}{Dt} = \frac{d\mathbf{u}}{dt}$ 。等式右端项中， $-\nabla p$ 项是流体质点间的压强力场，正是该项的存在保证了流体的不可压缩性；而 $\nabla \cdot \tau$ 为流体的粘性作用，其中 τ 为流体内部产生的剪切应力，该剪切应力一般是以流体速度为变量的场函数，其形式如下。

$$\tau = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad (2-10)$$

如果忽略掉流体的粘性和体力的作用，上述 NS 方程即可写为如下形式：

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho} \nabla p \quad (2-11)$$

由该方程可知，若某质点附近压强场的梯度不为零，会导致该点的速度发生变化，且速度变化的方向为压强梯度的反方向。此处可将压强的梯度视为流体某质点受到周围质点压力的合力。

2.3.3 不可压缩性条件

仅使用 NS 方程，并不能根据上一时刻的流体状态求解出当前时刻流体的速度场。因为 NS 方程中除当前时刻的速度场未知以外，其中的压强场也同为未知量。为了保证 NS 方程具有唯一可行解，需要引入额外的条件。对于不可压缩性流体，这一条件即为不可压缩性约束。

真实的流体往往很难被压缩，即使施加很大的外部压力，流体的密度也只会有微小的变化。若环境温度等条件不变，流体的密度可近似认为是常数，即流体静止时的密度 ρ_0 。相应地，流体的不可压缩性可使用 $\rho(\mathbf{x}, \mathbf{t}) = \rho_0$ 这一方程来表示。然而这一公式并不能与 NS 方程联立，因此需要引入连续性方程，将有关密度的约束条件转化为速度场的约束，即：

$$\nabla \cdot \rho \mathbf{u} = \frac{\partial \rho}{\partial t} \quad (2-12)$$

将 $\rho(\mathbf{x}, \mathbf{t}) = \rho_0$ 带入后可得：

$$\rho_0 \nabla \cdot \mathbf{u} = \frac{\partial \rho_0}{\partial t} = 0 \quad (2-13)$$

整理后，即可得到十分简洁的速度场微分方程，即速度无散度条件：

$$\nabla \cdot \mathbf{u} = 0 \quad (2-14)$$

将速度无散度条件与忽略掉粘性与体积力作用的 NS 方程联立并离散，并正确地施加边界条件与初始条件，理论上联立的方程组（下式）即可获得不可压缩性作用下的流体速度场，最终实现在不可压缩性作用下流体运动的模拟。

$$\begin{cases} \frac{d\mathbf{u}}{dt} = -\frac{1}{\rho_0} \nabla p \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2-15)$$

2.4 流体的粘性

粘性是流体普遍具有的性质（除超流体外）。本节将对流体粘性的主要性质进行简要的介绍和讨论。

当流体受到剪切力（一对大小相同、方向相反且平行的作用力）持续作用时，流体会发生持续的剪切形变，此时流体内部会产生剪切应力，该应力会抵抗流体发生剪切形变，最终使流体产生粘性的运动特征。

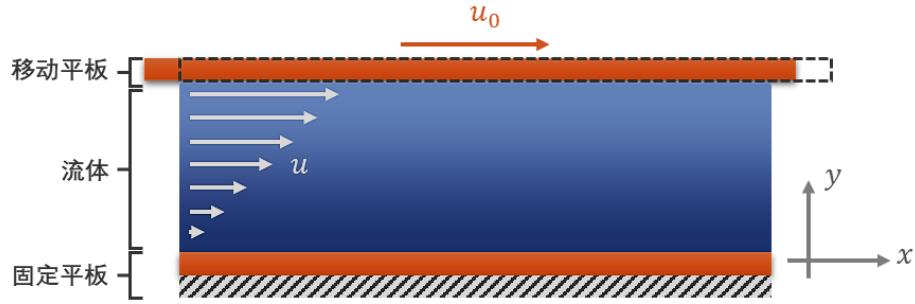


图 2-2 牛顿剪切流动试验。

Figure 2-2 Newton shear flow experiment.

从生活经验中可知，不同种类的流体其粘性作用的强弱并不相同。比如，蜂蜜的粘性远大于水，而沥青的粘性远远大于蜂蜜。若要定量表征不同流体粘性的强弱，就要引入流体的粘性系数的概念。流体粘性可以使用图 2-2 所示试验测定^[87]，这一实验中包含有两个固体平板，其中一个固定，而另一个可以在平板所在的平面上移动。在两平板之间填充充满某种粘性流体后，以速度 u_0 移动可活动的平板，两平板内的流体就会在流体内剪切应力 τ_{xy} 的作用下产生垂直于平板方向的非 0 速度场分布，且速度场分布与剪切应力之间的关系满足以下公式：

$$\tau_{xy} = \mu \frac{\partial u}{\partial y} \quad (2-16)$$

上式中的 u 是平行于平板方向的流体速度场分量， y 是垂直于平板的坐标轴方向， μ 是流体的粘性系数或动力粘度，其单位是 $\text{Pa} \cdot \text{s}$ 。

在上述测定流体粘性的实验中，速度场的空间微分只在 y 轴方向是非零的，因此由上式获得的剪切应力 τ_{xy} ，只是 NS 方程中的剪切应力 $\boldsymbol{\tau}$ 二阶张量中的一个分量。对于牛顿流体，流体的剪切应力一般与形变速率（Rate of Deformation）正比，因此剪切应力的定义式完整形式应为：

$$\boldsymbol{\tau} = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \quad (2-17)$$

其中， $\nabla \mathbf{u}$ 为速度场的梯度，将其对称化后的形式为： $\nabla \mathbf{u} + (\nabla \mathbf{u})^T$ ，对称化后的张量也被称为形变速率张量：

$$\nabla \mathbf{u} + (\nabla \mathbf{u})^T = \begin{pmatrix} 2 \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & 2 \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} & \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} & 2 \frac{\partial w}{\partial z} \end{pmatrix} \quad (2-18)$$

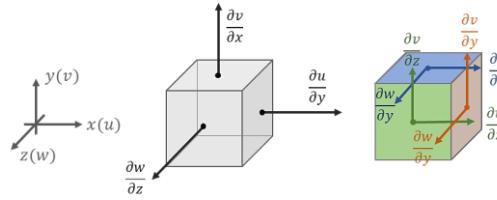


图 2-3 形变速率张量中各元素的物理意义。

Figure 2-3 The elements of the rate of deformation tensor.

其中，形变速率张量中对角元素的物理含义是流体的体积变化速率，而非对角元素则为剪切应变速率（图 2-3）。将流体的剪切应力带入到流体的 NS 方程后可得：

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla \cdot (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \mathbf{f} \quad (2-19)$$

将表征流体不可压缩性的速度无散度条件带入上式 ($\nabla \cdot (\nabla \mathbf{u})^T = \nabla(\nabla \cdot \mathbf{u}) = 0$)，上式即可进一步化简为如下形式。

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla \cdot \nabla \mathbf{u} + \mathbf{f} \quad (2-20)$$

粘性系数不发生变化的流体被称为牛顿流体。然而在现实世界中，有非常多的流体其粘性系数不为定值，这类流体被称为非牛顿流体。具有复杂分子结构的大粘性流体材料多数为非牛顿流体，比如淀粉糊、人体的血液和组织液、高分子聚合物溶液等。其中，粘性随形变速率的增大而减小的流体被称为剪切变稀流体；与之相反，粘性随形变速率的增大而增大的流体被称为剪切增稠流体^[141]。

2.5 流体模拟方法概述

基于上文关于流体的部分基础理论，本节将简要讨论流体的基本仿真策略，并介绍使用 SPH 法实现流体模拟的主要流程。

2.5.1 算子分裂法

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla \cdot \nabla \mathbf{u} + \mathbf{f} \quad (2-21)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2-22)$$

粘性不可压缩流体运动的控制方程是 NS 方程和无散度条件（上式）。虽然流体的运动令人眼花缭乱，但只要能够精确地求解出这两个联立的方程，理论上就能够精准地预测流体的运动。但求解这两个方程并不容易，需要解决两个问题：首先，这两个方程需要将其在时间上和空间上离散后才可以求解，而空间离散化的方法可以基于固定的网格（如欧拉网格法），或是使用移动的粒子（如 SPH 法）来实现。此外，带有多个微分算子的方程即使完成离散后仍过于复杂，这就需要

引入算子分裂法，简化流体模拟的相关计算。具体实现方法是将 NS 方程拆分成若干个独立且较小规模的偏微分方程，每个较小的偏微分方程相比于原 NS 方程要简化很多，且可以采用不同的方法分别求解若干个较小方程，显著降低求解的时间、空间复杂性。

在使用拉格朗日视角下的 SPH 法，一般会将 NS 方程拆分如下几个小方程（粘性部分以显式方法为例）进行分别求解。

$$\frac{\mathbf{u}^{n+\frac{1}{3}} - \mathbf{u}^n}{\delta t} = \mathbf{f} \quad (2-23)$$

$$\frac{\mathbf{u}^{n+\frac{2}{3}} - \mathbf{u}^{n+\frac{1}{3}}}{\delta t} = \frac{\mu}{\rho} \nabla \cdot \nabla \mathbf{u}^{n+\frac{1}{3}} \quad (2-24)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+\frac{2}{3}}}{\delta t} = -\frac{1}{\rho} \nabla p \quad (2-25)$$

上述三个方程由上至下分别描述了体积力、粘性、不可压缩性对流体速度的作用。需要注意的是上述方程中不包含求解压强的方程，因此求解式(2-25)之前需要根据流体的密度误差或速度散度误差计算出压强。

2.5.2 SPH 流体模拟流程

通过算子分裂法将流体的 NS 方程拆分为若干个小方程后，即可使用 SPH 对其进行离散并求解。使用 SPH 法进行一帧流体仿真计算的具体实现步骤以及每个步骤的讨论如下：

步骤 1. 粒子的邻域查找。SPH 法是一种无网格粒子法，粒子之间并不存在确定的拓扑关系。在使用 SPH 法模拟流体时，代表流体质点的粒子会发生难以预测的运动，因此粒子之间的相邻关系也同样是难以确定的。然而，在进行流体微分场的相关计算时，必须要使用到每个粒子的邻近粒子，因此在 SPH 法进行每一帧仿真计算之前，均需要对问题域内的全部粒子进行邻域查找，并将每个粒子的邻域存储下来，以用于后续的粘性、不可压缩性等方面的计算^[67]。

步骤 2. 体积力作用下的粒子速度求解。基于公式(2-23)，更新每个粒子在体积力（一般为重力）下的速度。

步骤 3. 粘性作用下的粒子速度求解。基于公式(2-24)，对速度场施加粘性约束。目前粘性求解方法主要分为两类，分别是显式和隐式粘性方法。其中隐式粘性需要求解线性系统以获得更好的稳定性^[123]，而显式粘性的实现方法较为简单^[116]，但稳定性很差^[122]。

步骤 4. 求解压强并计算出压强力，更新速度场。标准的不可压缩性求解过程包含两步：第一步是根据流体的密度误差或速度散度误差来计算压强，第二步则是计算压强梯度来获得压强力（Pressure force），进而更新速度场式(2-25)^[68]。

步骤 5. 使用速度场更新粒子的位置。使用当前的粒子速度来更新粒子的位置，计算公式如下^[68]:

$$\mathbf{x}_i^{new} = \mathbf{x}_i^{old} + \mathbf{u}_i^{new} * \delta t \quad (2-26)$$

步骤 6. 计算流体的表面张力作用。当模拟小尺度的流体时，流体的表面张力作用会表现得较为明显。而 NS 方程与无散度条件下并不包含流体表面张力的作用，因此需要引入表面张力求解器来施加表面张力的作用，具体可参考 He 等人提出的方法^[104]。

步骤 7. 输出粒子位置，重构流体表面网格，完成渲染。后处理一般包含有两个步骤，即：使用粒子数据重建流体表面；完成流体的渲染。如果采用粒子离线渲染的方式，后处理则仅为粒子数据输出一个步骤。待完成仿真任务后，统一对每一帧的粒子数据进行表面重建和渲染^[67]。

2.6 SPH 法的离散化方案

流体的模拟计算需要将流体动量方程与连续性方程离散化，然后求解离散化的代数方程。本节将介绍 SPH 法核心理论——基于核函数的离散化方法，为后续 SPH 粘性不可压缩流体的研究提供理论基础。

2.6.1 光滑核函数

SPH 法中的离散化方案是一种基于光滑核函数实现物理场插值计算的方法。为了更好地说明 SPH 法离散化的实现过程，首先需要引入狄拉克函数 $\delta(\mathbf{r})$ ，其定义如下^[68]:

$$\delta(\mathbf{r}) = \begin{cases} \infty & |\mathbf{r}| = 0 \\ 0 & |\mathbf{r}| \neq 0 \end{cases} \quad (2-27)$$

狄拉克函数满足如下的性质：

$$\int \delta(\mathbf{r}) dV = 1 \quad (2-28)$$

该式的含义是：狄拉克函数在全空间域内的积分值为 1。此处使用狄拉克函数 $\delta(\mathbf{r})$ 作为基函数，那么某个空间连续的场函数 $A(\mathbf{x})$ 对狄拉克函数 $\delta(\mathbf{r})$ 求卷积将等于场函数 $A(\mathbf{x})$ 的自身，即：

$$A(\mathbf{x}) = (A * \delta)(\mathbf{x}) = \int A(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') dV' \quad (2-29)$$

上式中的 dV' 是与位置变量 \mathbf{x}' 有关的体积^[34]。

上述公式中的狄拉克函数是不连续不可微的函数，这导致上述卷积公式会在使用中受到很多限制。为了对上述公式中的狄拉克函数进一步推广，首先可引入平滑核函数 $W: \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}$ 来替换上述卷积公式中的狄拉克函数，最终场函数 $A(\mathbf{x})$ 可近似为其自身与平滑核函数 W 的卷积^[68]，即：

$$A(\mathbf{x}) \approx (A * W)(\mathbf{x}) = \int A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) dV' \quad (2-30)$$

其中标量 h 为紧支域半径或平滑长度。平滑核函数 $W(\mathbf{r}, h)$ 须满足如下五个条件^[52]:

$$\int W(\mathbf{r}, h) dV = 1 \quad (\text{归一化条件}) \quad (2-31)$$

$$W(\mathbf{r}, h) \geq 0 \quad (\text{正则条件}) \quad (2-32)$$

$$\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r}) \quad (\text{狄拉克条件}) \quad (2-33)$$

$$W(\mathbf{r}, h) = W(-\mathbf{r}, h) \quad (\text{对称条件}) \quad (2-34)$$

$$W(\mathbf{r}, h) = 0, \text{ 若 } |\mathbf{r}| > h \quad (\text{紧支条件}) \quad (2-35)$$

上述五项条件的含义分别是：①函数 $W(\mathbf{r}, h)$ 对 \mathbf{r} 的全域积分须为 1；②函数 W 不能为负；③当支持域趋近于 0 时，核函数等价于狄拉克函数；④函数 $W(\mathbf{r}, h)$ 为正对称函数；⑤当变量 \mathbf{r} 的模（欧式距离）大于紧支域时，函数 $W(\mathbf{r}, h)$ 的值为 0。能够同时满足这五项条件，且具有高阶可微性的理想函数是高斯函数形式的核函数。然而高斯函数中包含有指数运算，会导致计算过于耗时。在实际使用中，经常会选取更易于计算的样条函数，其中最常被使用的一种是三次样条分段函数^[114]，即：

$$W(\mathbf{r}, h) = \alpha^d \begin{cases} \frac{2}{3} - \left(\frac{\mathbf{r}}{h}\right)^2 + \frac{1}{2} \left(\frac{\mathbf{r}}{h}\right)^3 & 0 < \frac{\mathbf{r}}{h} \leq 1 \\ \frac{1}{6} \left(2 - \frac{\mathbf{r}}{h}\right)^2 & 1 < \frac{\mathbf{r}}{h} \leq 2 \\ 0 & \frac{\mathbf{r}}{h} \geq 2 \end{cases}, \quad (2-36)$$

在一维、二维、三维空间中，上式中 α^d 的值分别为： $1/h$ 、 $15/(7\pi h^2)$ 、 $3/(2\pi h^3)$ 。

考虑到流体的质量、密度、体积存在关系： $dV = \frac{dm}{\rho(\mathbf{x})}$ ，则式(2-30)可以转换为如下形式：

$$\begin{aligned} A(\mathbf{x}) \approx (A * W)(\mathbf{x}) &= \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')} W(\mathbf{x} - \mathbf{x}', h) \rho(\mathbf{x}') dV' \\ &= \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')} W(\mathbf{x} - \mathbf{x}', h) dm' , \end{aligned} \quad (2-37)$$

即：物理场 A 与平滑核函数 W 的卷积，可以由体积积分转换为质量积分。如果使用粒子来表示连续的物理场 A ，则每个粒子 j 都携带有物理量 A_j 、密度值 ρ_j 和质量 m_j 。假设在式(2-37)中 \mathbf{x} 的位置上有一粒子 i ，则原积分式(2-37)可转换为求和公式：

$$\begin{aligned} A(\mathbf{x}_i) \approx (A * W)(\mathbf{x}_i) &= \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')} W(|\mathbf{x}_i - \mathbf{x}'|, h) \rho(\mathbf{x}') dV' \\ &\approx \sum_j \frac{m_j}{\rho_j} A_j W(|\mathbf{x}_i - \mathbf{x}'|, h) \end{aligned} \quad (2-38)$$

这个基于离散粒子来重建点 i 附近位置物理场 A 的方法，也就是 SPH 中基于光滑核函数的插值计算方法。为了便于书写，上式可进一步简化为如下形式：

$$A(\mathbf{x}_i) \approx \langle A_i \rangle = \langle A(\mathbf{x}_i) \rangle = \sum_j \frac{m_j}{\rho_j} A_j W(|\mathbf{x}_i - \mathbf{x}_j|, h) \quad (2-39)$$

每个粒子都携带有流体的一部分质量，即上式中的 m_j ，而粒子密度也可使用上式来计算。经过推导可获得流体粒子的密度计算式公式^[52]：

$$\begin{aligned} \rho_i &= \sum_j \frac{m_j}{\rho_j} \rho_j W(|\mathbf{x}_i - \mathbf{x}_j|, h) \\ &= \sum_j m_j W_{ij} \end{aligned} \quad (2-40)$$

由于 SPH 法中的各项物理量计算几乎都需要使用到粒子的密度，因此密度属性是粒子最基础的属性之一，只要粒子的位置发生变化，就需使用上式来更新粒子的密度值。

2.6.2 微分算子的离散化

使用 SPH 法来求解流体的 NS 方程和速度无散度条件，则需使用到光滑核函数实现其中物理场微分项的离散化。参考上一节的讨论，式(2-39)可用来离散物理场，而该公式可进一步扩展用来求解物理场的微分。对式(2-39)的等号两端取梯度，则可得到粒子 i 位置上物理场 A 的梯度计算式^[142]：

$$\begin{aligned} \nabla A(\mathbf{x}_i) &= \nabla_i A = \sum_j \frac{m_j}{\rho_j} A_j \nabla W(|\mathbf{x}_i - \mathbf{x}_j|, h) \\ &= \sum_j \frac{m_j}{\rho_j} A_j \otimes \nabla W_{ij} \\ &= \sum_j \frac{m_j}{\rho_j} A_j \nabla W_{ij}^T \end{aligned} \quad (2-41)$$

其中，物理场 A 既可为标量场也可为矢量场，符号 \otimes 表示并矢计算，上述公式中核函数的梯度可写为： $\nabla W_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|} \left(\frac{dW(r, h)}{dr} \right)_{r=|\mathbf{x}_i - \mathbf{x}_j|}$ 。类似地，粒子 i 位置上矢量场或二阶张量场 \mathbf{A} 的散度计算式为：

$$\nabla \cdot A(\mathbf{x}_i) = \nabla_i \cdot A = \sum_j \frac{m_j}{\rho_j} \mathbf{A}_j \cdot \nabla W_{ij} = \sum_j \frac{m_j}{\rho_j} A_j^T \nabla W_{ij} \quad (2-42)$$

根据文献^[52]中的建议，为了减轻粒子分布不均匀问题的影响，式(2-41)与式(2-42)需要改写为如下形式。

$$\nabla A(\mathbf{x}_i) = \nabla_i A = \sum_j \frac{m_j}{\rho_j} (A_j - A_i) \otimes \nabla W_{ij} \quad (2-43)$$

$$\nabla \cdot \mathbf{A}(\mathbf{x}_i) = \nabla_i \cdot \mathbf{A} = \sum_j \frac{m_j}{\rho_j} (\mathbf{A}_j - \mathbf{A}_i) \cdot \nabla W_{ij} \quad (2-44)$$

除上述两种计算物理场梯度和散度的方法外，更常被使用的形式为^[97]：

$$\nabla A(\mathbf{x}_i) = \nabla_i A = \rho_i \sum_j m_j \left(\frac{A_j}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \otimes \nabla W_{ij} \quad (2-45)$$

$$\nabla \cdot \mathbf{A}(\mathbf{x}_i) = \nabla_i \cdot \mathbf{A} = \rho_i \sum_j m_j \left(\frac{\mathbf{A}_j}{\rho_i^2} + \frac{\mathbf{A}_j}{\rho_j^2} \right) \cdot \nabla W_{ij} \quad (2-46)$$

相比式(2-43)与式 (2-44), 式(2-45)与式(2-46)能够更好地保证流体的动量守恒特性, 但也会导致更严重的拉伸不稳定问题^[12]。

2.7 本章小结

本章从连续介质力学的角度讨论了粘性不可压缩流体的重要动力学模型和特性, 并讨论了 SPH 流体的仿真计算步骤及其离散化方法, 为后续的研究提供理论指导。本章的叙述和讨论可划分为四部分: 首先介绍了连续介质力学中的重要基础概念, 即连续介质假设和场的概念, 以及场的梯度、散度计算方法和物理含义; 然后在连续介质力学的基础上讨论了在模拟流体材料过程中涉及的重要方程以及重要的性质; 接下来基于流体基础理论, 引入了流体仿真的基本实现策略与方法; 最后, 本章讨论了 SPH 法的流体物理场与微分场的基本离散化方案。

第 3 章 适用于粘性不可压缩 SPH 流体的 SIMPLE 算法

3.1 引言

粘性不可压缩流体在现实中很常见，因此如何高效、逼真、快速地对其模拟是计算机图形学中的重要研究内容。尽管 SPH 法已经被广泛用于模拟低粘性流体^[94, 95]，但模拟高粘性流体时，SPH 法长期面临着稳定性差、难以使用较大的时间步长等问题^[116, 117, 119]。针对这一问题，有研究者提出了隐式粘性求解方法^[119, 122, 143]，这些方法显著提高了 SPH 法在模拟大粘性流体时的稳定性，使得仿真时间步长与粘性系数可以同时取较大值，但这也引入了其它的数值计算问题，即流体的不可压缩性约束条件和粘性约束条件难以同时被满足，导致不可压缩性与粘性求解之间产生了严重的冲突^[143]，影响了大粘性流体材料的仿真稳定性。

SPH 流体的粘性与不可压缩性之间的冲突是算子分裂法导致的^[70]。算子分裂法将原本应该同时产生作用的不可压缩性和粘性，分时分步骤地施加于流体的速度场中。这种策略虽然能够显著提高仿真计算的效率，降低编程实现的难度，但也导致在计算不可压缩性之后，粘性计算的结果被破坏，并最终影响了流体的仿真质量。Larionov 等人在研究中发现，由于粘性、不可压缩性之间的相互影响，导致大粘性流体不能稳定地保持形态与表面细节，较难模拟出粘性不可压缩流体特有的卷绳效应（Rope Coiling Effect）^[49]。

抑制这一冲突问题最直接的方法即是对粘性与不可压缩性进行整体迭代计算，以逐渐减轻两者之间的冲突。然而在本章的试验中发现，若选择已有的不可压缩性求解器（比如 DFSPH 法^[97]）与隐式粘性求算法进行迭代计算，迭代计算的结果不仅不收敛，反而会发散。因为不匹配的粘性和不可压缩性求解方法会在迭代过程中发生误差的积累，最终破坏掉流体原本相对正确的运动。抑制上述冲突问题的另一种策略，则是 Peer 等人提出的重建速度场的方法^[123, 143]。该方法通过削减形变速率然后重建流体速度场的方式来实现粘性不可压缩流体的模拟。由于不再需要分步骤地求解粘性与不可压缩性，因此两者间的冲突可以完全地避免。然而，该方法在重建速度场的过程中会引入严重的数值耗散，反而降低了粘性不可压缩性流体的仿真质量，导致该方法无法实现粘性流体的卷绳效应等复杂行为的模拟^[122]。

为了有效地提高大粘性不可压缩性流体材料的仿真质量，减轻不可压缩性与粘性求解器之间的冲突，本文借鉴了原欧拉网格方法中的 SIMPLE 算法（Semi-implicit method for pressure linked equations）^[144]。该方法通过对包含压强求解的相关计算进行整体迭代，来提高模拟计算的精度（图 3-1），因此可使用 SIMPLE 算法的迭代策略减轻粘性与不可压缩性之间的冲突。若粘性与不可压缩性算法严格由流体的 NS 推导获得，而且使用相同的方法来离散各微分算子与物理场，理

论上可以有效避免误差的累积漂移，进而保证迭代的稳定收敛。

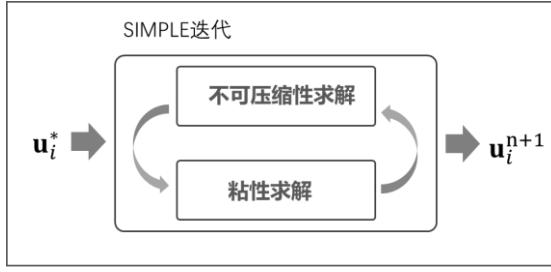


图 3-1 本文 SIMPLE 算法的基本实现方式。

Figure 3-1 The framework of our SIMPLE algorithm.

在本文的研究工作之前，尚没有使用 SIMPLE 算法用于模拟 SPH 流体的相关研究，其原因是 SPH 法中的粒子很难保证类似于欧拉网格法中网格节点的正交与均匀性，导致基于欧拉网格离散的 SIMPLE 迭代算法很难直接套用在 SPH 法中。因此若使用 SIMPLE 迭代算法来减轻 SPH 流体的粘性与不可压缩性间的冲突，需要针对 SPH 法的特性对原 SIMPLE 迭代算法进行改进，具体的实现方法如下：由于近似投影法可以降低 SPH 法零能问题的影响^[99]，因此本文采用该方法离散了流体压强泊松方程；为降低粒子分布不均匀以及邻域缺失问题对计算的影响，压强泊松方程的离散过程中使用了半解析式边界条件；为避免流体不可压缩性求解过程因密度漂移而导致体积损失（Volume Loss），需要对压强泊松方程的源项进行密度补偿。基于这些改进，本章设计了面向 SPH 流体的 SIMPLE 迭代算法。经过测试，这种基于 SPH 法的 SIMPLE 迭代过程可以稳定地收敛，有效抑制粘性与不可压缩性间的冲突，进而显著提高大粘性材料以及粘性可变的非牛顿流体材料的仿真稳定性。

3.2 粘性不可压缩性流体 SIMPLE 迭代算法概述

本节将详细讨论 SPH 法中粘性与不可压缩性之间冲突产生的原因，以及如何使用 SIMPLE 迭代算法来减轻这一冲突。

首先本节以计算机图形学中被广泛使用的预测—矫正策略^[145]为例，来说明粘性流体材料中粘性与不可压缩性间的冲突问题。粘性不可压缩流体运动的控制方程为 NS 方程与速度无散度条件，其具体形式如下：

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \mathbf{f} \quad (3-1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3-2)$$

在流体模拟计算的“预测”步骤中，不考虑不可压缩性，即忽略式(3-1)中压强梯度项的作用，只考虑流体粘性项与体积力项作用，采用式(3-1)更新流体的速度场后，流体速度场的不可压缩性条件会被破坏，此时流体速度场不再满足式(3-2)。

在“修正”步骤中，则不再考虑流体的粘性和体积力作用，只考虑不可压缩性条件（式(3-2)）作用，根据速度散度计算出压强，然后使用压强梯度项来更新流体的速度场，此时流体的速度场虽然满足了速度无散度条件（式(3-2)），但此时流体速度场不再满足式(3-1)。

消除粘性与不可压缩性求解步骤之间的相互干扰，则需要流体同时满足式(3-1)与式(3-2)。因此忽略掉体积力作用，对式(3-1)使用后向欧拉法对速度的时间微分进行离散化，可得到如下的公式：

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\delta t} = -\nabla p^{n+1} + \mu \nabla \cdot \nabla \mathbf{u}^{n+1} + \mu \nabla \nabla \cdot \mathbf{u}^{n+1} \quad (3-3)$$

根据速度无散度条件（式(3-2)），可得到： $\mu \nabla \nabla \cdot \mathbf{u}^{n+1} = \mu \nabla (\nabla \cdot \mathbf{u}^{n+1}) = 0$ ，因此上述方程可以进一步简化为如下形式：

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\delta t} = -\nabla p^{n+1} + \mu \nabla \cdot \nabla \mathbf{u}^{n+1} \quad (3-4)$$

上式中的 $n + 1$ 为每次仿真步（step）结束后得到的新值。对上式的等式两端同时取散度即可得到包含有剪切应力作用的压强的泊松方程：

$$\nabla \cdot \nabla p^{n+1} = \frac{\rho}{\delta t} \nabla \cdot \mathbf{u}^* + \nabla \cdot (\mu \nabla \cdot \nabla \mathbf{u}^{n+1}) \quad (3-5)$$

若能够多次交替求解公式(3-4)与式(3-5)获得 \mathbf{u}^{n+1} 与 p^{n+1} ，理论上可减轻流体粘性与不可压缩性作用间的相互干扰。但这种迭代策略存在两个问题：第一，上述每个微分方程中等式两端都包含有未知量 \mathbf{u}^{n+1} 与 p^{n+1} ，直接将这两个微分方程离散为线性系统是较为困难的；第二，这种迭代策略是很不稳定的，原因是式(3-5)中存在三阶微分项： $\nabla \cdot (\mu \nabla \cdot \nabla \mathbf{u}^{n+1})$ ，而对于粒子分布常常不均匀、不充足、不正交的 SPH 法难以实现三阶微分场的插值计算，因此采用此策略构建的流体仿真方法会非常的不稳定。

原欧拉网格法中的 SIMPLE 算法^[144]，可以有效地解决上述问题。该策略能够保证不可压缩性与粘性的迭代计算收敛，获得同时满足粘性和不可压缩性约束条件的速度场，由此可抑制不可压缩性与粘性之间的冲突。在 SIMPLE 迭代算法中，流体可以首先通过如下动量方程计算获得粘性的剪切应力与压强作用下的速度场 \mathbf{u}' ：

$$\rho \frac{\mathbf{u}' - \mathbf{u}^*}{\delta t} = -\nabla p^* + \mu \nabla \cdot \nabla \mathbf{u}' \quad (3-6)$$

在第一次迭代计算时，由于流体压强 p^* 尚未被求解出来，因此可以先使用上一帧计算的压强结果或直接以 0 作为初值来计算该方程。但由于粘性项的存在，该方程的计算结果 \mathbf{u}' 是不满足速度无散度条件的，即： $\nabla \cdot \mathbf{u}' \neq 0$ ，因此当该方程结束计算后，需要对流体的速度场 \mathbf{u}' 施加不可压缩性，获取能够消除掉散度误差的压强场 p' 。

求解压强 p' 的方法可以通过压强泊松方程来实现：首先忽略掉粘性项作用，流体的动量方程可转换为如下形式：

$$\mathbf{u}'' = \mathbf{u}' - \frac{\delta t}{\rho} \nabla p' \quad (3-7)$$

其中 \mathbf{u}'' 是满足速度无散度条件的速度场，即： $\nabla \cdot \mathbf{u}'' \neq 0$ ， p' 为能够消除速度散度误差的流体压强场。由于缺少约束条件，式(3-7)中未知量 p' 无确定解，因此需要将 \mathbf{u}'' 的无散度约束条件带入到该方程中。具体方法是对式(3-7)等号两端同时求速度散度，然后目标速度场 \mathbf{u}'' 就会被消除，得到如下求解压强的方程，即压强泊松方程：

$$\nabla \cdot \frac{\delta t}{\rho} \nabla p' = \nabla \cdot \mathbf{u}' \quad (3-8)$$

相比于式(3-6)，式(3-8)舍去了流体动量方程（式(3-4)）中的粘性项。虽然省略掉粘性项后的式(3-7)与式(3-8)可能会导致计算偏离精确解，但在测试中发现交替对式(3-6)与式(3-8)进行求解并迭代，迭代依然会收敛（如图 3-10 所示），式(3-7)因避免引入三阶微分而删去的粘性项并不会对原 SIMPLE 迭代的收敛性造成影响。而这也是 SIMPLE 迭代算法名称中“半隐式”（Semi-implicit）的含义，即：在求解不可压缩性时，并不需要完全包含所有的速度微分项^[144]。

经过上述理论推演，用于模拟粘性不可压缩流体的 SIMPLE 迭代的实现过程可梳理为如下步骤：

- 步骤 1. 将上一帧的压强计算结果作为压强的初值，或是将压强置零；
- 步骤 2. 计算流体的动量方程（式(3-6)），求解 \mathbf{u}' ；
- 步骤 3. 计算压强泊松方程（式(3-8)），求解 p' ；
- 步骤 4. 将 p' 累计到 p^* ；
- 步骤 5. 返回步骤 2 并进行迭代，直到迭代收敛到阈值或达到迭代次数上限。

上述迭代步骤中所使用到的方程均由流体的 NS 方程（式(3-1)）与速度无散度条件（式(3-2)）推导而来。而若迭代计算中使用的方程与上述式(3-6)、式(3-8)区别较大，或者并不是由 NS 方程（式(3-1)）与速度无散度条件（式(3-2)）推导获得的，则迭代可能无法收敛。

3.3 将 SIMPLE 算法扩展到 SPH 法中

虽然 SIMPLE 算法理论上能够提高粘性不可压缩流体的仿真质量，但该方法的离散化需要使用空间网格来实现。相较于欧拉网格法，SPH 法中的粒子存在分布不均、邻域缺失等问题^[48]，因此在对原 SIMPLE 算法使用 SPH 粒子离散时，需要尽可能降低粒子分布不均匀、邻域缺失问题的影响。

3.3.1 离散化方案选择

参考 SPH 法的离散化方案^[34]，任意物理场函数 ϕ 都可以使用一组离散的粒子来表示。假设有一粒子 i ，该粒子位置上的物理量 ϕ 的近似值可以通过其邻域粒子 j 计算获得：

$$\phi_i = \sum_j V_j \phi_j W_{ij} \quad (3-9)$$

在 SPH 法中，粒子 i 位置上物理场 ϕ 的梯度 ($\nabla_i \phi$) 有非常多的离散化方法，其中泰勒级数一致性 (Tylor-series model) 梯度离散化方法形式如下：

$$\nabla_i \phi = - \sum_j V_j \phi_{ij} \nabla_i W_{ij} \quad (3-10)$$

相对于其它的梯度离散化方法，式(3-10)的方法虽然不能保证动量守恒，但可以更精确地实现零阶精度，也能够相对更稳定的维持流体的拉伸状态^[12]。粒子 i 位置上，物理场 ϕ 的拉普拉斯计算 ($\nabla \cdot \nabla_i \phi$) 方法也有较多的选择。由于采用类似于网格有限差分的方法，可以有效减轻 SPH 法中的零能问题的影响^[68, 99]，因此本文选用此方法来离散拉普拉斯算符，其具体形式为：

$$\nabla \cdot \nabla_i \phi = \sum_j V_j \frac{\phi_{ij} \cdot \mathbf{x}_{ij}}{\mathbf{x}_{ij}^2 + \eta^2} \nabla_i W_{ij} \quad (3-11)$$

其中， η 为取值较小的常数，用于避免因粒子 i 与 j 距离过近而导致上式的分母为零的问题。

3.3.2 动量方程与粘性求解

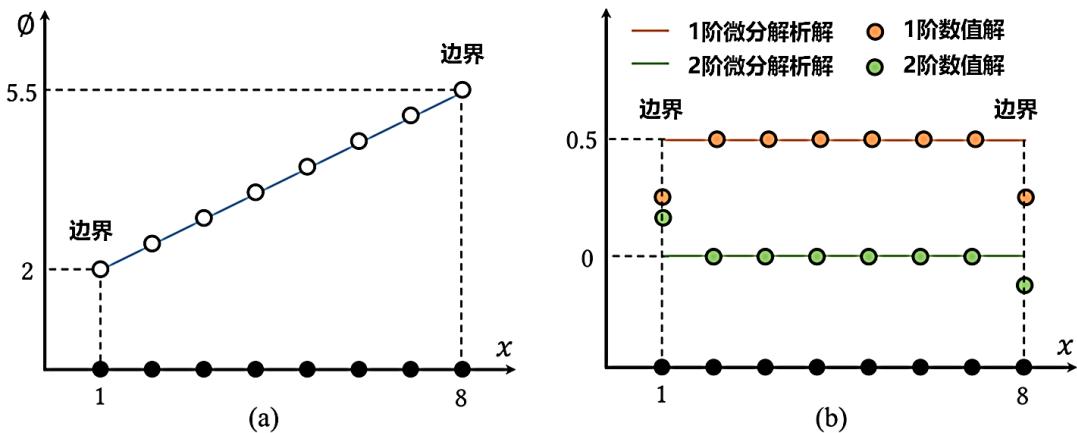


图 3-2 SPH 粒子的邻域缺失问题。

Figure 3-2 The particle deficiency problem in SPH.

在求解上述 SIMPLE 迭代算法中的式(3-6)之前，则需要离散该式中的压强梯度项 ∇p^* 与粘性项 $\mu \nabla \cdot \nabla \mathbf{u}'$ 。如果粒子位于远离流体表面的位置，则可以直接采用式(3-10)与式(3-11)对 ∇p^* 与 $\mu \nabla \cdot \nabla \mathbf{u}'$ 进行离散化。而当粒子位于流体表面附近时，

这两项的计算结果都可能会受到粒子邻域缺失问题的影响。为直观展示粒子邻域缺失问题的影响，图 3-2 比较了某一维物理量微分的解析解曲线与数值解曲线。该测试中计算所使用的核函数为三次样条（cubic spline）形式核函数^[68]，平滑距离为 2 ($h = 2$)。在图 3-2(a)中有一组携带有物理量 ϕ 的 SPH 粒子， ϕ 的空间分布为线性函数；图 3-2(b)为物理量 ϕ 的一、二阶微分的解析解曲线与 SPH 数值计算解的对比。在图 3-2(b)中可以观察到，边界位置上使用 SPH 法计算的一阶、二阶微分与解析解偏差很大，这一偏差会在模拟过程中引入较为严重的错误。

(1) 半解析式边界处理与压强梯度的离散化。解决边界附近粒子邻域缺失的常用方法，是在流体边界以外布设幽灵气态（ghost-air）粒子^[128]，但布设幽灵粒子往往需要花费更多的计算资源，会降低仿真计算的效率，而半解析式边界处理方法可以有效的避免这一问题^[104-106]。参考这部分研究工作，如果液态粒子位于流体边界附近，则它含有两部分的邻域粒子：气态幽灵粒子 j^a 和液态粒子 j^l 。若粒子 j^l 与 j^a 是均匀分布的，则可以得到如下的关系式。

$$\sum_j V_j \nabla_i W_{ij} = \sum_{jl} V_{jl} \nabla_i W_{ijl} + \sum_{ja} V_{ja} \nabla_i W_{ija} = \nabla 1 = 0 \quad (3-12)$$

因此，位于流体表面附近的粒子其压强梯度项的离散化可以写为如下形式：

$$\begin{aligned} \nabla_i p^* &= \sum_{jl} V_{jl} p_{jl}^* \nabla_i W_{ijl} + \sum_{ja} V_{ja} p_{ja}^* \nabla_i W_{ija} \\ &\quad - p_i^* \sum_j V_j \nabla_i W_{ij} \end{aligned} \quad (3-13)$$

多数情况下，流体表面外幽灵气态粒子的压强值取值为常数，为了便于处理，常将其设置为零，也即： $p_{ja}^* = 0$ 。将其带入上式，则可得到以下流体粒子不同位置上的梯度计算式：

$$\nabla_i p^* = \begin{cases} \sum_{jl} V_{jl} (p_{jl}^* - p_i^*) \nabla_i W_{ijl} & i \in \mathfrak{B} \\ \sum_{jl} V_{jl} p_i^* \nabla_i W_{ijl} & i \notin \mathfrak{B} \end{cases} \quad (3-14)$$

上式中的 \mathfrak{B} 代表的是支持域被流体边界截断的粒子的集合，也即边界粒子集合。判断粒子是否是边界粒子的方法将在下一节进行详细讨论。

(2) 粘性项的离散化。液体表面外的幽灵气态粒子速度一般认为与其临近流体的速度相同，也即： $\mathbf{u}_{ja} = \mathbf{u}_i$ ，以避免这类粒子对流体的运动产生非物理的粘性。因此式(3-6)中的粘性项可以直接采用如下方程计算：

$$\nabla \cdot \nabla_i \mathbf{u}' = 2(d+2) \sum_{jl} V_{jl} \frac{\mathbf{u}'_{ijl} \cdot \mathbf{x}_{ijl}}{|\mathbf{x}_{ijl}|^2 + \eta^2} \nabla_i W_{ij} \quad (3-15)$$

Monaghan 与 Weiler 等人的研究工作中曾使用过上述方程^[52, 122]，本文的研究中进一步发现上述方程中隐含了半解析式边界条件，因此该方程能够较好地应对

SPH 法中的粒子邻域缺失的问题。

基于上述的离散化方法，则原式(3-6)的求解可以通过分别求解如下两公式来实现：

$$\mathbf{u}^{**} = \mathbf{u}^* - \frac{\delta t}{\rho} \nabla p^* \quad (3-16)$$

$$\mathbf{u}' = \mathbf{u}^{**} - \mu \frac{\delta t}{\rho} \nabla \cdot \nabla \mathbf{u}' \quad (3-17)$$

其中，式(3-17)是用于求解粘性作用的隐式微分方程，该方程可以使用式(3-15)转化为对称正定的线性系统，然后通过共轭梯度法^[122]快速求解出粘性作用下的流体速度场。完成式(3-16)与式(3-17)的求解之后，获得的速度 \mathbf{u}' 会被带入到压强泊松方程中，用于求解流体的不可压缩性。这方面的讨论将在下一节中展开。

3.3.3 不可压缩性求解

参考本文 3.2 节讨论的内容，为了保证粘性不可压缩性流体的 SIMPLE 迭代的收敛性，不可压缩性需要基于压强泊松方程计算实现（式(3-8)），进而将流体的速度场投影到无散度空间中。由式(3-8)可知，流体粒子 i 位置上的压强泊松如下：

$$\nabla_i \cdot \frac{\delta t}{\rho} \nabla p' = \nabla_i \cdot \mathbf{u}' \quad (3-18)$$

由于对流体施加不可压缩性后，流体的密度会接近于静止密度 ρ_0 ，此处可以假设流体密度场分布为不随空间变化的常数，因此流体粒子 i 位置上的压强泊松方程可进一步改写为如下形式：

$$\nabla_i^2 p' = \frac{\rho_0}{\delta t} \nabla_i \cdot \mathbf{u}' \quad (3-19)$$

如上式所示，压强泊松方程需要离散的算子有两项，分别为压强拉普拉斯算子 ($\nabla_i^2 p'$) 与散度算子 ($\nabla_i \cdot \mathbf{u}'$)。离散化后的泊松方程会转化为一线性系统。

(1) 泊松方程拉普拉斯项的离散化。考虑到 SPH 法容易出现粒子分布不均匀的问题，因此压强泊松方程中的拉普拉斯项，可以采用对于粒子分布不过于敏感的近似投影法来离散^[99]，即：

$$\nabla_i^2 p' = - \sum_{j,l} (p_i - p_j) \frac{V_j}{|\mathbf{x}_{ij}| + \eta^2} \frac{\partial W_{ij}}{\partial r_{ij}} \quad (3-20)$$

使用上述离散化方法后，式(3-19)的左端会转化为系数矩阵与未知向量的乘法： $\nabla_i^2 p' = \mathbf{A} \mathbf{p}'$ 。其中系数矩阵 \mathbf{A} 为对称矩阵。但由于矩阵 \mathbf{A} 中每行非对角元素的绝对值之和与对角元素的绝对值总是相等的，导致矩阵 \mathbf{A} 是不可逆的，因此需要对其施加狄雷克利边界 (Dirichlet boundary) 条件来保证上述线性系统的可解性^[103]。

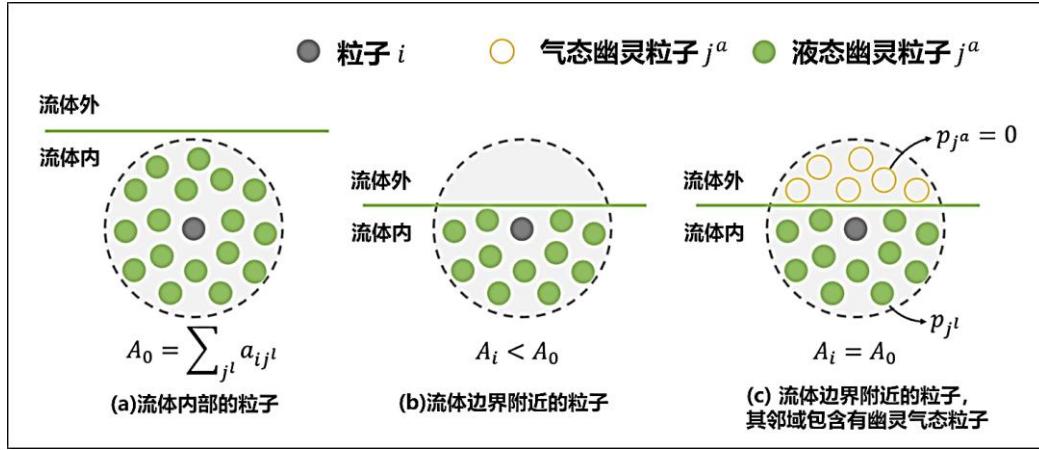


图 3-3 半解析式边界条件。

Figure 3-3 Semi-analytical boundary condition.

(2) 边界条件。压强泊松方程的狄雷克利边界条件的本质为流体表面压强为 0 的条件。该条件的常规施加方法是在流体表面之外布设压强为零的幽灵气态粒子，并使用幽灵气态粒子与流体粒子共同离散化压强泊松方程的拉普拉斯项，但这种幽灵气态粒子需要耗费额外的计算资源，而且最终的计算精确性也会受到幽灵气态粒子分布均匀性的影响。本文采用了半解析式自由表面边界条件来避免这些问题^[104-106]。假设某粒子位于流体的自由表面附近，则其邻域可能包含有气态幽灵粒子 j^a 和液态粒子 j^l （如图 3-3(c) 所示），式(3-20)可转换为如下形式：

$$\nabla_i^2 p' = - \sum_{j^l} a_{ij^l} p'_i - \sum_{j^a} a_{ij^a} p'_i + \sum_{j^l} a_{ij^l} p'_{j^l} \quad (3-21)$$

其中 a_{ij^l} 的计算式为： $a_{ij^l} = -\frac{V_j}{|x_{ij}|+\eta^2} \frac{\partial W_{ij}}{\partial r_{ij}}$ 。上述公式隐含了流体的狄雷克利表面边界条件，原系数矩阵也就转换为了非奇异的弱对角占优阵。进一步将流体内部远离表面的粒子考虑在内，原压强泊松方程（式(3-8)）即可离散为下式。

$$\nabla_i^2 p' = \begin{cases} \sum_{j^l} a_{ij^l} (p'_j - p'_i) & i \in \mathfrak{B} \\ -A_0 p'_i + \sum_{j^l} a_{ij^l} p'_{j^l} & i \notin \mathfrak{B} \end{cases} \quad (3-22)$$

其中 A_0 是在仿真开始之前，流体未处于压缩状态时，选择流体内部的粒子计算获得的（如图 3-3(a)），其计算公式为 $A_0 = \sum_{j^l} a_{ij^l}$ 。在仿真开始后，若粒子位于流体边界附近，则其液态粒子邻域数量相对较少，此时该粒子对应的系数值 $A_i = \sum_{j^l} a_{ij^l}$ 满足条件： $A_i < A_0$ 。因此流体中的粒子可以根据其系数值 A_i 与 A_0 大小关系来判断其是否位于表面边界附近，即：

$$\mathfrak{B} = \{i \mid A_i < A_0\} \quad (3-23)$$

(3) 速度散度的离散化。在实现了压强泊松方程中（式(3-19)）拉普拉斯项的离散化之后，下一个待离散的算符是等式右端的速度散度项，即： $\nabla_i \cdot \mathbf{u}'$ 。为

避免幽灵气态粒子对流体的运动造成粘性等非物理的影响，需要假设流体外部的幽灵气态粒子总是与流体边界附近粒子有相同的运动速度，即： $\mathbf{u}'_{jl} = \mathbf{u}'_i$ 。因此采用式(3-10)的离散化方法，可以将压强泊松的速度散度离散为如下形式：

$$\nabla_i \cdot \mathbf{u}' = \sum_{jl} V_{jl} (\mathbf{u}'_{jl} - \mathbf{u}'_i) \nabla_i W_{ijl} \quad (3-24)$$

(4) 密度漂移补偿。通过式(3-19)求解流体压强来施加流体不可压缩性，会导致粒子密度的漂移问题。式(3-8)形式的泊松方程源项中（等式右端项）只包含有流体的速度散度误差。参考流体的连续性方程（式(2-8)），严格保证速度无散度条件可以避免流体密度误差的产生。但 SPH 法并不能保证每一帧计算都将速度误差完全消除，因此误差会随着仿真的进行逐渐积累，直到流体产生严重的体积丢失问题导致仿真失败。为了避免这一问题的出现，需要对原压强泊松方程的源项进行密度补偿（ECS，Error Compensating Source）^[146]。补偿后的压强泊松方程形式如下：

$$\nabla_i^2 p' = \frac{\rho_0}{\delta t} \nabla_i \cdot \mathbf{u}' + \Lambda_i \quad (3-25)$$

$$\Lambda_i = \alpha \frac{\max(\rho_i - \rho_0, \rho_0)}{\rho_0 \delta t} \quad (3-26)$$

上式中的 Λ_i 为泊松方程的源项密度补偿项， α 为常数，其数量级应与 $\frac{\rho_0}{\delta t}$ 保持相同，即： $\alpha = 10^6$ 。当流体某粒子*i*的密度超过了其静态密度时， Λ_i 的存在可以在流体粒子*i*及其邻域粒子位置上产生相应的压强，使粒子间产生斥力抵抗压缩，避免因误差漂移、累积，导致流体体积丢失。

在完成压强泊松方程边界条件的施加以及源项的密度补偿处理之后，原压强泊松方程（式(3-19)）就可以转化为对称正定的线性系统，使用共轭梯度法可以快速地求解出压强。然后，计算动量方程（式(3-17)），即可获得满足不可压缩性条件的流体速度场。

(5) 不可压缩性求解算法的测试。为了验证上述不可压缩性求解算法的稳定性与有效性，图 3-4 使用不同的时间步长模拟了流体溃坝场景。该测试中压强泊松方程求解的收敛阈值为最大误差的 10^{-5} ，粒子的采样间距为 $0.005m$ ，平滑距离为 $0.0125m$ ，重力加速度为 $9.8m/s^2$ 。图 3-4 (a)、(b)、(c)、(d)中使用的时间步长分别为 $0.002s$ 、 $0.003s$ 、 $0.004s$ 、 $0.005s$ 。由该图可知，即使是使用了非常大的时间步长（ $0.005s$ ），模拟结果依然稳定，而且不同时间步长下模拟的流体，其行为也基本一致。

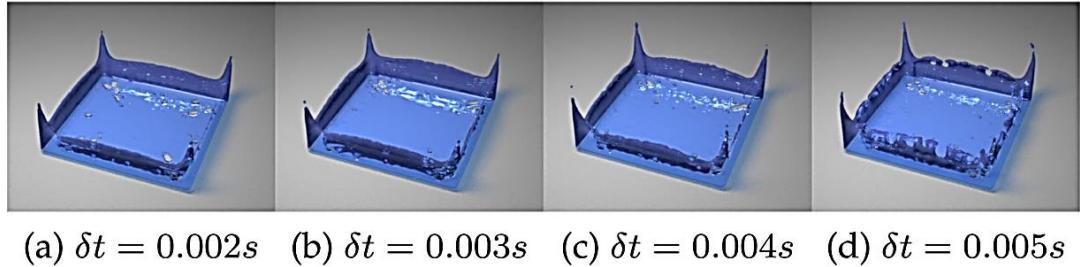


图 3-4 不可压缩性求解算法的测试。

Figure 3-4 A test of the incompressibility solver.

```

While  $t < t_{stop}$  do:
     $\delta t \leftarrow CFL(\mathbf{u}_i^n);$  //根据 CFL 条件更新时间步长
    For all fluid particle  $i$  do: //更新粒子位置与速度
         $\mathbf{u}_i^* \leftarrow \mathbf{u}_i^n + \delta t \mathbf{F}^{ext};$ 
         $\mathbf{x}_i^* \leftarrow \mathbf{x}_i^n + \delta t \mathbf{u}_i^*;$ 
    For all fluid particle  $i$  do: //邻域查找
         $\mathcal{N}_i \leftarrow \text{Find neighbors using } \mathbf{x}_i^*$ 
    Detect boundary particles;
    Set SIMPLE iteration number  $iter_{max}$  do: //设置 SIMPLE 迭代次数
    While  $iter < iter_{max}$  do:
        For all fluid particle  $i$  do:
            Compute  $\nabla_i \cdot \mathbf{u}'$  (3-24) //计算泊松方程源项 (速度散度误差)
            Add  $\Lambda_i$  to  $\nabla_i \cdot \mathbf{u}'$  (3-26) //在速度散度加密度补偿项
        While  $\epsilon^p > \epsilon_0^p$  do: //不可压缩性迭代
            Compute one CG iteration to update  $p'$  (3-19)
            Compute relative error  $\epsilon^p$ 
        For all fluid particle  $i$  do: //累计迭代中的压强
            Compute  $p^* += p'$ 
        For all fluid particle  $i$  do: //计算压强梯度并更新速度
            Compute  $\nabla_i p^*$  and update velocity (3-16)
        While  $\epsilon^v > \epsilon_0^v$  do: //粘性迭代
            Compute one CG iteration for viscosity (3-17)
            Compute relative error  $\epsilon^v$ 
        For all fluid particle  $i$  do: //更新粒子速度
            Set new velocity  $\mathbf{u}_i^{n+1} = \mathbf{u}_i'$ 

```

图 3-5 面向粘性不可压缩 SPH 流体模拟的 SIMPLE 算法。

Figure 3-5 Adapted SIMPLE algorithm for viscous incompressible SPH fluids.

3.3.4 迭代与验证

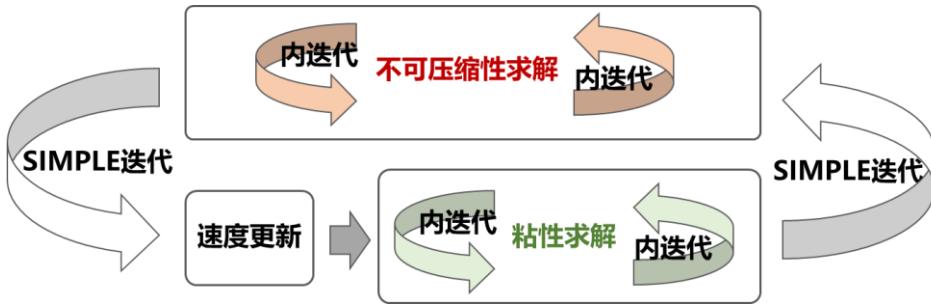


图 3-6 本文方法的内迭代与 SIMPLE 迭代。

Figure 3-6 Inner-iterations and SIMPLE iteration in our method.

使用上文中讨论的方法可以实现粘性不可压缩性流体的仿真算法的构建，算法实现的流程如图 3-5 所示。本文改进的 SIMPLE 算法包含有两个层次的迭代计算，分别是求解隐式粘性方程和求解压强泊松方程过程中的迭代计算，即：内迭代；以及粘性、不可压缩性交替求解的整体迭代计算，即：外迭代或 SIMPLE 迭代（如图 3-6 所示）。若要验证本文 SIMPLE 算法的有效性，则首先要保证内迭代是收敛的，以确保内迭代中两种求解过程对流体施加了粘性约束和不可压缩性约束。图 3-7 记录了本文方法模拟卷绳效应（图 3-8 场景）时，粘性与不可压缩性迭代计算的收敛曲线。由该图中的曲线可知，本文粘性求解与不可压缩性求解算法可以快速地消除相对误差，而且共轭梯度法的收敛速度比雅克比法快很多。

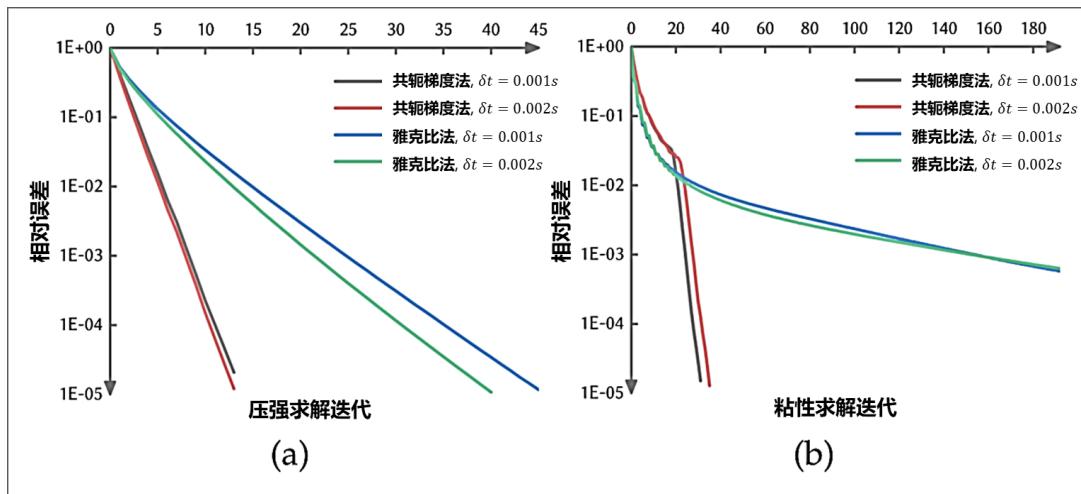


图 3-7 SIMPLE 算法中内迭代收敛性分析。(a) 不可压缩性求解器的迭代收敛曲线；
(b) 粘性求解器的迭代收敛曲线。

Figure 3-7 Inner-iterative convergence analysis of our SIMPLE method.

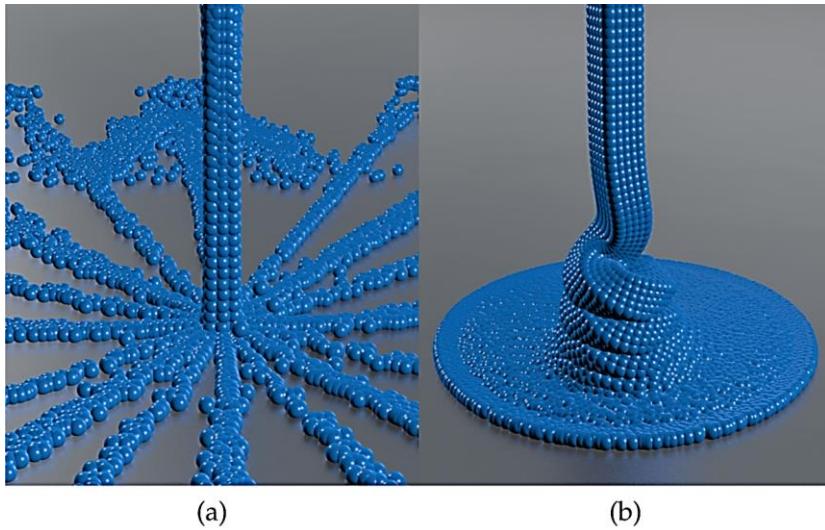


图 3-8 粘性系数为 $500\text{Pa}\cdot\text{s}$ 流体的卷绳效应模拟。(a) 先计算粘性; (b) 先计算不可压缩性。

Figure 3-8 A fluid with a viscosity $500\text{Pa}\cdot\text{s}$ is simulated.

若只使用一次 SIMPLE 迭代来模拟大粘性流体的卷绳效应，流体粘性、不可压缩性之间的冲突会严重影响流体的行为。如图 3-8 所示，在一帧计算中，如果先计算流体的粘性，然后再计算流体的不可压缩性（图 3-8(a)），则流体的粘性计算结果会被不可压缩性计算干扰，导致流体原本卷绳效应消失；而若在一帧计算中先计算流体的不可压缩性，然后再计算流体的粘性(图 3-8(b))，则可观察到粘性计算影响到了不可压缩性计算的结果，导致流体发生了明显的体积丢失问题。

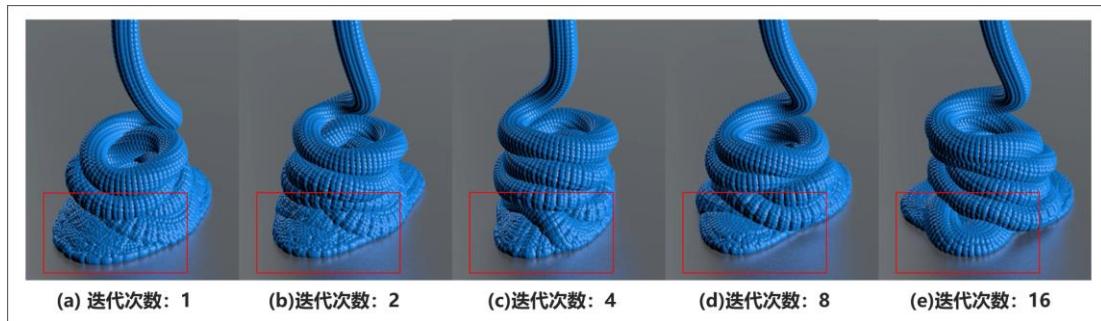


图 3-9 使用不同 SIMPLE 迭代次数模拟的粘性流体卷绳效应。

Figure 3-9 The rope coiling effect of viscous fluids are simulated using different SIMPLE iteration counts.

上述粘性与不可压缩性之间的冲突，可以尝试通过增加 SIMPLE 算法的迭代次数式逐渐削弱。图 3-9 使用不同的 SIMPLE 迭代次数模拟了流体的卷绳效应。从测试结果中可观察到：随着迭代次数的增加，流体的表面细节逐渐变得锐利，而且流体的形态也保持得更加稳定。

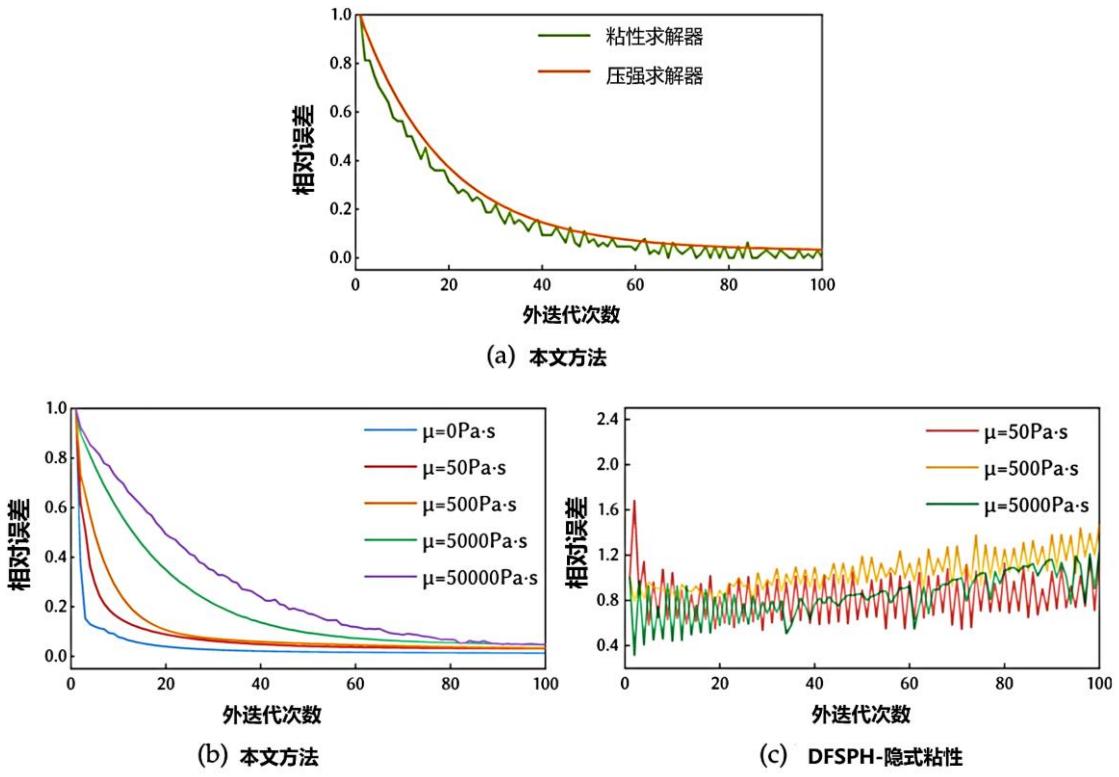


图 3-10 SIMPLE 迭代的收敛分析。 (a)粘性系数为 $5000 \text{ Pa}\cdot\text{s}$ 时，粘性与不可压缩性求解相对误差的变化曲线；(b)不同粘性系数下的速度散度相对误差的变化曲线；(c)将 SIMPLE 算法中不可压缩性求解器替换为 DFSPH 法后的速度散度误差变化曲线。

Figure 3-10 Evaluation of the SIMPLE convergence rate.

为验证 SIMPLE 迭代是否能够收敛到全局最优解，图 3-10(b)使用了与图 3-9 场景中相同条件测试了迭代中速度散度误差的变化曲线。该项测试中内迭代的收敛阈值为最大误差的 10^{-5} 。在图 3-10(b)中可观察到，由于粘性会干扰不可压缩性的计算结果，因此粘性计算完成后会产生新的散度误差，但随着 SIMPLE 迭代次数的增加，散度的相对误差逐渐减小，而且粘性与不可压缩性计算的结果也逐渐趋于稳定，这一结果证明了随着 SIMPLE 迭代次数的增加，不可压缩性与粘性之间的冲突会被逐渐抑制。图 3-10(a)表明粘性与不可压缩性求解的相对误差随着迭代的进行而逐渐降低，该收敛曲线进一步证明了本文方法可有效抑制粘性与不可压缩性间的冲突。

如若将本文方法中的不可压缩性求解器替换为现阶段计算机图形学领域中最先进的不可压缩性求解器——DFSPH 法^[90, 97]，在同样的测试场景中，迭代计算不能收敛(如图 3-10(c)所示)。尽管 DFSPH 可以削减流体速度场的散度误差，但 DFSPH 法并不是由流体的 NS 方程与无散度条件推导而来的，因此它与粘性求解算法之间存在不匹配的问题，这会导致迭代计算无法收敛到全局最优解。

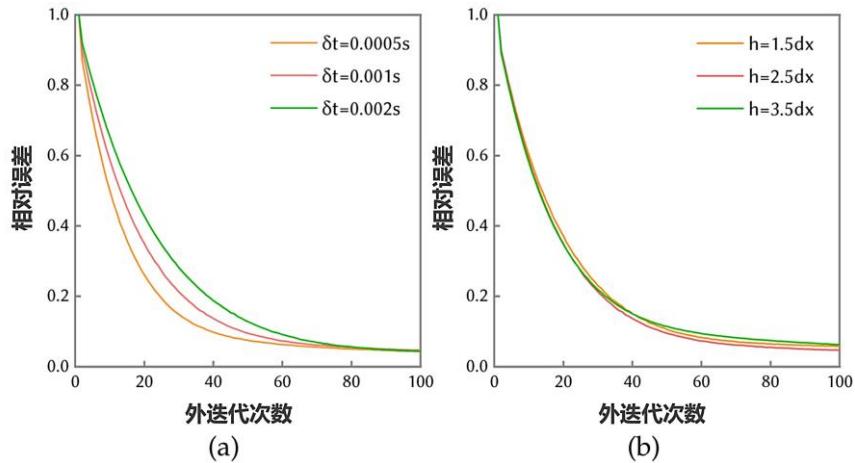


图 3-11 时间步长与粒子采样距离对于 SIMPLE 迭代收敛速率的影响。(a)不同时间步长下的收敛曲线; (b)不同采样间距下的收敛曲线。

Figure 3-11 Evaluation of the influence of the time step size and particle sampling distance on the global convergence rate of our method.

为进一步验证其它因素对 SIMPLE 迭代收敛速率的影响, 图 3-11 测试了时间步长与粒子间距对 SIMPLE 迭代收敛速度的影响。该测试中的流体粘性系数为 $5000 Pa \cdot s$, 图 3-11 (a) 中粒子采样间距为 2.5 倍的支持域半径; 图 3-11 (b) 粒子时间步长为 0.001s。由该图中的曲线可知, 时间步长对于 SIMPLE 迭代的收敛速度影响相对较小, 而粒子的采样间距对于迭代的收敛速度几乎没有影响。此外, 在图 3-10(b) 中, 不同的粘性系数下对应着不同的收敛曲线, 表明本文方法的收敛速度会受到粘性系数的影响, 且粘性系数越大收敛速度越慢。其原因是较大的粘性系数会产生较大的剪切应力, 相应地也就会与压强产生更严重的冲突。

图 3-12 展示了使用不同迭代次数模拟的不同粘性的流体材料, 在这一场景中, 每个流体方块都由 8 千个粒子构成, 每个流体的迭代次数则被标注在了对应流体的前方。为了能够保留粘性流体的表面细节, 该测试中没有使用表面张力计算。在该图中可以观察到, 粘性系数无论是设置为较小值 ($500 Pa \cdot s$) 或是较大值 ($10000 Pa \cdot s$), 只要迭代次数设置为一次, 流体就会快速的“融化”, 并丢失掉流体原有的表面细节。若将迭代次数设置为 5 次, 就可以显著增强流体表面的细节。随着迭代次数的增加, 流体形态逐渐趋于稳定, 且具有更加锐利的形态。

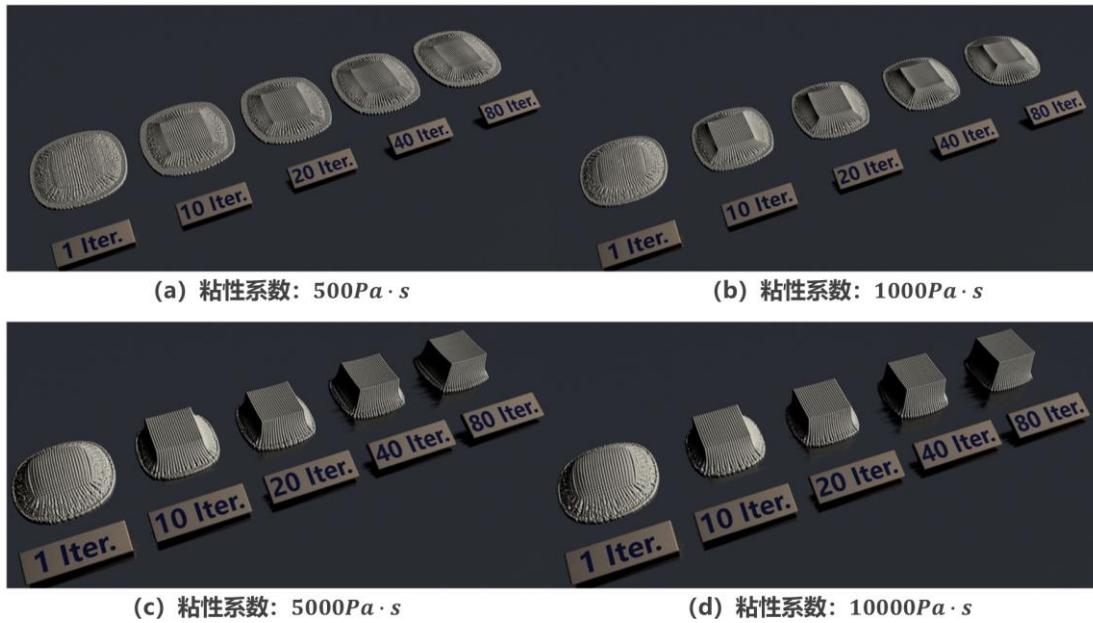


图 3-12 SIMPLE 迭代次数对粘性流体行为的影响。

Figure 3-12 Evaluation of the dependence of viscous behaviors on the SIMPLE iteration number.

由本节中的测试结果可知(图 3-10(b)), 本文所使用的 SIMPLE 迭代算法收敛的趋势明显, 但收敛速度相对较慢, 而且由于每次迭代计算都要重新计算一次粘性和不可压缩性, 因此迭代过程引入的计算量非常大。但图 3-12 测试结果也表明 10 次迭代就已能够显著提高大粘性流体的视觉真实性, 继续提高迭代次数, 效果的改善已不再明显。因此, 10 次以内的迭代已经可以满足多数面向计算机图形学应用场景的需求。

在本文方法中, 增加迭代次数可以使流体的表面细节变得更加锐利, 但并不会增大流体的粘性。本文的 SIMPLE 迭代算法的本质, 是通过迭代获取能够同时满足速度无散度约束条件 ($\nabla \cdot \mathbf{u} = \mathbf{0}$) 和粘性约束条件 ($\mathbf{u} = \mathbf{u}^{old} + \mu \frac{\delta t}{\rho} \nabla \cdot \nabla \mathbf{u}$) 的流体速度场。而本文的迭代计算能够收敛, 则说明迭代最终能够获取到同时满足无散度约束和粘性约束的速度场。流体的粘性是由粘性约束条件中的系数 μ 来控制的, 因此迭代过程并不会改变流体的粘性。在图 3-9 的测试中, 不同迭代次数下流体卷绳效应的形态也是基本相同的, 这也说明流体的粘性系数并没有随着迭代次数发生较大的变化^[121]。

3.4 测试与讨论

为进一步测试本文 SIMPLE 算法的特性, 本章节使用该方法模拟了多种形态、性质的粘性流体。所有测试场景都是在安装有 Intel I7-8700K 与 NVDIA Geforce RTX2080 GPU 的主机上模拟计算的。其中邻域查找计算、不可压缩性计

算、粘性计算等代码实现都使用了 GPU 加速策略以提高模拟的效率。测试中所使用的表面张力计算是基于 He 等人^[104]提出的方法实现的，流体的表面网格重构则使用了 Bhattacharya 等人的方法^[147]。各测试场景中的粒子间距均为 0.005m，平滑距离为 0.0125m。为了简化流体与固体间交互的处理方法，固体边界使用了有向距离场来表示。当检测到流体粒子穿入到固体中时，直接将粒子的位置修正回固体表面附近位置，并耗散掉粒子法向和切向方向的部分速度，进而实现流体与固体之间的碰撞^[52]。

3.4.1 与其它方法的对比

当大粘性流体材料在重力作用下以薄层形态垂直地流到平面上时，会产生另一种特有的效应——压曲效应^[148]。使用本文方法模拟的粘性流体的压曲效应如图 3-13(d)、(e)、(f)所示，模拟结果展示出本文方法只要使用 5 次迭代就能够显著提高流体压曲效应的视觉质量，而将迭代次数提高到 10 次，则视觉质量的提升已不再明显。若将本文的 SIMPLE 迭代算法中基于压强泊松方程的不可压缩性求解替换为 DFSPH 法并模拟相同的场景（图 3-13(a)、(b)、(c)），可观察到由于 DFSPH 与隐式粘性^[122]之间的迭代不收敛，流体压曲效应模拟的稳定性没有改善，而且流体的褶皱数量也有所减少。

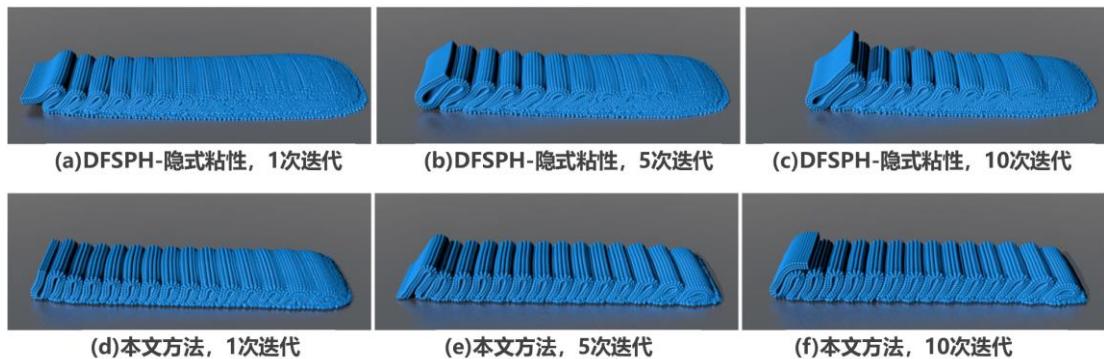


图 3-13 流体压曲效应的模拟。(a)~(c)本文 SIMPLE 迭代算法中的不可压缩性求解器被替换为 DFSPH 法^[90, 97]；(d)~(f)使用本文方法在不同迭代次数下模拟的压曲效应。

Figure 3-13 Comparison of the buckling effect.

在本文研究提出之前，也有研究者尝试减轻 SPH 流体不可压缩性与粘性之间的冲突，即：Peer 等人提出的隐式粘性求解方法^[123, 143]。该方法通过重建流体速度场的方式来同时求解不可压缩性与粘性的作用，可以避免两求解步骤间的冲突。然而，Weiler 等人的研究指出^[122]，这种重建速度场的方法会引入严重的数值耗散。为了验证 Peer 等人方法与本文方法在大粘性流体模拟方面的区别，图 3-14 中模拟了粘性流体的卷绳效应。模拟的结果显示 Peer 等人的方法由于存在严重的数值耗散，因此无法正确的模拟流体的卷绳效应（图 3-14(a)与(b))。将本文方法中不可压缩性计算方法替换为 DFSPH^[97]法后，只执行一次迭代时虽然能

够较为真实地模拟流体的卷绳效应，但流体会快速地“融化”掉，无法稳定地维持流体的形态（图 3-14(c)）；但将迭代次数设置为 5 次，原本正确的卷绳效应就会被破坏掉（图 3-14(d)）。使用本文方法模拟的流体如图 3-14(e)和(f)所示，当迭代次数设置为 5 次时，流体的表面细节会更为明显，而且形态也较为稳定。

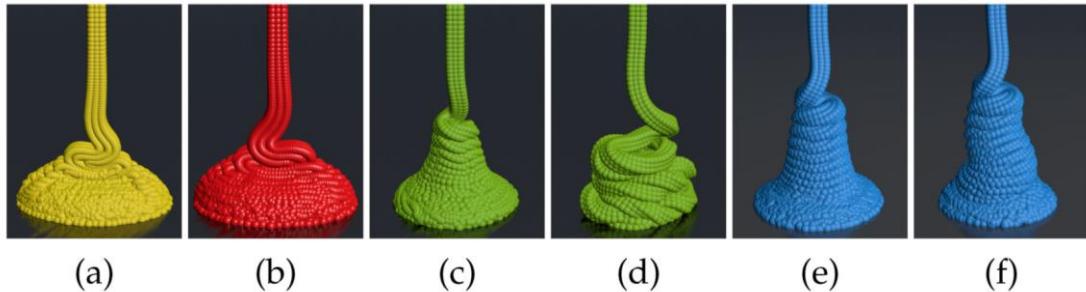


图 3-14 流体卷绳效应的模拟。(a)Peer 等人 2015 年方法^[143]; (b)Peer 等人 2017 方法^[123]; (c)本文算法中不可压缩性算法替换为 DFSPH 法^[109]并迭代 1 次; (d) 本文算法中不可压缩性算法替换为 DFSPH 法^[109] 并迭代 5 次; (e)本文方法迭代 1 次; (f) 本文方法迭代 5 次。

Figure 3-14 Comparison of rope coiling effect.

3.4.2 牛顿流体的模拟

为验证本文 SMIPLE 算法的稳定性，本节测试了不同粘性系数以及不同粒子规模下的牛顿流体运动。牛顿流体是粘性系数不会发生变化的流体材料，如图 3-15 中模拟的三种不同粘性的牛顿流体。该组测试中的内迭代收敛阈值为 10^{-2} 倍的最大误差，时间步长为 0.001s。由于图 3-15 (a)中粘性系数设置为 0，不存在粘性与不可压缩性之间的冲突，因此该项测试的 SIMPLE 迭代次数被设置为 1 次；而在图 3-15 (b)与(c)中流体的 SIMPLE 迭代次数为 2 次。测试结果显示，即使模拟的是无粘性流体或是粘性系数极大的流体，本文方法均没有发生严重的不稳定问题，而且在流体处于拉伸的状态时，模拟也能相对稳定的进行。图 3-16 模拟了不同粘性流体材料之间的耦合，该组测试中 SIMPLE 迭代次数选择为 2 次，内迭代收敛阈值为 10^{-2} 的最大误差，时间步长为 0.001s。蓝色、橙色、绿色的流体材料粘性分别为 $3Pa \cdot s$ 、 $100Pa \cdot s$ 、 $5 \times 10^6 Pa \cdot s$ ，粒子数量均为 13.6 万。尽管该场景中的不同流体粘性相差较大，但不同流体间的接触面上并没有产生明显的错误。

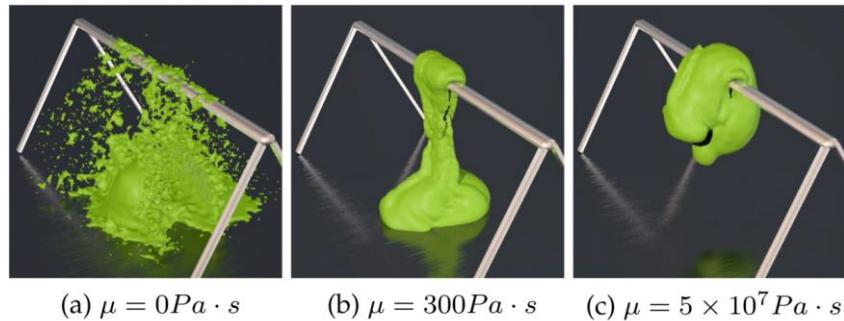


图 3-15 牛顿流体。 (a)粘性系数为 0 的流体; (b)粘性系数为 $300 \text{ Pa} \cdot \text{s}$ 的流体; (c)粘性系数为 $5 \times 10^7 \text{ Pa} \cdot \text{s}$ 的流体。

Figure 3-15 Newtonian fluids.

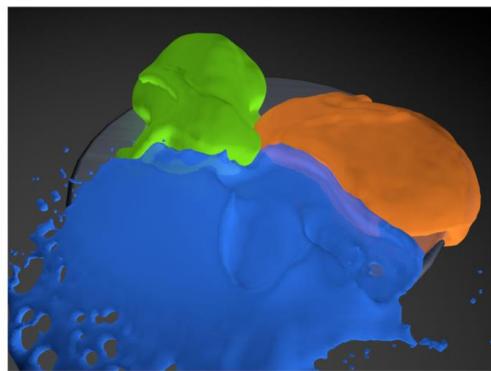


图 3-16 不同粘性的牛顿流体材料间的碰撞。

Figure 3-16 Collision of Newtonian fluids with different viscosity.

为进一步验证本文方法在较大规模场景中的稳定性，图 3-17 模拟了包含有约一百万个粒子的流体。该组测试中 SIMPLE 迭代次数选择为 4 次，粘性系数为 $400 \text{ Pa} \cdot \text{s}$ ，内迭代收敛阈值为最大误差的 10^{-2} 倍，时间步长为 $1.2 \text{ ms} \sim 1.9 \text{ ms}$ 。该场景中有 5 个具有复杂形态、由大粘性流体构成的犰狳逐个坠落到地面。模拟过程中没有产生明显的不稳定问题。而且由于 SIMPLE 迭代减弱了粘性与不可压缩性间的冲突，流体保持有较为锐利、丰富的表面细节。

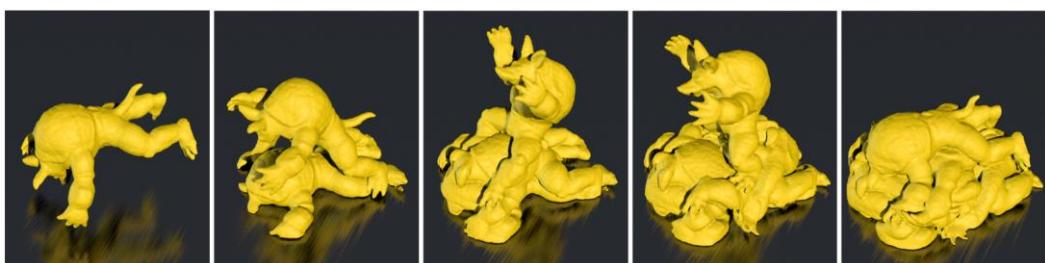


图 3-17 包含有 100 万个粒子的大粘性流体。
Figure 3-17 High viscosity fluid with 1 million particles.

3.4.3 非牛顿流体的模拟

与牛顿流体材料不同，非牛顿流体的形变速率与剪切应力之间不存在线性关系，也即是非牛顿流体材料的粘性系数并非为恒定值。由于这类材料在模拟过程中粘性系数会发生快速、剧烈的变化，因此模拟过程中很容易产生不稳定。为了进一步评估本文方法的稳定性，同时将本文方法扩展到非牛顿流体仿真的范围中，本文引入了 Cross 模型^[149]，并基于本文方法构建了非牛顿流体材料的仿真方法。Cross 模型下的粘性系数可通过如下非线性方程计算：

$$\mu = \mu_\infty + \frac{\mu_0 - \mu_\infty}{1 + (k\dot{\gamma})^n} \quad (3-27)$$

其中， μ_0 和 μ_∞ 分别为材料粘性变化范围的下限和上限；材料参数 k 控制了材料粘性变化对于剪切形变速率的敏感性； n 决定了材料粘性随剪切形变速率变化的曲率； $\dot{\gamma}$ 是与流体剪切形变速率相关的参数，其计算式如下：

$$\dot{\gamma} = \sqrt{0.5 \operatorname{trace}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)^2} \quad (3-28)$$

如果相邻流体粒子的粘性系数差别较大，会导致严重的动量不守恒问题，因此本文将式(3-6)中的剪切应力项进行了平滑处理，即：

$$\mu \nabla_i \cdot \nabla \mathbf{u}' = 2(d+2) \sum_{j^l} V_{j^l} \left(\frac{\mu_i + \mu_{j^l}}{2} \right) \left(\frac{\mathbf{u}'_{ij^l} \cdot \mathbf{x}_{ij^l}}{|\mathbf{x}_{ij^l}|^2 + \eta^2} \right) \nabla_i W_{ij} \quad (3-29)$$

将本文的 SIMPLE 迭代算法中的粘性计算替换为上式后，原方法即可实现 Cross 模型非牛顿流体材料的模拟。当式(3-27)中 n 取值大于零时，使用上述方法模拟的材料为剪切增稠流体；当取值小于零时，模拟的材料即为剪切变稀流体；当 n 取值为零时，上述方法就退化为了非牛顿流体材料的仿真方法^[149]。基于式(3-27)可绘制出三种材料的粘性系数变化曲线，如图 3-18 所示。由该图可知，当流体产生了较大的形变速率时，剪切变稀流体会因粘性突然降低而突然变“软”，而剪切增稠流体则正相反，相同情况下会突然变“硬”。

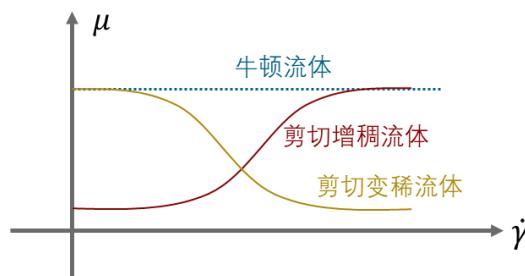


图 3-18 Cross 模型^[149]中剪切变稀、剪切增稠、牛顿流体材料的粘性变化曲线。

Figure 3-18 Viscosity curves for shear-thinning, shear-thickening and Newtonian fluids in the cross model.

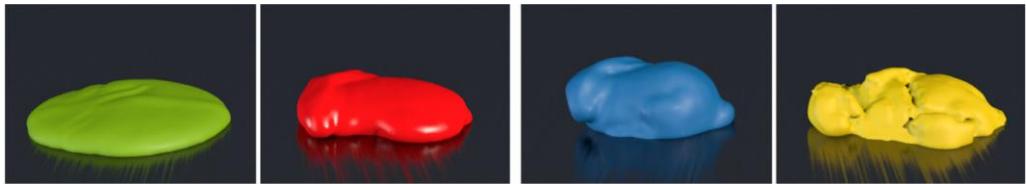


图 3-19 牛顿流体与非牛顿流体的对比。由左至右分别为低粘性牛顿流体、剪切增稠流体、高粘性流体、剪切变稀流体。

Figure 3-19 Newtonian versus non-Newtonian fluids.

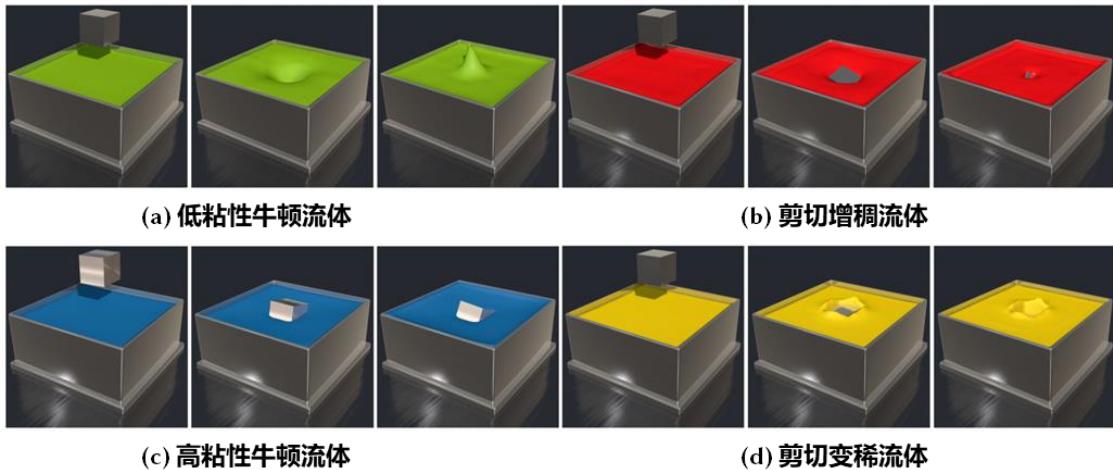


图 3-20 固体方块掉落到牛顿流体与非牛顿流体中。

Figure 3-20 A solid cubic falls into a Newtonian or non-Newtonian fluid.

图 3-18 使用本文方法模拟对比了非牛顿流体与牛顿流体。图中绿色流体为粘性系数 $50Pa \cdot s$ 的牛顿流体，红色流体为剪切增稠流体($\mu_0 = 50Pa \cdot s$ 、 $\mu_\infty = 5 \times 10^5 Pa \cdot s$ 、 $k = 10$ 、 $n = 6$)，蓝色流体为粘性系数 $5 \times 10^5 Pa \cdot s$ 的牛顿流体，黄色流体为剪切变稀流体($\mu_0 = 50Pa \cdot s$ 、 $\mu_\infty = 5 \times 10^5 Pa \cdot s$ 、 $k = 10$ 、 $n = -6$)。当流体在坠落到平面上之前，绿色的牛顿流体与红色的剪切增稠流体有相同的粘性系数，即： $\mu_0 = 50Pa \cdot s$ 。而当与平面碰撞产生形变时，红色的剪切增稠流体的粘性系数会突然变大，待形变速率减小后才缓慢的“融化”。类似地，在坠落之前蓝色的牛顿流体与黄色的剪切增稠流体的粘性系数均为 $\mu_0 = 5 \times 10^5 Pa \cdot s$ ，而在与平面碰撞后，黄色的剪切变稀流体的局部位置因形变速率过大而突然变“软”，随即发生碎裂。为进一步展示牛顿流体与非牛顿流体的区别，图 3-20 模拟了一个固体方块掉落到不同流体中的场景，其中(a)图为粘性系数 $5Pa \cdot s$ 的牛顿流体；(b)图为剪切增稠流体($\mu_0 = 5Pa \cdot s$ 、 $\mu_\infty = 5 \times 10^5 Pa \cdot s$ 、 $k = 1$ 、 $n = 6$)；(c)图为粘性系数 $5 \times 10^5 Pa \cdot s$ 的牛顿流体；(d)图为剪切变稀流体($\mu_0 = 5Pa \cdot s$ 、 $\mu_\infty = 5 \times 10^5 Pa \cdot s$ 、 $k = 50$ 、 $n = -10$)。固体落入到低粘性牛顿流体中时，固体会快速的下沉(图 3-20(a))；固体落在剪切增稠流体中时，由于流体的粘性突然变大，固体会突然停顿在流体表面，然后再缓缓沉入到流体中(图 3-20(b))；对于

大粘性牛顿流体场景(图 3-20(c)), 固体会需要很长的时间才能完全沉入流体中; 固体坠入剪切变稀流体中时(图 3-20(d)), 与固体相接处位置的流体粘性会突然变得很低, 因此固体会快速沉入流体中并在流体上留下深坑。待固体完全沉入流体后, 流体的形变速率会降低, 此时流体的粘性又会恢复为较大值, 导致深坑保持存在很久。

3.4.4 果酱、黄油、奶油的模拟

为验证本文 SIMPLE 算法在机算计图形学领域的应用潜力, 本节将模拟人们生活中常见的粘性不可压缩性流体材料, 如果酱、黄油、奶油等。

在图 3-21 中, 分别使用 1 次 SIMPLE 迭代和 5 次 SIMPLE 迭代模拟了曲奇饼干上的奶油。该测试中流体粘性系数为 $5 \times 10^6 \text{ Pa} \cdot \text{s}$, 内迭代收敛阈值为最大误差的 10^{-5} , 时间步长为 1 ms。使用 1 次迭代模拟的奶油会很快融化掉, 视觉上比较不真实。而使用 5 次迭代方法模拟的奶油, 由于粘性与不可压缩性之间的冲突被抑制, 奶油的形态更加稳定, 而且会产生更明显的压曲效应。

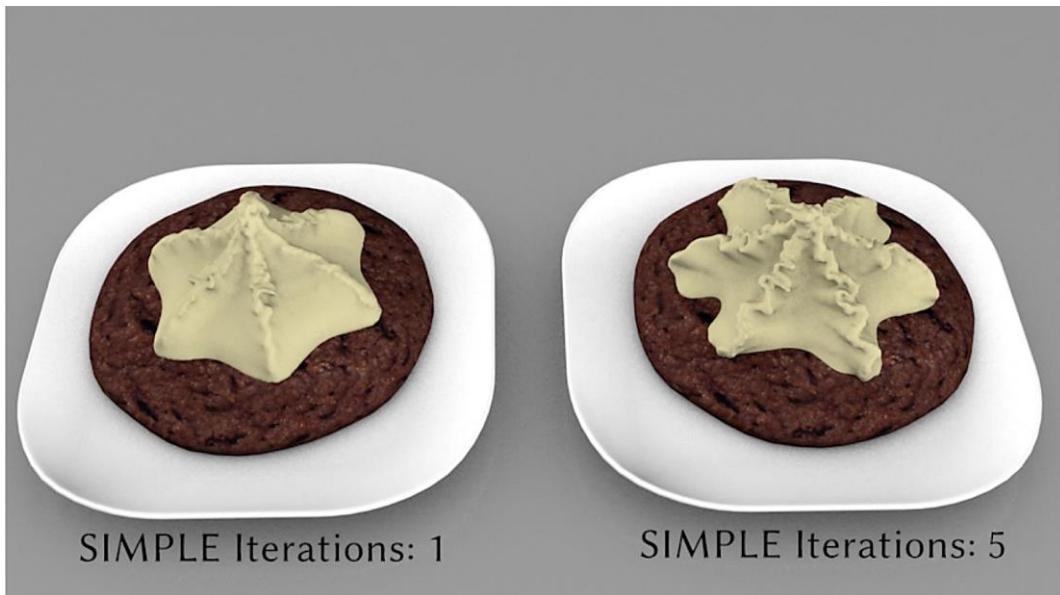


图 3-21 曲奇饼干上的奶油。

Figure 3-21 Creams on cookies.

图 3-22 中模拟了面包上黄油的压曲效应。该测试将黄油视为牛顿流体, 其粒子数量为 10 万, SIMPLE 迭代次数为 5 次, 粘性系数为 $400 \text{ Pa} \cdot \text{s}$, 粒子间距为 0.005m, 平滑距离为 0.0125m, 内迭代收敛阈值为最大误差的 10^{-2} , 时间步长为 2ms, 无表面张力作用。得益于本文的 SIMPLE 迭代方法, 即使是使用了大到 1ms 的时间步长, 黄油依然能够较稳定地保持形态。

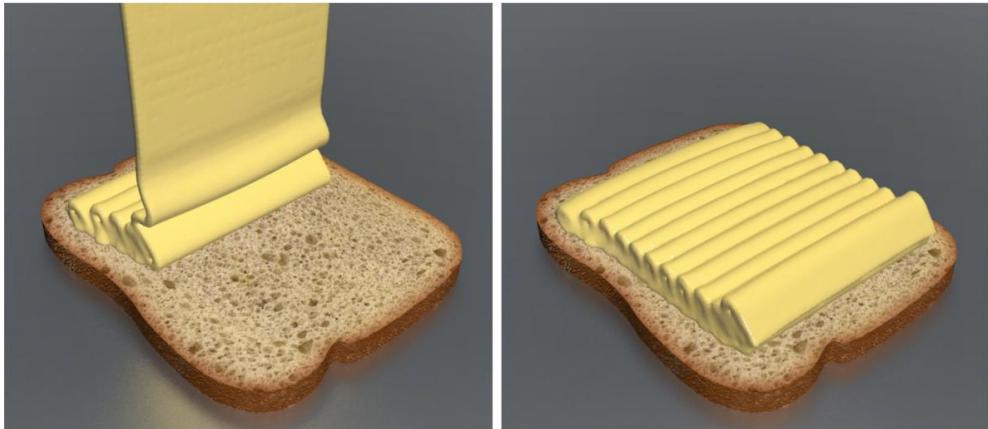


图 3-22 黄油的压曲效应模拟。
Figure 3-22 The buckling effect of the butter.

图 3-23 中模拟了坠落在长凳上的果酱，该场景中包含有 46.2 万个粒子。由于果酱一般具有较强的表面张力，因此该场景在原算法中增加了表面张力计算步骤^[104]。该组测试中 SIMPLE 迭代次数选择为 2 次；粘性系数为 $400 \text{ Pa} \cdot \text{s}$ ；内迭代收敛阈值为最大误差的 10^{-2} ；时间步长为 $1.4 \text{ ms} \sim 2.2 \text{ ms}$ ；表面张力^[104]强度系数为 0.002。同一测试数据中分别渲染为表面网格（左图）和粒子（右图）。本文方法在果酱处于拉伸状态下时依然能够较稳定地仿真，而且能够模拟出流体的薄层、拉丝等形态。此外，该场景中还出现了明显的卷绳效应和压曲效应。

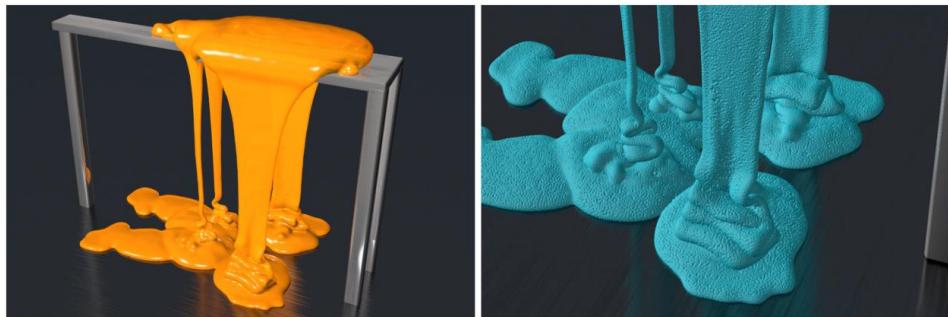


图 3-23 果酱。
Figure 3-23 Jam.

图 3-24 中使用本文方法模拟了不同形态、性质的奶油和果酱。该场景中包含有多种不同的粘性流体材料，所有流体均进行了 5 次 SIMPLE 迭代。其中橘黄色的奶油（左）被施加了表面张力作用，粘性系数为 $300 \text{ Pa} \cdot \text{s}$ ；处于拉伸状态的巧克力酱粘性为 $300 \text{ Pa} \cdot \text{s}$ ；白色的奶油（中上）展示了粘性流体的卷绳效应，其粘性为 $5 \times 10^5 \text{ Pa} \cdot \text{s}$ ；暗黄色的奶油（右）展示了粘性流体的压曲效应，其粘性为 $5 \times 10^4 \text{ Pa} \cdot \text{s}$ 。即使使用较大的时间步长（1ms），本文方法依然能够稳定地模拟具有大表面张力的稀奶油、果酱、巧克力酱，同时也能够模拟具有锐利表面细节、稳定形态的奶油花。



图 3-24 不同形态奶油的模拟。
Figure 3-24 Creams and jams simulation.

3.5 本章总结

本章首次提出了适用于 SPH 法的 SIMPLE 迭代算法。该方法旨在抑制 SPH 流体的压强与剪切应力之间的冲突，进而提高粘性不可压缩性流体的仿真稳定性。本章在不同的测试场景中验证了该迭代算法的收敛性，同时也证明了该方法能够显著提高粘性流体模拟的视觉质量。相比其它方法，新的方法即使使用了较大时间步长，也可以较为真实地模拟粘性流体的卷绳效应、压曲效应，较好地保持流体表面锐利的细节，同时能够较为稳定地模拟粘性流体的拉伸状态。此外，本章的研究中也提出了可稳定模拟 Cross 模型非牛顿流体的方法。

第 4 章 对偶粒子投影法

4.1 引言

SPH 法自被提出以来, 其拉伸不稳定问题就成为了限制其应用的巨大障碍之一。拉伸不稳定问题会导致处于拉伸状态下的材料发生结块、断裂等非常严重的错误, 使 SPH 法无法真实地模拟流体的薄层、细流、飞溅等小尺度细节^[12]。为抑制 SPH 流体的拉伸不稳定问题, 目前计算机图形学领域研究中常采用的策略主要有如下两种:

第一种策略是设置背景压强以避免粒子间吸附力的出现。如果流体具有一个取值为正的背景压强, 将流体各位置的压强整体抬高, 那么流体中就可以不出现负压强, 粒子之间只会存在排斥力而不存在会引起粒子结块的吸附力^[95, 97]。虽然这种策略能够使粒子分布相对均匀, 但这种消除负压强的方法同时也存在着以下几方面的问题: ①这类方法常需要调整背景压强这种非物理的参数, 以避免在流体在仿真中发生不稳定^[48]; ②由于压强泊松方程需要设置合适的狄雷克利边界条件, 因此这种方法不适于在投影法中使用^[104]; ③负压强的缺失会导致 SPH 法很难模拟流体很多小尺度的细节, 比如细流、薄层等^[91]。

第二种策略是使用类似于有限差分法的压强梯度算子^[102]。这种经过修正的压强梯度算子能够修正粒子所受到的压强力, 减小粒子运动过程中产生的压强扰动, 有效减弱粒子的结块等问题。然而, 当粒子距离过近时, 粒子间的斥力会过大, 影响仿真的稳定性。此外, 这一方法并不能从根本上解决拉伸不稳定问题, 导致其在提升流体小尺度细节的仿真稳定性方面作用很有限。

SPH 法是拉格朗日视角下的仿真方法, 但却使用了基于欧拉视角的核函数进行插值, 这一冲突是 SPH 法产生拉伸不稳定问题的根本原因^[150]。在过去 SPH 固体模拟方面的研究中, Belytschko 与 Xiao 等人提出了能够完全消除 SPH 拉伸不稳定问题的方法, 该方法使用了一组额外的粒子作为应力场的采样点, 同时其核函数被定义在了材料的物质坐标上^[150], 可以完全避免 SPH 拉伸不稳定问题的产生。但这种采用物质坐标、拉格朗日视角核函数的方法是无法模拟大形变流体材料的。因此, 如何使用欧拉视角的核函数来抑制流体拉伸不稳定问题, 是本章研究的重点。针对这一问题, 本文提出了用于抑制 SPH 流体拉伸不稳定问题的对偶粒子架构(图 4-1)。该对偶粒子架构包含有两种不同的粒子, 一种是与标准 SPH 粒子类似的粒子, 这种粒子携带有质量、速度等信息; 另一种是作为压强采样点的粒子。在本文后续讨论中, 这种带有质量、速度的粒子将被称为“实粒子”, 而携带有压强的粒子则被称为“虚粒子”。在对偶粒子架构下, 原流体 NS 方程相应地会使用 SPH 核函数与对偶粒子进行离散化。由于虚粒子不携带有流体的质量属性, 因此可以采用比实粒子更加均匀的方式来生成虚粒子, 提高模拟的稳

定性。而且引入虚粒子可以缩小 SPH 法在拉伸状态下的不稳定区间，提高流体在拉伸状态下的稳定性。相比其它类型的 SPH 方法，本文的对偶粒子方法简单、易于编程实现，而且无需调节非物理参数即可有效抑制拉伸不稳定问题的影响。

本章研究的理论贡献主要包含以下几个方面：

- (1) 提出了一种可用于抑制 SPH 拉伸不稳定问题的对偶粒子方法；
- (2) 改进了原 SPH 近似投影法，使得压强泊松方程中的压强和速度采样点即使不定义在相同位置上也能够实现流体不可压缩性；
- (3) 验证了虚粒子不同分布对于改善流体仿真稳定性的影响，当虚粒子以均匀、等间距、正交的方式排布时，能够显著提高流体的拉伸稳定性。因此本文提出了基于 GPU 的空间自适应虚粒子快速生成方法，以高效高质量地实现稳定的流体仿真。

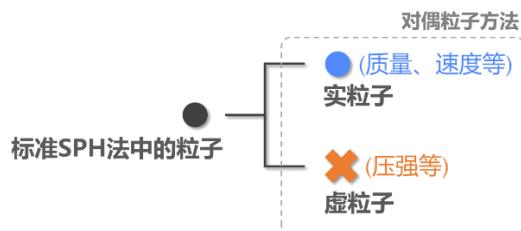


图 4-1 对偶粒子方法。

Figure 4-1 The dual-particle approach.

4.2 SPH 流体的拉伸不稳定

参考 swegle 等人的研究^[151]，SPH 法的拉伸不稳定问题是粒子负压强（负应力）与二阶微分核函数符号之间冲突引起的。这一冲突的存在导致粒子间距离在减小的同时，粒子之间的吸附力也会显著增大。因此处于拉伸状态下的材料，其粒子总会被临近的粒子吸附，导致模拟的材料发生非物理地断裂。如图 4-2 所示，在一维空间中有两个相互远离的粒子 A 与粒子 C，此时位于两粒子中间位置的 B 粒子会同时受到粒子 A 与粒子 C 的吸附力。只要粒子 B 不是精确地位于粒子 A 与粒子 C 的正中间，那么与粒子 B 离得更近的粒子 A 就会对粒子 B 有更大的吸附力，并导致粒子 B 跟随粒子 A 一起运动，造成该组粒子所代表材料断裂。

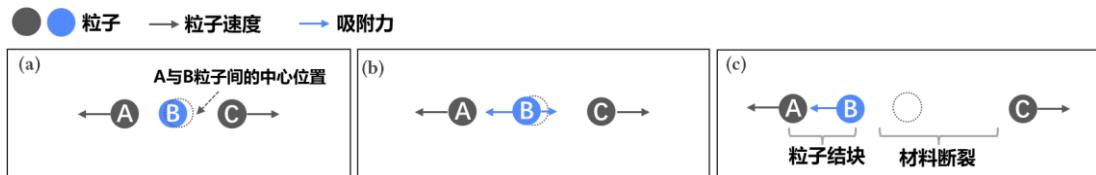


图 4-2 一维空间中的粒子拉伸不稳定体展示。

Figure 4-2 Demonstration of tensile instability in 1-d space.

目前，在抑制 SPH 法的拉伸不稳定方面已有较多的研究。在计算机图形学领域中，最常见的方法即是避免粒子间吸附力的产生，比如直接将粒子的负压强置零^[95, 97]，但这会直接导致流体很多运动细节消失。另一种方法是设置背景压强^[129, 152]，以避免负压强的出现，但这种方法需要根据仿真的场景人工去调节背景压强的取值，不仅不便于使用，还容易因参数不合适导致仿真不稳定；除此之外，这种设置背景压强的方法会使粒子间产生过多的排斥力，影响仿真的精确性^[48]。近些年在工程领域的研究中，设置背景压强方法被进一步改善，改善后的方法被称为“拉伸不稳定控制法（Tension instability control）^[153-155]”。该方法通过在仿真算法中添加一些用于控制背景压强的“开关”，使流体不同位置的背景压强可以取不同值，有效抑制了 SPH 流体的拉伸不稳定问题。然而，这种不平滑的非物理背景压强会引入其它的不稳定问题，比如粒子震荡等，因此需要与其它用于提高 SPH 法稳定性的算法结合使用，比如粒子位置修正（Particle shifting）技术^[156]等。另一种用于抑制拉伸不稳定问题的方法是梯度修正方法，这种方法最早由 Khayyer 与 Gotoh 提出^[146]，后被 He 等人引入到图形学领域中^[91, 102]，但这种方法不能消除负压强与核函数不匹配的问题^[151]，因此对于改善拉伸不稳定的作用较为有限，不能显著提高流体小尺度运动细节的视觉质量。

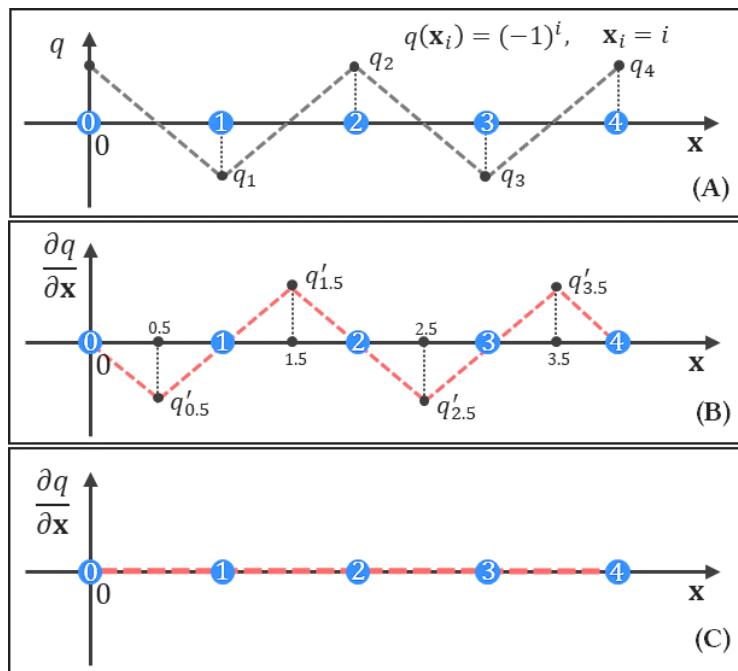


图 4-3 零能问题。(A)一维空间中一组携带属性 q 的粒子；(B)在粒子之间位置上可计算出正确的 q 的微分；(C)如果在粒子上计算 q 的微分，会得到错误的结果。

Figure 4-3 Zero-energy problem.

零能问题是恶化拉伸不稳定问题的另一种数值计算不稳定问题，导致零能问题的原因是流体的物理场及其微分场均被定义在了同一组粒子上，而这会导致微分场的计算错误（如图 4-3 所示）。若在 SPH 粒子之外引入一组额外的采样点用

于定义应力即可有效消除零能问题, 相应地这一方法也就是应力点(stress-point)SPH 方法^[47, 157]。但由于这类方法中, 应力点与用于存储速度的粒子需要错列排布, 而在模拟流体时很容易破坏这一排布方式, 因此应力点 SPH 法目前尚不能用于模拟流体。

参考 Belytschko 与 Xiao 等人的研究^[150], 拉伸不稳定问题来自于拉格朗日视角下的粒子与欧拉视角下的核函数之间的不匹配。若 SPH 法能够避免使用欧拉视角的核函数, 则可以避免拉伸不稳定问题的出现。因此采用物质坐标的拉格朗日核函数方法以及纯拉格朗日公式化的 SPH 方法(Total Lagrangian Formalism)^[158-161]可以避免 SPH 法的拉伸不稳定问题。但是这种使用物质坐标的核函数以及插值计算方法, 很难用于模拟具有大形变速率的流体材料。

直接处理拉伸不稳定问题较为困难, 因此计算机图形学领域中有部分研究绕过了 SPH 法的拉伸不稳定问题, 尝试增强流体的小尺度运动细节模拟的视觉质量: 为提高流体表面附近粒子的密度计算精确性, Schechter 与 Bridson 等人提出了在流体表面附近布设幽灵气体粒子的方法^[128]; 为了提高流体的薄层、拉丝等细节, He 等人提出了使用两种不同尺度下的核函数来计算压强, 并引入各项异性核函数来稳定流体的压强分布^[104]。此外, 由于表面张力可以最小化流体的表面积, 并增强流体溅射等现象模拟的视觉质量, 因此可以一定程度降低拉伸不稳定问题对流体行为的影响^[104, 162-164]。然而, 上述研究由于没能触及到拉伸不稳定问题的本质, 因此相关方法对于改善流体小尺度行为的仿真稳定性方面作用有限。

4.3 对偶粒子框架

为引出本文抑制拉伸不稳定方法的研究, 本节将首先简要论述不可压缩 SPH 流体的基础理论及其拉伸不稳定问题的原因, 然后将阐述本文提出的对偶粒子框架的研究动机与原理。

4.3.1 SPH 流体与其拉伸不稳定问题

流体的不可压缩性可以通过求解压强的泊松方程来施加, 即:

$$\nabla_i \cdot \frac{\delta t}{\rho} \nabla p = \nabla_i \cdot \mathbf{u}' \quad (4-1)$$

将上述方程离散化, 并施加合适的自由表面边界条件, 可求解出能够消除流体速度散度误差的压强场。然后通过动量方程(下式)就能获得满足速度无散度条件的粒子速度。

$$\mathbf{u}^{n+1} = \mathbf{u}' - \frac{\delta t}{\rho} \nabla p \quad (4-2)$$

理想情况下，当流体处于拉伸状态时，流体的速度散度会为负值，此时根据上述压强泊松方程计算获得压强取值也可能为负，相应地流体负压强会表现为流体粒子间的吸附力，阻止流体被拉伸，并使流体形成拉丝、薄层等特别的形态。然而在 SPH 法中，该计算过程会产生严重的数值不稳定问题，即：随着粒子的相互靠近，粒子间的吸附力也会快速地增大。这一问题导致流体无法稳定地处于拉伸状态，粒子会发生结块，因此流体的拉丝、薄层等形态也难以维持。

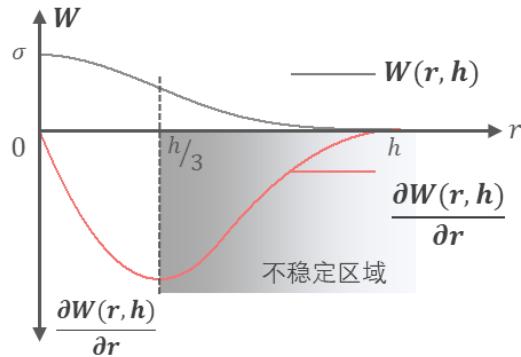


图 4-4 SPH 法中的核函数及其一阶微分。

Figure 4-4 Kernel function and its 1st-order differential in SPH method.

Swegle 等人对 SPH 法的拉伸不稳定问题产生的根本原因进行了论证^[151]，并给出了避免拉伸不稳定问题出现的充分条件。对于不可压缩性流体的求解算法，这一充分条件的形式为：

$$pW'' > 0 \quad (4-3)$$

上式中的 W'' 为 SPH 核函数的二阶微分。若 SPH 法中采用三次样条函数形式的核函数^[68]（图 4-4），则当粒子间距大于 $\frac{h}{3}$ (h 为 SPH 法的支持域半径) 时，核函数的二阶微分 W'' 为正，与此同时若粒子因处于拉伸状态导致其压强为负时， pW'' 的计算结果小于零，相应地拉伸不稳定问题就会产生。因此对应核函数所在的函数空间中粒子间距大于 $\frac{h}{3}$ 的位置是拉伸状态下的不稳定区域。为了更简要地展示、说明拉伸不稳定问题，图 4-5 展示了一维空间中处于拉伸状态下的 3 个粒子，该图中的场景假定粒子都具有相同的物理量，并且粒子只会受到吸附力的影响。在标准 SPH 法中，粒子同时携带着流体的速度和压强。如果粒子 i 精确地位于其两个邻域粒子的中间位置，那么当其邻域粒子以相反方向运动时，则粒子 i 因所受到的合力为零，会保持静止。但若粒子 i 的位置稍偏向右侧，那么由于粒子 i 位于其邻域两个粒子的不稳定区间，此时不满足式(4-3)中的充分条件，其右侧邻域粒子 $i+1$ 对其产生的吸附力 f_{i+1} 则会大于其左侧粒子对其产生的吸附力 f_{i-1} ，导致粒子 i 向右侧运动，远离粒子 $i-1$ 并与粒子 $i+1$ 聚集，拉伸不稳定问题也由此产生。

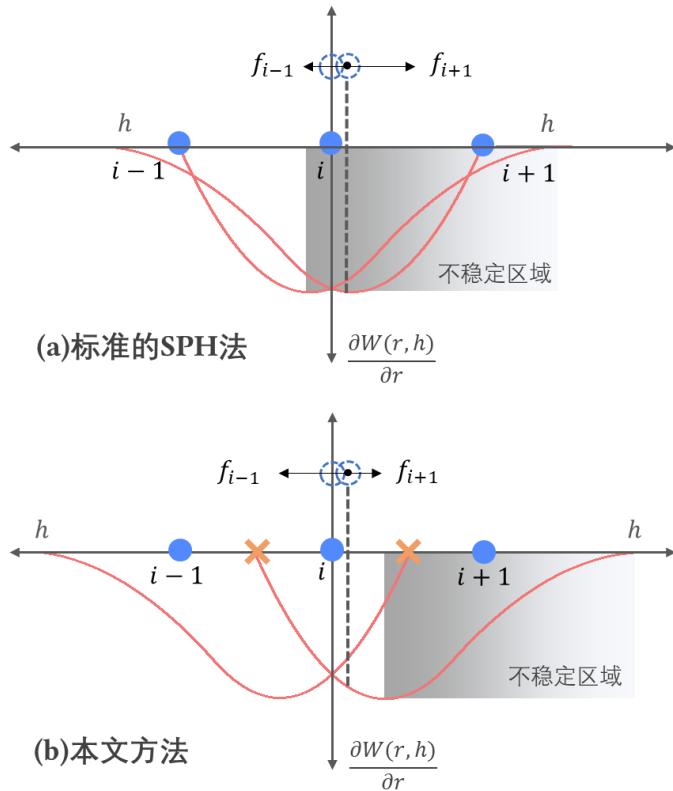


图 4-5 标准 SPH 法与本文对偶粒子方法的拉伸不稳定问题。

Figure 4-5 Demonstration of tensile instability in a standard SPH and our dual particle method.

4.3.2 使用对偶粒子缩小拉伸状态下的核函数不稳定区间

若要保证粒子不总是位于拉伸不稳定区间，以抑制拉伸不稳定问题，最简单的方法是增大支持域半径 h 。然而过大的支持域半径会导致流体的精确性降低，并平滑掉流体大量高频运动信息，此外也会显著提高仿真计算的效率^[165]。

受到应力点 SPH 法的启发^[47, 157]，本文研究中尝试引入了一组额外的粒子——虚粒子来作为流体压强的采样点。在使用改进方法之后，粒子 i 所受到吸附力则需要使用其邻域中的虚粒子来计算。根据图 4-5(b) 所示，当粒子 i 偏向于右侧时，仍位于虚粒子核函数的稳定区间，因此尽管粒子 i 更偏向于粒子 $i + 1$ ，但其受到粒子 $i - 1$ 的吸附力会大于来自粒子 $i + 1$ 的吸附力，会将粒子 i 拉回到中心位置，减弱粒子发生聚集的趋势。因此只要选择合适的核函数，该方法就可以有效地抑制拉伸不稳定问题。此外这种不将速度和压强定义在相同位置上的方法，也可以避免零能问题的产生^[166]，进一步提高流体仿真的稳定性。

4.3.3 对偶粒子的离散化方法

基于上文的讨论可知，对偶粒子方法中两种粒子携带有不同的物理量：实粒子携带流体的质量、速度等信息，其位置必须位于流体的内部或表面；虚粒子仅

仅作为流体所在空间中压强的采样点，因此其位置可以不限制在流体内部和表面上。在仿真过程中，为保证虚粒子能够跟随着流体运动，虚粒子需要在每一帧的仿真计算预先生成，并在当前帧结束计算之前被删除掉。

为使 SPH 法能够与对偶粒子框架适配，需要适当改进 SPH 法的离散化方案。基于 SPH 法的基本离散化方法^[68]，并引入对偶粒子后，物理场 q 可以使用如下的四个方程来计算：

$$q_i = \sum_j V_j q_j W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (4-4)$$

$$q_I = \sum_j V_j q_j W(\mathbf{x}_I - \mathbf{x}_j, H) \quad (4-5)$$

$$q_i = \sum_J V_J q_J W(\mathbf{x}_i - \mathbf{x}_J, h) \quad (4-6)$$

$$q_I = \sum_J V_J q_J W(\mathbf{x}_I - \mathbf{x}_J, h) \quad (4-7)$$

上述方程中所使用的下标 i 与 j 用于指代实粒子， I 与 J 指代虚粒子； H 与 h 分别为虚粒子和实粒子的支持域半径； V_j 与 V_J 分别为虚粒子和实粒子的体积。实粒子体积的计算式如下：

$$V_j = \sum_j \frac{m_j}{\rho_j} \quad (4-8)$$

为能够在虚粒子所在位置上相对正确地估计流体相关物理量，虚粒子只有位于流体内部或边界附近才会具有体积，因此虚粒子体积的计算式如下：

$$V_J = \sum_j V_j^2 W(\mathbf{x}_i - \mathbf{x}_J, h) \quad (4-9)$$

由上式可知，虚粒子的体积是通过实粒子体积的核函数插值获得。若虚粒子位于流体之外较远处，由于其邻域不包含有携带流体质量的实粒子，使用上式计算的虚粒子体积为零。

4.3.4 对偶粒子投影方法实现概述

参照 SPH 投影法^[99]，并基于上一节中讨论，使用对偶粒子的流体不可压缩性求解流程可以概括为如下的步骤：

- 第 1 步：根据实粒子的位置，生成虚粒子；
- 第 2 步：计算实粒子与虚粒子的体积；
- 第 3 步：使用实粒子与虚粒子离散流体的压强泊松方程，并施加合适的边界条件；
- 第 4 步：求解离散后的压强泊松方程；
- 第 5 步：计算每个实粒子所受到的压强力；
- 第 6 步：更新粒子的速度和位置，并删除虚粒子；

为了更直观地展示上述步骤, 图 4-6 演示了使用了对偶粒子的不可压缩性计算流程。

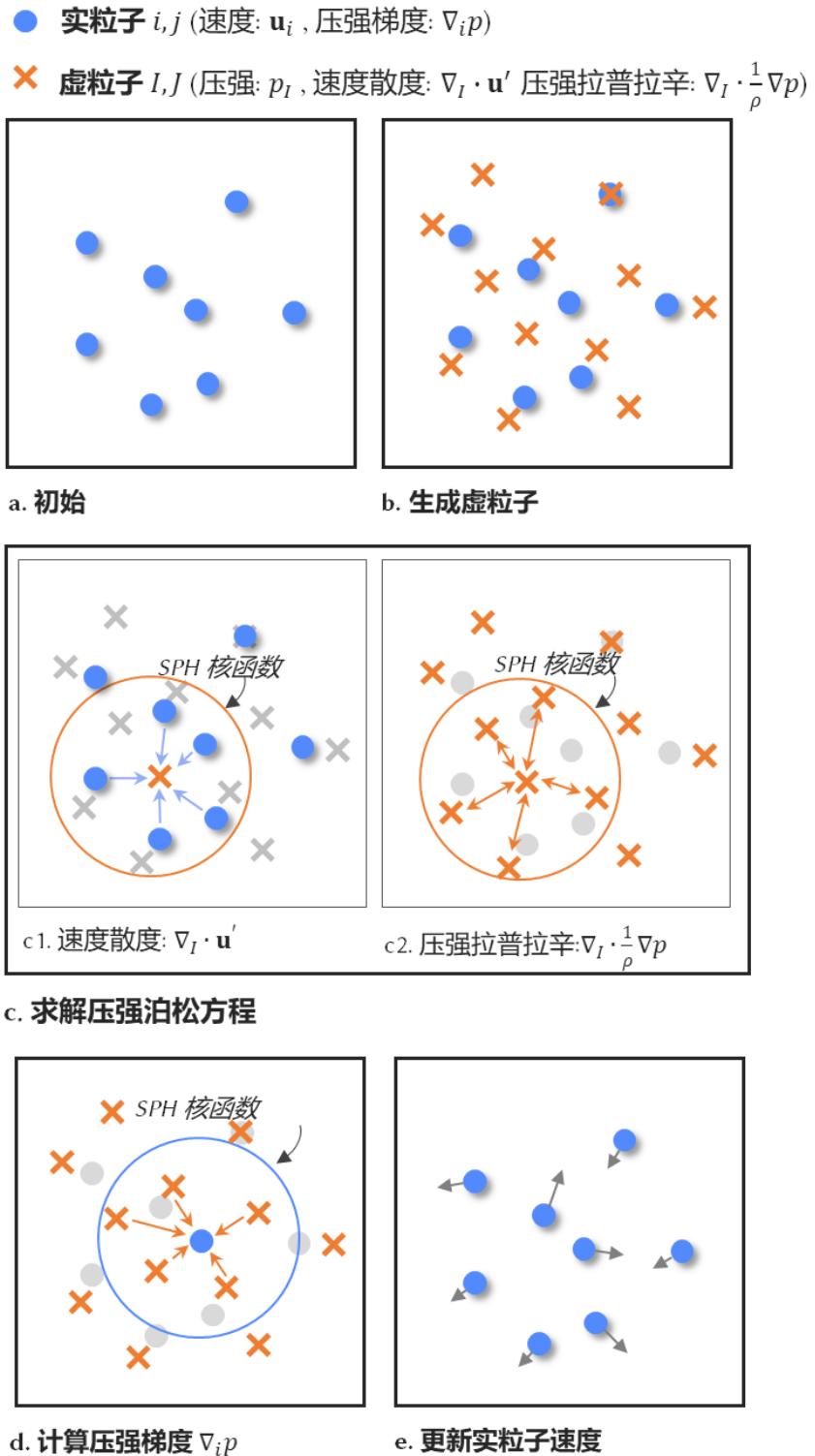


图 4-6 对偶粒子方法的计算步骤。

Figure 4-6 An overview of our dual-particle method.

4.4 使用对偶粒子的近似投影方法

本节将结合上节讨论的内容，详细讨论如何使用对偶粒子策略实现不可压缩性流体的投影计算，并确保投影计算过程能够与不同虚粒子采样策略结合使用。

4.4.1 压强的泊松方程离散化

基于投影法的不可压缩性求解，其核心步骤为压强泊松方程的求解过程。压强泊松方程中包含有两个微分算子，即：压强的拉普拉辛算子和速度的散度算子。在 SPH 法中，两种算子的离散方法较多，考虑到不可压算法的稳定性以及实现复杂性，本文选用了易于程序实现，且对 SPH 零能问题有较好抑制效果的近似投影法^[99]。

由于对偶粒子法中流体的压强采样点位于虚粒子上，因此流体的压强泊松方程也应在虚粒子位置上进行离散：

$$\nabla_I \cdot \frac{1}{\rho} \nabla p = \frac{1}{\delta t} \nabla_I \cdot \mathbf{u}^* \quad (4-10)$$

压强的拉普拉辛算子（上式的左端项）的离散化可以完全使用虚粒子实现，其具体形式如下：

$$\nabla_I \cdot \frac{1}{\rho} \nabla p = \frac{2}{\rho_0} \sum_J V_J \frac{P_I - P_J}{r_{IJ} + \eta} W'_{IJ} \quad (4-11)$$

上式中的 W'_{IJ} 为一阶微分核函数，其计算式为： $W'_{IJ} = \frac{\partial W(r_{IJ}, H)}{\partial r_{IJ}}$ ，此处假定流体的密度始终为流体静止时的密度 ρ_0 。采用类似的方式，式(4-11)的右端项（即压强泊松方程的源项）可以离散为如下的形式：

$$\nabla_I \cdot \mathbf{u}^* = - \sum_j V_j \mathbf{u}_j^* \nabla_I W_{Ij} \quad (4-12)$$

其中 V_j 为虚粒子 I 的邻域实粒子体积， $\nabla_I W_{Ij}$ 为核函数在 I 位置的梯度，即： $\nabla_I W_{Ij} = \frac{\mathbf{x}_{Ij}}{r_{Ij}} W'_{Ij}$ 。

经过上述的拉普拉辛算子以及速度散度的离散化后，原压强泊松方程就会转化为求解未知压强场的线性系统。然而若要使用该线性系统实现流体的仿真，有两个数值计算问题需要解决，分别是粒子邻域缺失问题以及密度漂移问题。以下是这两方面内容的讨论。

(1) 邻域缺失问题。如图 4-7 所示，当某虚粒子的邻域被流体表面截断后，可能会导致该粒子的邻域分布不对称，直接使用式(4-12)计算散度会引入严重的误差。若能够估计虚粒子位置的流体速度，参照 SPH 法中常用的散度计算公式（式(3-24)），则可以避免邻域缺失对散度计算的影响。参考 Chen 与 Beraun 的研究（Corrective SPH）^[167]，虚粒子位置上的速度可以通过如下方式计算获得：

$$\bar{\mathbf{u}}_I^* = \frac{\sum_j V_j \mathbf{u}_j^* W_{Ij}}{\sum_j V_j W_{Ij}} \quad (4-13)$$

然后使用虚粒子位置上的速度来修正散度计算:

$$\nabla_I \cdot \mathbf{u}^* = \sum_j V_j (\bar{\mathbf{u}}_I^* - \mathbf{u}_j^*) \cdot \nabla_I W_{Ij} \quad (4-14)$$



图 4-7 虚粒子的实粒子邻域缺失会导致其散度计算错误。

Figure 4-7 The deficiency of real particle neighbors of a virtual particle will lead to the error of its divergence calculation.

(2) 密度漂移问题。与 4.4.4 节中讨论的内容类似, 如果压强泊松方程不能直接消除流体的密度误差, 就会导致流体仿真的体积丢失。因此需要对压强泊松方程的速度散度源项进行密度补偿, 以避免这一问题的产生。在对偶粒子方法中, 密度补偿项的计算式如下:

$$\Lambda_I = \kappa \frac{\max(\rho_I - \rho_0, 0)}{\rho_0 \Delta t} \quad (4-15)$$

上式中的 κ 为常数; ρ_I 为虚粒子的密度, 其计算需要使用到虚粒子的实粒子邻域, 即: $\rho_I = \sum_j m_j W_{Ij}$ 。

4.4.2 自由表面边界条件

虽然在上一小节的讨论中, 已经将压强泊松方程转化为了线性系统, 但该线性系统是奇异的, 需要施加自由表面边界条件来保证该线性系统的对称正定特性。参照本文 3.3.3 的讨论, 此处可以使用半解析式边界条件来处理此问题^[91]。但由于对偶粒子方法中, 狄雷克利边界条件需要施加在虚粒子上, 因此相关的计算公式与本文 3.3.3 节的内容有一定差别。

首先, 原压强泊松方程中的压强拉普拉辛算子离散化公式可以进一步简化为下述形式:

$$\nabla_I \cdot \frac{1}{\rho} \nabla p = \hat{a}_I P_I - \sum_J \hat{a}_{IJ} P_J \quad (4-16)$$

其中, 系数 \hat{a}_I 与 \hat{a}_{IJ} 的计算式为:

$$\hat{a}_I = \sum_J \hat{a}_{IJ} \quad (4-17)$$

$$\hat{a}_{IJ} = \frac{2}{\rho_0} \sum_J V_J \frac{W'_{IJ}}{r_{IJ} + \eta} \quad (4-18)$$

上述公式中的 \hat{a}_I 为对应线性系统系数矩阵的对角元素，而 \hat{a}_{IJ} 代表非对角元素。如图 4-8(a)所示，当检测到虚粒子的系数 \hat{a}_I 小于阈值 a_0 ，则认为该虚粒子位于流体边界附近。而 a_0 可以设置为流体未处于压缩和拉伸状态下时，流体内部某虚粒子的系数 \hat{a}_I 的取值。若虚粒子位于流体边界附近，其系数必定小于 a_0 ，此时可以将其系数 \hat{a}_I 的取值直接替换为 a_0 ，即： $\hat{a}_I = \max(a_0, \hat{a}_I)$ ，进而实现流体的狄雷克利边界条件的施加，保证原线性系统系数矩阵的弱对角占优性。由于该线性系统的系数矩阵一般为对称正定矩阵，因此可以使用共轭梯度法快速求解出未知量，即虚粒子的压强。如图 4-8 所示，若流体处于压缩状态，某个位于流体内部并远离表面的虚粒子，根据其邻域计算的系数 \hat{a}_I 可能会大于 a_0 （图 4-8(a)），这种情况下该粒子的系数不会被修正为 a_0 ；若一个粒子的邻域被流体边界截断，则根据其邻域计算的系数 \hat{a}_I 会小于 a_0 （图 4-8(b)），这种情况该粒子的系数 \hat{a}_I 需要被设置为 a_0 。

上述半解析式自由表面边界条件不仅能够有效地在流体自由表面施加零压强条件（狄雷克利边界条件），而且实粒子的运动也可在该条件下表现的较为平滑，避免因虚粒子突然增删导致实粒子的运动出现错误。

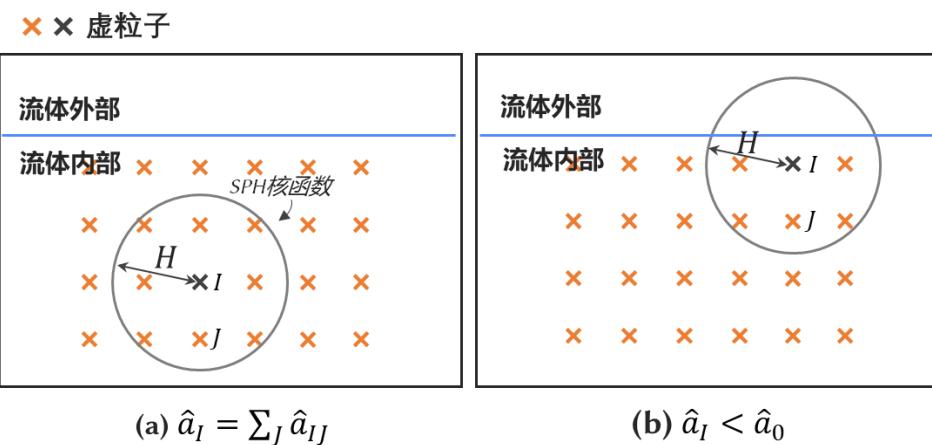


图 4-8 自由表面边界条件。

Figure 4-8 The free surface boundary condition.

4.4.3 实粒子速度的更新

在完成压强泊松方程的求解之后，定义在虚粒子上的压强会用于更新实粒子的速度。参考 SPH 法中梯度计算方法，实粒子位置上的压强梯度可以通过下式计算：

$$\nabla_i p = \sum_j V_j P_j \nabla_i W_{ij} \quad (4-19)$$

相应地，满足不可压缩性条件的实粒子的速度可以根据下式来更新：

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \frac{\delta t}{\rho_0} \nabla_i p \quad (4-20)$$

其中 \mathbf{u}_i^{n+1} 就是满足速度无散度条件的实粒子速度。使用该速度更新实粒子位置后，即可完成不可压缩性的计算。

4.5 虚粒子的生成方法

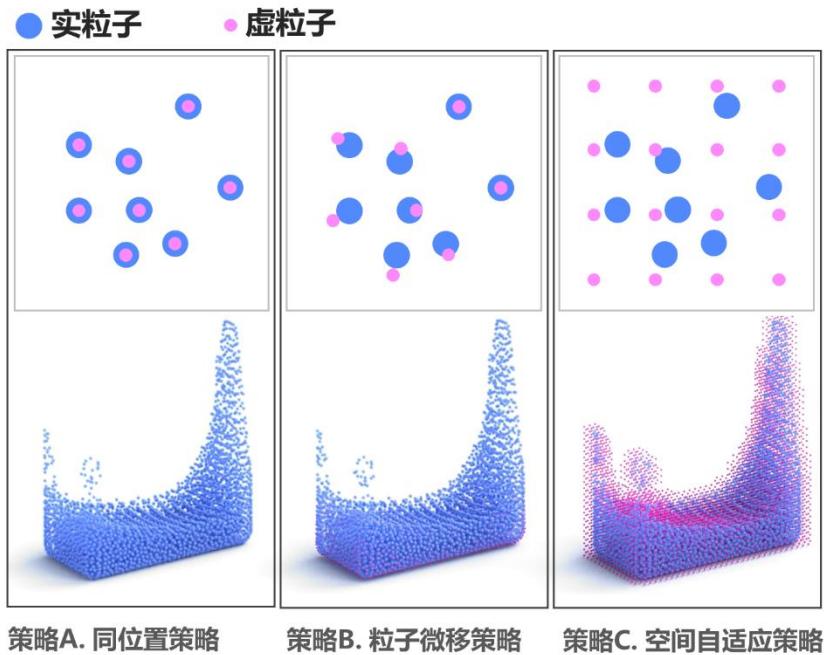


图 4-9 对偶粒子法研究中选用的 3 种虚粒子生成策略。

Figure 4-9 Three virtual particle generation strategies of our dual-particle approach.

在所有的粒子法中，均匀的粒子分布对于保证模拟的稳定性非常重要^[168]。因此理论上虚粒子的分布对于对偶粒子方法的稳定性也应有类似的作用。为了验证虚粒子分布是否会影响流体模拟的稳定性，本文引入了三种不同的虚粒子采样策略来生成虚粒子，分别为：共位置策略（Colocation strategy）、粒子微移策略（Particle shifting strategy）、以及空间自适应策略(Spatially adaptive strategy)，三种虚粒子采样见图 4-9 所示。以下是三种采样策略的实现方法与讨论。

(1) 策略 A. 共位置策略：如图 4-9(策略 A)所示，该策略中虚粒子与实粒子位置相同（以下简称策略 A）。为了在后续研究中，合理地验证评估对偶粒子策略^[99, 103]在改善流体模拟方面的作用，本文设计了等价于标准 SPH 投影法的共位置策略作为对比的“基线”。该虚粒子生成策略的实现方法，即是直接将实

粒子的位置复制给虚粒子，并将两种粒子的支持域半径设置为相同值 ($H = h$)。此时对偶粒子法就会退化为标准的 SPH 投影法。

(2) 策略 B. 粒子微移策略：如图 4-9(策略 B)所示，虚粒子的位置通过小幅度移动实粒子位置获得（以下简称策略 B）。在给出能够生成非常均匀且充足的虚粒子策略之前，本文尝试设计了一种能够小幅度改善虚粒子分布的策略，以用于验证是否虚粒子越趋向均匀，流体的仿真质量改善幅度也愈加明显。在这一策略中，首先将实粒子的位置复制作为虚粒子的初始位置，然后使用不包含速度更新与边界作用的 PBF 算法^[16]多次迭代快速修正虚粒子的位置。由于在 PBF 计算过程中忽略了边界、外力以及惯性的作用，修正后的虚粒子位置一定比实粒子更加均匀。

(3) 策略 C. 空间自适应策略：如图 4-9(策略 C)所示，虚粒子以均匀网格的形式排布在实粒子周围（以下简称策略 C）。为了保证虚粒子分布的均匀性，而且尽可能减小计算时间和内存的消耗，一种简单直接的方法即是使虚粒子等间距且正交的排布在实粒子周围，该策略下虚粒子覆盖的区域会随着流体的运动而发生变化。

得益于 SPH 离散化方法的无网格特性，合适的虚粒子生成策略可能不止上述三种。然而，上述三种策略已经足够用于验证虚粒子分布在抑制拉伸不稳定方面的作用。在使用策略 C 来生成虚粒子后，虚粒子则类似于欧拉视角的网格，这使得其与粒子-网格混合法具有一定的相似性。然而这两这种方法本质是完全不同的，在本文第 5 章将会对种方法的对比研究进行更深入讨论。

4.5.1 粒子微移策略（策略 B）虚粒生成方法

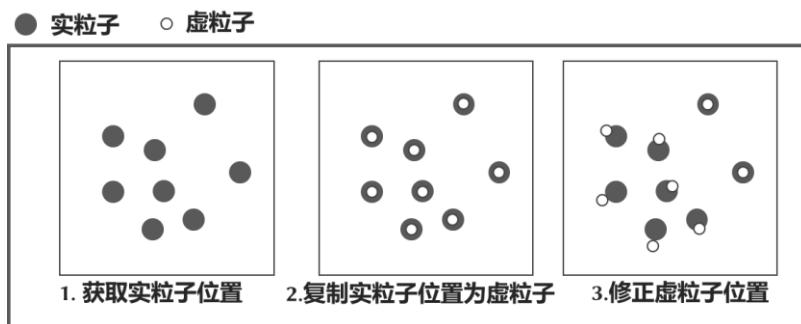


图 4-10 粒子微移策略的（策略 B）虚粒子生成方法。

Figure 4-10 The particle shifting strategy (S.B.) to generate virtual particles.

策略 B 的虚粒子采样方法是用于获取与实粒子位置较近，但分布更均匀的一组虚粒子。在这种策略中，虚粒子的数量与实粒子数量保持一致，本文改进了 PBF 算法^[16]来快速获取这一策略中的虚粒子。

PBF 法是用于模拟流体运动的一种非物理方法，该方法能够获得满足恒定密

度约束条件的粒子分布。使用该方法来获得虚粒子的位置需要对原算法进行相应地修改。由于虚粒子不具有速度，无需进行对流运动计算，因此原方法中的速度更新步骤以及人工粘性计算可以省去；此外，由于虚粒子不严格限定在流体内部，也无需考虑其是否会穿入到固体边界，因此原方法中边界条件相关计算也可以省略。使用修改后的 PBF 算法获取虚粒子的步骤如图 4-10 所示，具体为：

步骤 1. 获取实粒子位置；

步骤 2. 复制实粒子位置作为虚粒子的起始位置；

步骤 3. 使用修改后的 PBF 算法迭代修正虚粒子位置，相关方程为：

$$\delta \mathbf{x}_I = \frac{1}{\rho_0} \sum_J (\lambda_I + \lambda_J) \nabla W_{IJ} \quad (4-21)$$

$$\lambda_I = \frac{\rho'_I - \rho_0}{\rho_0 \left(\sum_J |\nabla W_{IJ}|^2 + \sum_J |\nabla W_{IJ}|^2 \right)} \quad (4-22)$$

上式中的 I, J 均为虚粒子， ρ_0 流体的静止密度， ρ'_I 为虚粒子使用虚邻域计算的密度。由于用于计算虚粒子位置的 PBF 算法没有考虑惯性、外力以及与边界的碰撞，因此其分布一定比实粒子均匀；此外，由于虚粒子的初始位置是实粒子的位置，PBF 算法在施加恒定密度约束时会尽可能减小粒子位置的修正量，因此最终获得的虚粒子与实粒子位置总是相互接近的。

4.5.2 空间自适应策略（策略 C）虚粒子生成方法

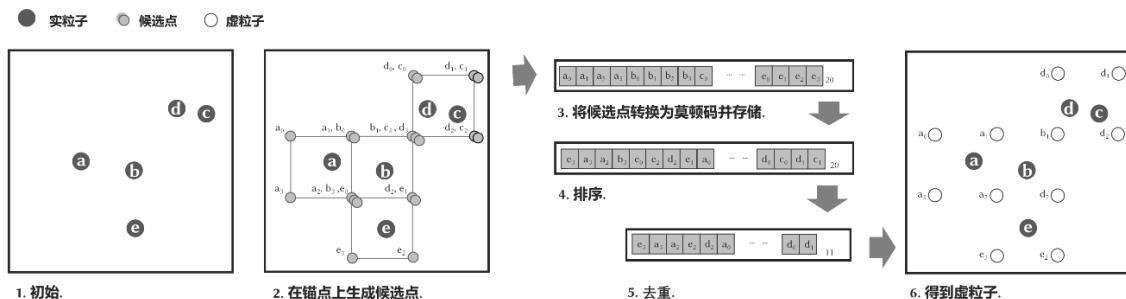


图 4-11 空间自适应策略（策略 C）虚粒子生成方法。

Figure 4-11 The spatially adaptive strategy (S.C.) to generate virtual particles.

在策略 C 中，虚粒子数量要多于实粒子，且虚粒子会正交、等间距地排布在实粒子支持域内。为了能够快速地在 GPU 上实现计算，本文采用了如下步骤来生成虚粒子：

- 第一步：获取实粒子的位置，同时将上一帧的虚粒子删除；
- 第二步：在每个实粒子支持域内的固定间距点位上生成候选点，这些候选点均位于均匀欧拉网格的网格中心；
- 第三步：将所有候选点的空间坐标转换为莫顿码；
- 第四步：对代表候选点位置的莫顿码进行排序；

- 第五步：将排序好的莫顿码存入向量中，并将其中重复的元素删除；
- 第六步：将该向量中剩余元素的莫顿码转换为空间坐标，作为网格节点（虚粒子）的位置。

上述生成虚粒子的步骤如图 4-11 所示。该策略可以很容易在 GPU 上实现并行计算。

4.6 验证与讨论

本节将对采用不同虚粒子策略的对偶粒子方法进行对比测试，以验证其稳定性^[169]；由于对偶粒子法与应力点 SPH 法^[150, 157]具有一定的相似性，因此本节也对比研究了两种方法的区别。

4.6.1 拉伸稳定性验证

为了验证对偶粒子法对于抑制拉伸不稳定方面的作用，本小节首先模拟了二维空间中旋转的方块形流体。在无重力空间中，流体方块以如下方程定义的初速度进行旋转：

$$\begin{cases} u_0 = \omega(y - y_c) \\ v_0 = \omega(x - x_c) \end{cases} \quad (4-23)$$

上式中的 $(x_c \ y_c)$ 为流体方块的中心点， ω 为流体方块旋转的角速度。使用上式获得的流体初速度满足速度无散度条件，即： $\nabla \cdot \mathbf{u}_0 = 0$ 。在测试过程中，流体方块会因离心加速度的作用而处于拉伸的状态，因此负压强会在流体内产生，很容易引起 SPH 法的拉伸不稳定问题。理想情况下，旋转的流体方块会逐渐产生四个逐渐伸长的旋臂。由于这一测试方法可以非常直观、直接地验证流体模拟方法的拉伸稳定性，因此在工程领域中该试验常被作为验证流体拉伸稳定性的标准试验（benchmark）^[155, 170]。

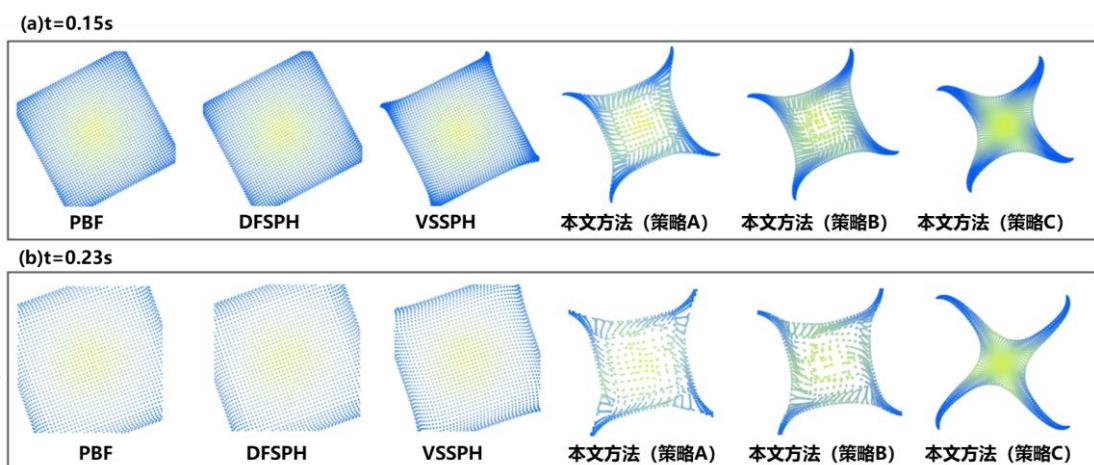


图 4-12 旋转的流体方块。
Figure 4-12 Rotating fluid square.

本文采用不同虚粒子采样策略的对偶粒子方法对比测试的结果如图 4-12 所示，测试中的本文方法均采用了相同的核函数、支持域半径与边界条件，每个流体方块由 2300 个粒子构成，并使用了人工粘性 XSPH 法，其控制粘性的参数被设置为 0.03^[128]。由对比结果可知，采用策略 A 的本文方法由于保留了负压强，因此能够出现四个伸长的旋臂，但受到了严重的拉伸不稳定的影响，粒子发生了明显的结块；采用策略 B 的本文方法相较于策略 A 有了一定改善，但改善并不明显；而采用策略 C 的本文方法，粒子没有出现结块且分布均匀，伸长的旋臂也相较于其它方法更为明显。

由图 4-12 中的测试可知，虚粒子生成策略对流体的行为以及实粒子的分布有非常大的影响。相较之下，使用策略 B 模拟的流体要优于策略 A，而策略 C 远优于策略 B，类似的仿真质量逐渐改善的现象也可以在后续的测试中观察到。策略 B 优于策略 A 的原因，是策略 B 的虚粒子分布相比策略 A 更加均匀。而且策略 B 中有部分虚粒子会与实粒子相互分离，参考本文 4.3.2 节中的讨论，这会有助于提高流体的拉伸稳定性。然而，策略 B 中依然有较多虚粒子位于与实粒子相同或相近的位置，且其分布也远不如策略 C 的均匀。因此，流体在拉伸稳定性方面的改善并不是十分的显著。相较于其它两种策略，策略 C 中的粒子不仅更加均匀、充足，而且该策略下的虚粒子不会跟随实粒子移动，完全摆脱了标准 SPH 法中压强采样点与速度采样点共同运动的特征，因此其模拟的流体的拉伸稳定性明显提高。

图 4-12 中也对三种具有代表性的其它方法进行了对比测试。其中 PBF 法是基于位置约束的方法^[16]，虽然该方法实现过程无需计算压强，但由于缺少粒子的吸附力，使得该方法不能模拟正确的流体行为；DFSPH 法中虽然能够对流体施加速度无散度条件^[97]，但该方法在实现过程中消除了负压强，同样因为缺失粒子间的吸附力作用，无法正确地模拟这一场景中的流体；VSSPH 法^[91]中包含有梯度修正^[146]、核函数归一化、错列粒子（staggered particle）^[102]、半解析式边界条件^[105]以及粒子位置修正（Particle shifting）^[46]等多项提高精确性与稳定性的方法和技术，但由于该方法中粒子间的吸附力的强度不够，也同样无法较好地模拟该场景中的流体运动。

4.6.2 稳定性的综合测试

溃坝场景是一种简单、但能够综合验证流体模拟算法鲁棒性的测试方法。本小节使用这一类场景对本文方法进行了更加综合、全面的测试，以验证本文方法的可用性与稳定性。

为了验证虚粒子突然的增删是否会影响流体的运动，本节使用策略 C 模拟二维空间中的流体溃坝场景。在图 4-13 所示的测试场景中，虚粒子突然增删的过程并未对流体的运动造成影响，模拟过程中也未发生明显的错误。

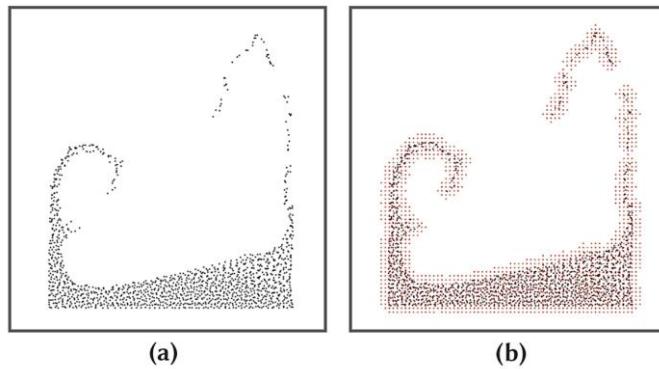


图 4-13 策略 C 模拟的二维溃坝场景。

Figure 4-13 The 2-d Dam break scenario is modeled by our method with S.C.

图 4-14 使用策略 C 以及不同虚粒子分辨率来模拟流体，该测试中虚粒子的间距由左至右分别为实粒子间距的 1.0 倍、1.2 倍、1.4 倍与 1.6 倍，而实粒子的间距均为 0.005m。测试结果表明不同分辨率虚粒子模拟的流体，其行为是较为接近的。在本文的对偶粒子法中，虚粒子的间距建议为实粒子间距的 1 到 2 倍。如果虚粒子的间距比实粒子间距小的过多，流体的运动可能会被过渡平滑化，造成大量流体运动细节的丢失。

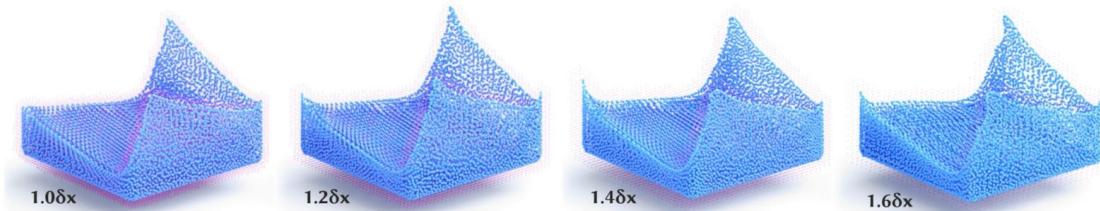


图 4-14 使用了不同虚粒子分辨率的策略 C 模拟的溃坝场景。

Figure 4-14 Dam break scenarios are simulated by S.C. with four different virtual particle sampling distances.

标准的 SPH 法不能稳定地模拟无粘性流体，一般需要使用 XSPH 法等人工粘性方法平滑掉部分流体的高频运动，同时避免数值误差引起的震荡，使流体粒子的运动保持相对一致。为了验证本文方法是否能够稳定地模拟无粘性流体，图 4-17 中选择了三种代表性的方法（PBF 法^[16]、DFSPH^[97]、VSSPH 法^[91]）以及策略 A、策略 B 和策略 C 来模拟无粘性流体。由图中的测试结果可知，本文之前的三种方法和策略 A 均无法稳定地模拟无粘性流体，策略 B 的结果虽然有所改善但并不明显。相比之下策略 C 模拟的流体有着相对正确、稳定的形态。策略 C 之所以能够稳定模拟无粘性流体，是因为它能够较准确地消除负的速度散度误差。如图 4-16 所示，当流体粒子相互远离时会产生速度散度误差，此时粒子之间会产生吸引力。由于策略 C 在流体处于拉伸状态下也较为稳定，因此能够将速

度散度误差大幅度削弱，使该组粒子的速度散度误差接近于 0，此时散度为零的粒子将会有相近的运动速度。

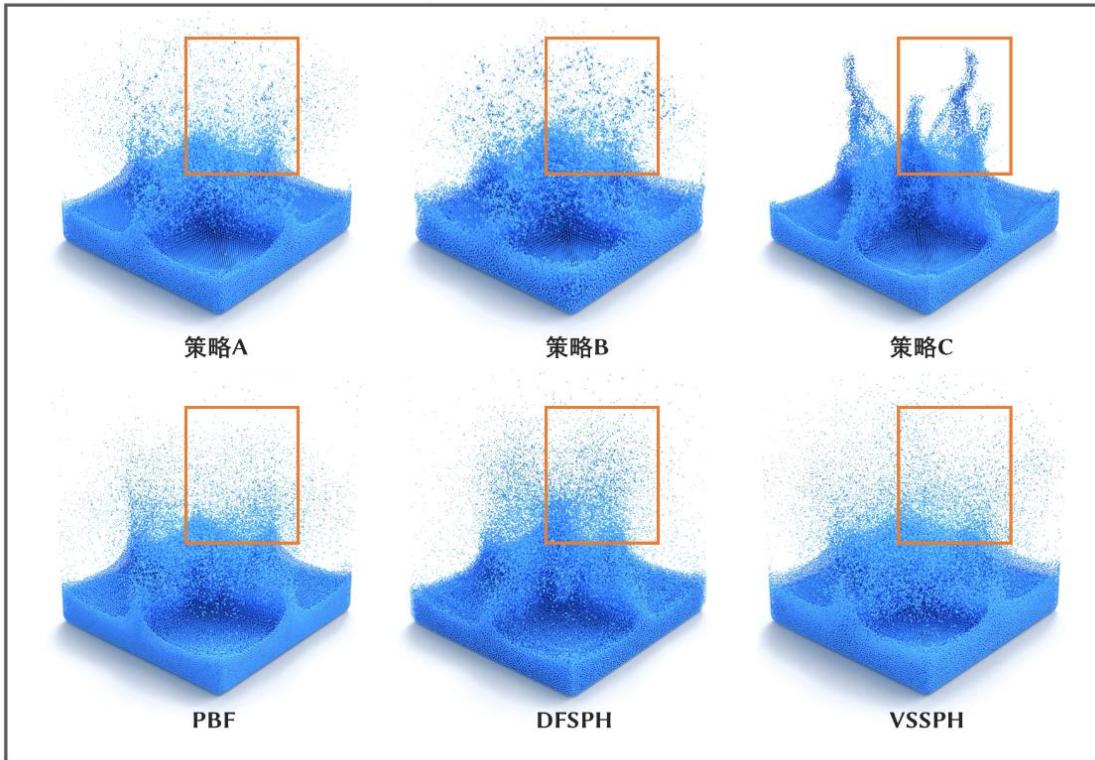


图 4-15 无粘性流体的溃坝测试。
Figure 4-15 Dam break test for inviscid fluids.

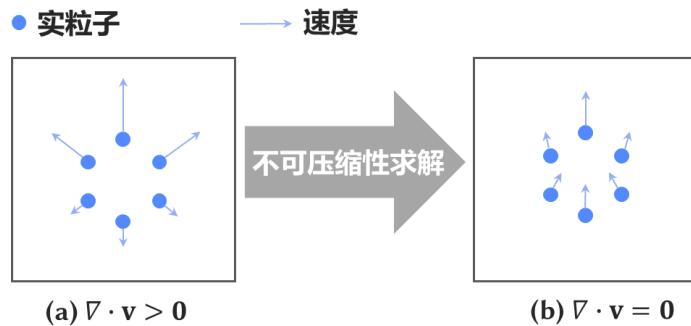


图 4-16 策略 C 能够稳定模拟无粘性流体。
Figure 4-16 S.C. can simulate the inviscid fluid stably.

如果对图 4-17 场景中各流体施加人工粘性 (XSPH) (如图 4-16)，粘性的参数被设置为 0.02，则各方法模拟的流体稳定性都有明显的改善。如图 4-16 所示，策略 B 相比于策略 A，其模拟的流体具有相对更准确的溅射细节，而只有策略 C 能够较为准确地模拟流体的薄层状态。此外，PBF 法、DFSPH 法、VSSPH

法三种方法依然无法稳定的模拟流体的薄层形态。该测试结果表明，仅在对比方法中增加粘性计算并不能抑制拉伸不稳定问题，同时也进一步验证了策略 C 抑制 SPH 流体拉伸不稳定问题的有效性。

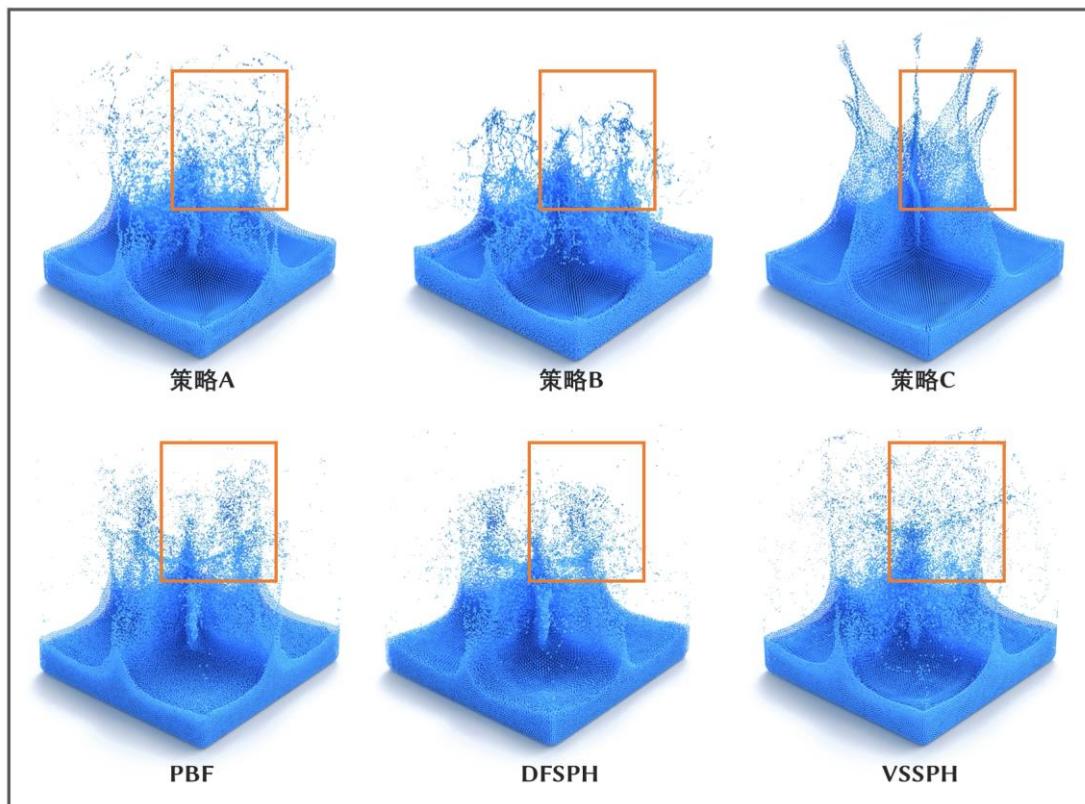


图 4-17 低粘性流体的溃坝测试。

Figure 4-17 Dam break test for low viscosity fluids.

4.6.3 收敛性测试

迭代收敛速度与可使用的时间步长上限对于流体仿真的效率非常重要。由于模拟流体的不可压缩性需要通过迭代求解压强泊松方程来实现，该求解过程的迭代收敛速度越快，单帧的计算量越少。而如果算法足够鲁棒，能够在保证一定仿真稳定性的前提下，使用更大的时间步长，则也可以显著降低仿真任务所需要计算的帧数，进而提高仿真计算的效率。

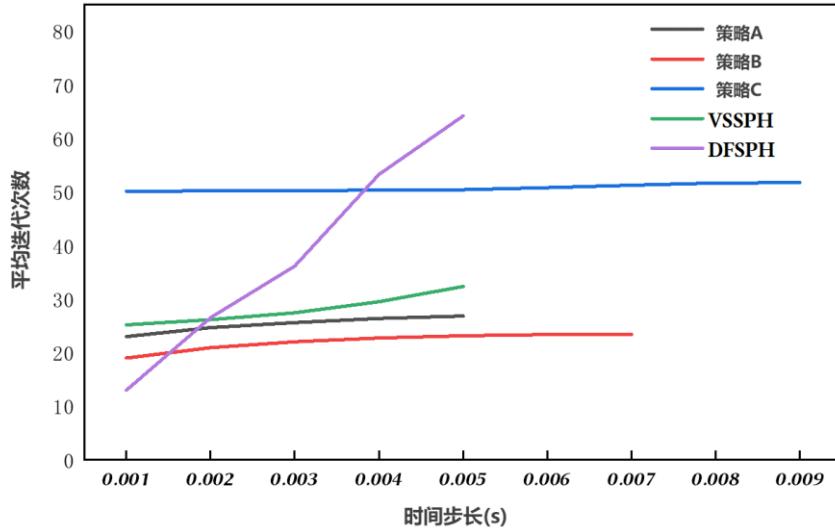


图 4-18 不同的时间步长下的平均收敛次数对比。
Figure 4-18 Average iteration numbers for different time steps.

为了验证对偶粒子方法中时间步长对收敛速率的影响，本小节使用与图 4-14 相同的测试场景，测试了使用不同时间步长的 DFSPH、VSSPH、策略 A、策略 B、策略 C 平均收敛次数，各方法的收敛阈值均为最大误差的 10^{-4} 。测试结果如图 4-18 所示。其中 VSSPH、DFSPH 与策略 A 能够使用的最大时间步长为 0.005 秒。由于策略 B 中虚粒子具有相对均匀的分布，这显著提高了其能够使用的时间步长上限；而策略 C 能够使用的时间步长上限远高于其它方法。此外，DFSPH 法虽然在小时间步长下能够快速的收敛，但当时间步长增大后其迭代次数显著增加。相较之下，VSSPH 与本文方法的收敛次数不会随时间步长显著变化。导致这一现象的原因，是 VSSPH 与本文方法均为投影法。在投影法中，时间步长只会影响泊松方程的源项，而不会影响到泊松方程的系数矩阵，而泊松方程计算的收敛速度只取决于其系数矩阵的性质，因此时间步长对于需要求解泊松方程的方法影响较小。

图 4-19 进一步对比了采用 0.001 秒时间步长，对偶粒子法模拟流体溃坝过程的迭代次数变化曲线。该测试中三种方法不可压缩性求解器的迭代收敛阈值均为最大误差的 10^{-4} 。相较于策略 A，策略 B 的迭代次数波动较小；而策略 C 的迭代次数远大于策略 A 与策略 B。理论上策略 C 中作为压强采样点的虚粒子更加均匀，本应具有更快的收敛速率^[12]。但在该测试中，策略 C 的测试结果与理论有较大的偏差。其原因是策略 C 中虚粒子的数量和位置会发生变化，导致上一帧的计算结果不能作为下一帧计算的初值。而缺少合适初值会大幅度增加不可压缩性计算的迭代次数。

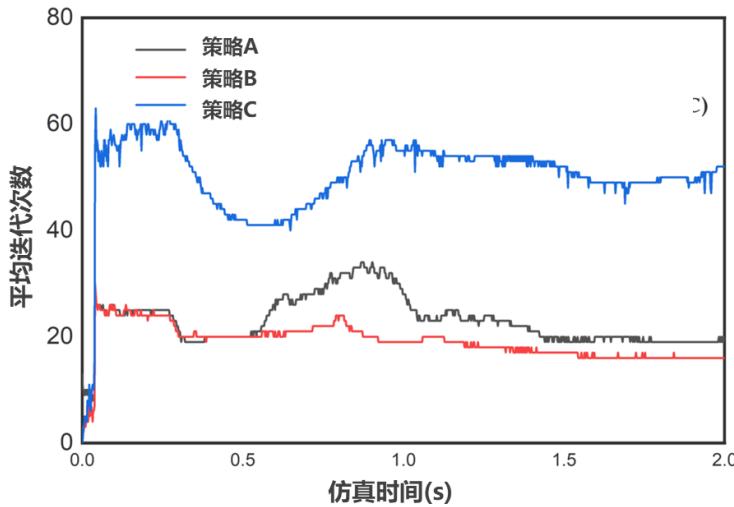


图 4-19 采用 0.001s 时间步长三种策略迭代次数对比。

Figure 4-19 Curve of the iteration number for a simulation with a time step size of 0.001.

4.6.4 与应力点 SPH 法的对比研究

本文对偶粒子法的提出受到了应力点 SPH 法的启发^[150, 157]。相较之下，两种方法都引入了用于定义应力或压强的“应力点”来避免零能问题。然而，由于应力点法中，携带速度的粒子和应力点必须保持着间隔错列的排布方式，在模拟大形变材料时，两种粒子的排布很容易被扰乱破坏，因此这类方法一般只能用来模拟小形变材料。为了使该方法能够模拟较大形变的材料，比如土壤、山坡等，Chalk 等人^[169]最近提出了一种新的速度粒子与应力点的排布方法。但是这类应力点法是否能够用于稳定、高效地模拟流体这类具有大形变速率的材料尚没有研究。

本文的对偶粒子法可被视为是应力点 SPH 法的改进，本文对偶粒子法中的虚粒子携带的压强，实为各项同性的“应力”。与本文的对偶粒子法相比较，应力点法中不包含有流体不可压缩性投影求解算法。此外，本文方法中虚粒子需要在每一个仿真步（step）中生成一次，而应力点 SPH 法中的应力点会跟随速度粒子一起运动。为验证应力点 SPH 法是否能够模拟流体，本文使用了 Chalk 等人^[169]提出的应力点与速度粒子排布策略，并结合对偶粒子法中的投影法求解器模拟了流体的溃坝场景，测试结果如图 4-20 所示。测试中两种方法均使用了系数为 0.05 的人工粘性（XSPH）算法来提高流体的稳定性，但应力点 SPH 法模拟的流体一旦发生形变，仿真会立即崩溃。导致这一不稳定现象的原因，是应力点 SPH 法中的应力点与速度粒子排布方式不适合模拟流体。如图 4-21 所示，一旦流体发生大幅度形变，跟随速度粒子运动的应力点很可能发生距离过近甚至重合的问题，该情况对于压强泊松方程计算的影响很严重，会直接导致求解很难收敛甚至发散。

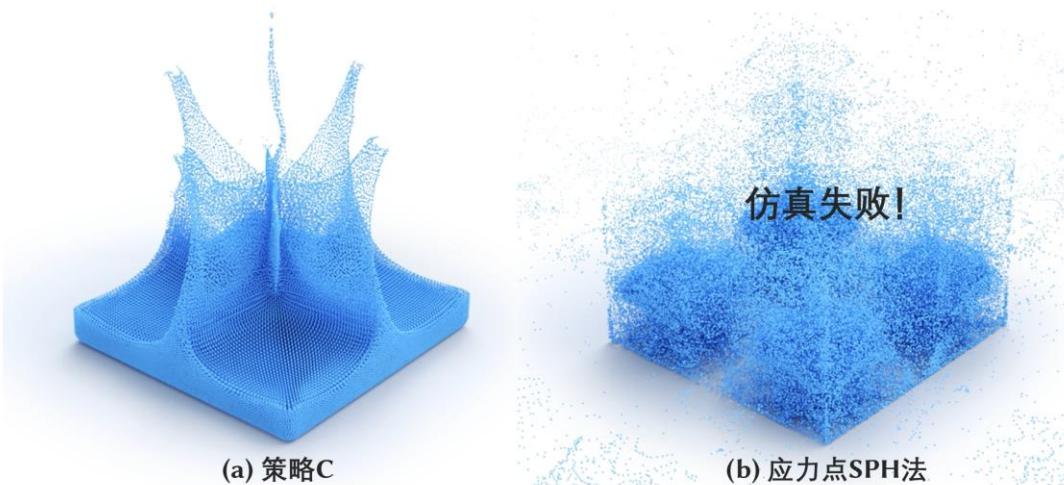


图 4-20 策略 C 与应力点 SPH 法的对比。

Figure 4-20 Comparison to Stress-Point SPH Method

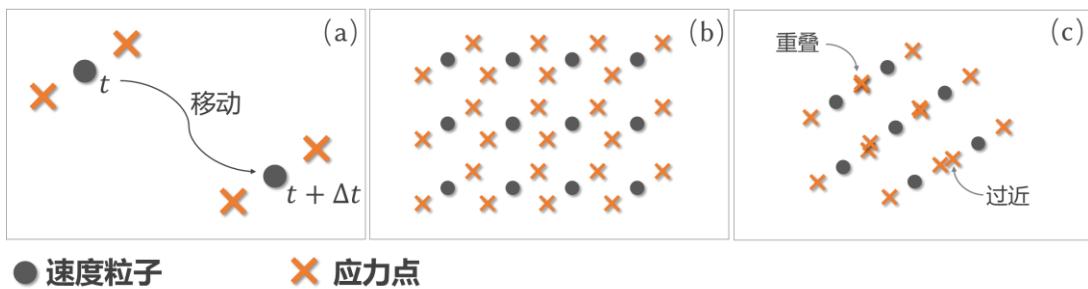


图 4-21 Chalk 等人提出的应力点 SPH 法^[169]。(a)应力点跟随速度粒子移动; (b)材料未形变时, 速度粒子与应力点的排布较均匀; (c)材料形变时, 应力点间很容易距离过近甚至重合。

Figure 4-21 Illustration of the Stress-Point SPH method.

4.7 综合性测试

为了更充分验证本文方法的稳定性，本节在更为复杂的测试场景中使用对偶粒子法模拟了不同特性的流体材料。所有的测试都是在安装有 Intel i7-8700k CPU 与 NVIDIA Geforce RTX2080 GPU 的个人电脑上进行的。模拟流体的算法中较为耗时的部分，比如邻域查找、不可压缩性求解、粘性求解等都使用了 GPU 加速。在所有的测试场景中，测试所使用的时间步长均为 1ms，实粒子与虚粒子的采样间距均为 $\delta x = 0.005m$ ，所使用的核函数均为三次样条函数形式（Cubic Spline Kernel），虚粒子与实粒子的支持域半径均为 2.5 倍的 δx 。

4.7.1 与其它粒子法的对比测试

为了更全面地验证本文对偶粒子方法对于提升流体小尺度细节模拟的作用，

本小节选择了三种代表性的方法与本文方法进行了对比测试。第一种方法是在计算机图形学中被广泛使用的粒子法——PBF 法^[16]；第二种方法是目前计算机图形学领域中最先进（State of art）的不可压缩性求解算法——DFSPH 法^[97, 114]，该方法不仅能够消除流体的密度误差，也能够减小速度散度误差，具有较高的效率和较好的鲁棒性；第三种方法是 VSSPH 法^[91]，该方法是错列粒子投影法^[102]结合变分框架^[171]改进后的方法，其实现过程使用了梯度修正方案^[146]、核函数归一化、半解析式边界条件^[105]以及粒子位置修正（Particle shifting）^[46]等能够提高算法稳定性的技术。

(a) 无粘性

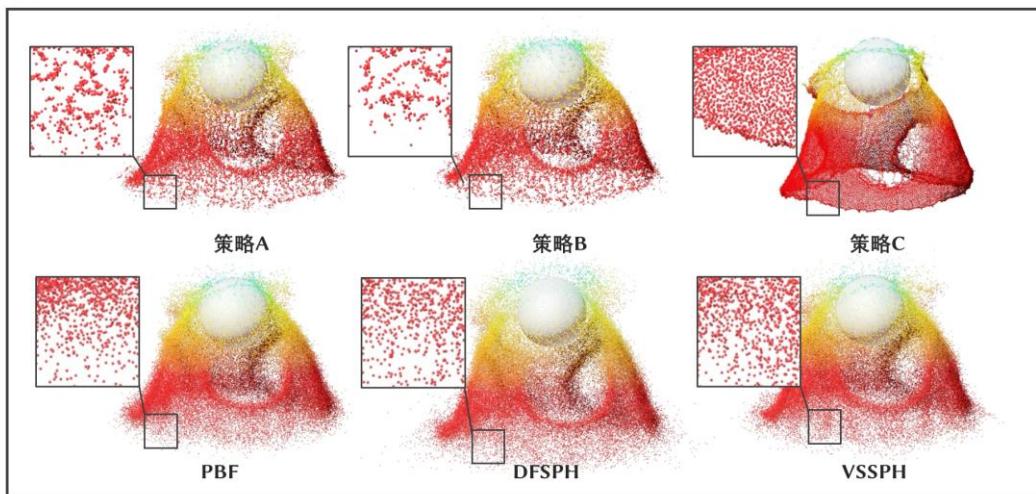


图 4-22 无黏性流体拉伸状态的对比。

Figure 4-22 Comparison of inviscid SPH fluids in tensile state.

(b) 人工粘性

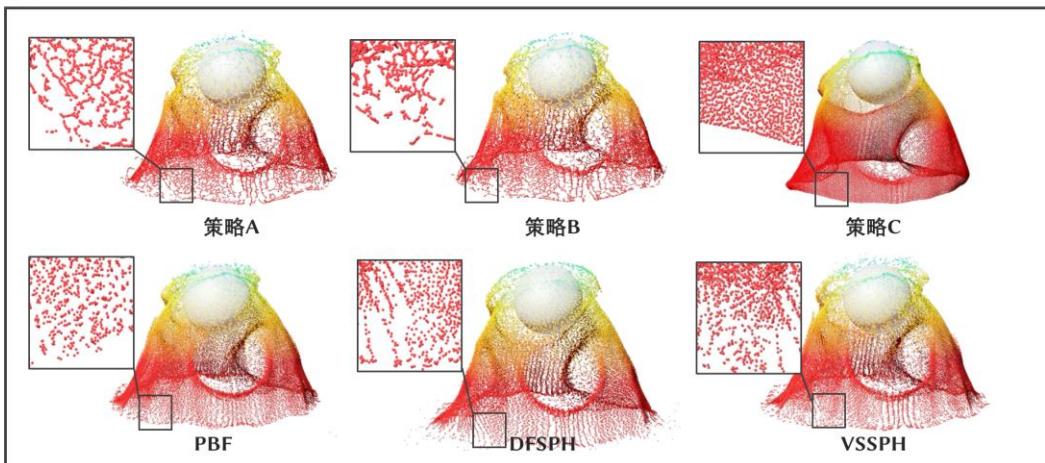


图 4-23 使用了 XSPH 粘性的流体拉伸状态的对比。

Figure 4-23 SPH Fluids using XSPH model in tensile state.

(c) 人工粘性+表面张力

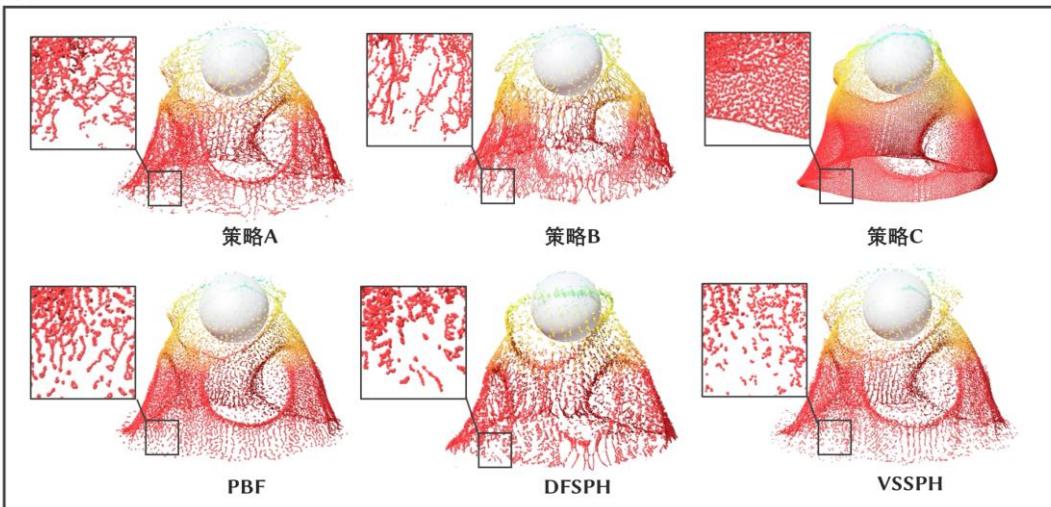
图 4-24 施加了人工粘性 (XSPH) 与表面张力作用^[104]的流体拉伸状态的对比。

Figure 4-24 SPH Fluids using XSPH model and the surface tension in tensile state.

为了展示不同方法在模拟流体薄层形态方面的稳定性，图 4-22 中使用不同方法模拟了重力与惯性作用下的流体拉伸形态。图 4-22 的测试结果说明，无人工粘性作用下的策略 C 能够非常稳定地维持流体的薄层形态。而相较之下，因为缺少人工粘性的作用，PBF 法、DFSPH 法、VSSPH 法中模拟的流体粒子运动都较为杂乱。此外，策略 A 与策略 B 因存在拉伸不稳定问题，其模拟的流体薄层形态会发生碎裂。为验证人工粘性是否能够有效增强流体拉伸状态的稳定性，图 4-23 在图 4-22 测试场景中进一步地增加了人工粘性的作用。由测试结果可知，虽然人工粘性作用能够使粒子的运动相对一致，但依然无法提高拉伸稳定性。PBF、DFSPH、VSSPH 法模拟的流体薄层发生明显的断裂，而策略 A 与策略 B 模拟的流体因拉伸不稳定问题，粒子出现严重的结块。相较之下，策略 C 依然展现出了非常稳定的流体薄层形态，拉伸状态下的粒子分布非常均匀。现实世界中，大多数流体的小尺度运动会很大程度受到流体的表面张力的影响，而且也有研究表明表面张力作用可以增强流体溅射等细节的模拟质量^[104]，因此图 4-24 在图 4-23 测试基础上进一步增加了表面张力作用的计算，表面张力算法采用具有优异鲁棒性的 He 等人提出的方法^[104]。测试结果显示表面张力虽然能够增强流体的溅射效果，但依然无法改善 PBF、DFSPH、VSSPH 的拉伸稳定性。

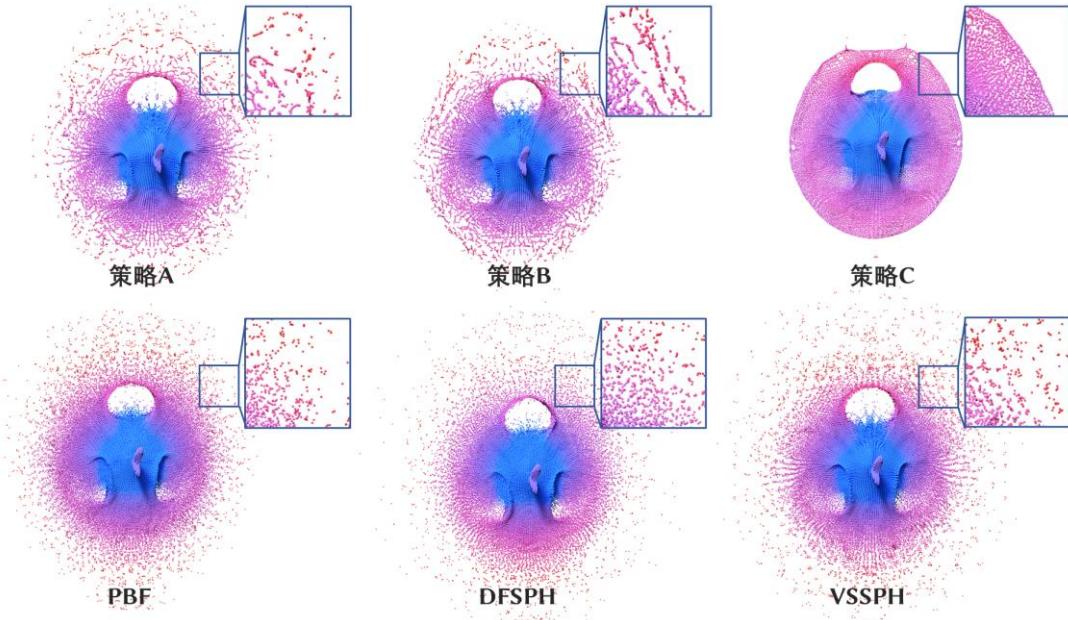


图 4-25 两个流体在无重力空间中的碰撞。

Figure 4-25 Collision of two fluids in gravity-free space.

为了进一步展示策略 C 在模拟流体薄层方面的优越性，图 4-25 测试了无重力空间中两流体对撞后形成薄层的过程，其中各流体都被施加了 XSPH 人工粘性。对撞的两流体以镜像对称的方式排布粒子。对撞发生后，只有策略 C 能够稳定地模拟流体的拉伸形态，且该方法获得的粒子分布较为均。策略 B 的稳定性虽然相比策略 A 有小幅度的提升，但依然不明显。

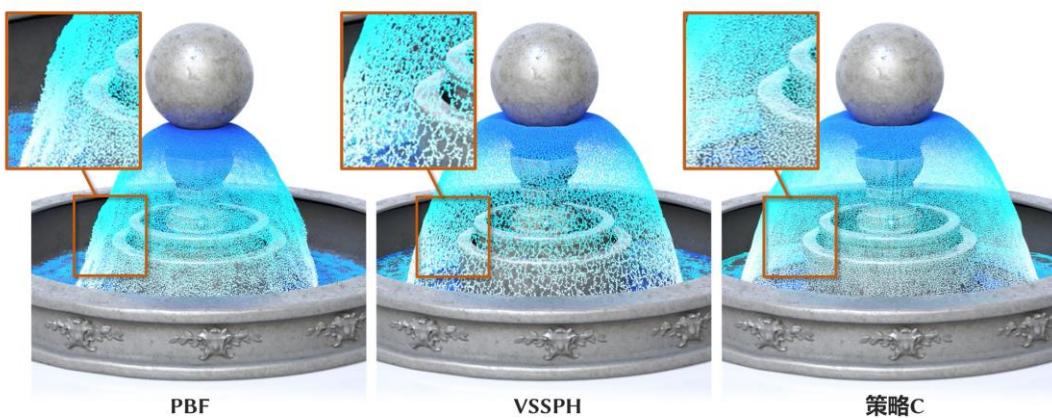


图 4-26 喷泉。

Figure 4-26 Fountain.

图 4-26 展示了使用 PBF 法、VSSPH 法、本文方法模拟的喷泉场景，该场景每组测试中都包含有 100 万个粒子，流体的人工粘性控制系数被设置为 0.05。该测试中，只有本文方法模拟的喷泉，能够更真实、更自然地表现出水流碰撞到石

球后在重力作用下的薄层形态。

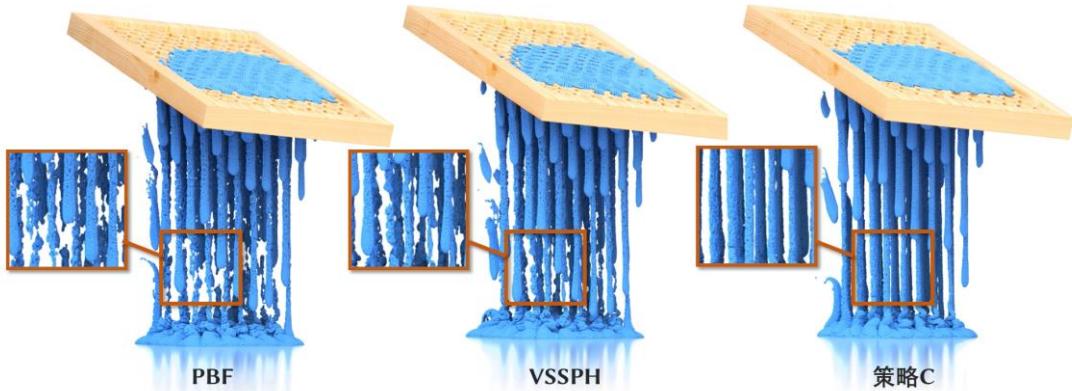


图 4-27 大粘性不可压缩性流体。

Figure 4-27 High viscosity incompressibility fluids.

SPH 的拉伸不稳定问题也会严重影响大粘性流体材料的仿真稳定性。在人们的日常生活中，经常能见到蜂蜜、果酱等大粘性流体材料的拉丝、薄层的效果。在使用 SPH 法模拟这类现象时，虽然粘性计算能够缓解流体的拉伸不稳定问题，但依然会在模拟中产生错误的断裂。为验证对偶粒子方法是否能够提高大粘性流体仿真的稳定性，图 4-27 使用 PBF、VSSPH 以及基于策略 C 的本文方法模拟了大粘性流体材料。该场景中每组流体的粒子数量为 50 万个，粘性基于本文第 5 节方法实现，粘性系数为 $500 \text{ pa} \cdot \text{s}$ 。此外，流体被施加了表面张力作用^[104]。由该图可知，PBF 法模拟的流体在拉伸状态下会快速的断裂，并碎裂成不规则的流体块，VSSPH 法能够相对更稳定地模拟这一过程，但依然会产生明显的错误。相较之下，策略 C 则几乎不存在前两种方法的问题，能够稳定地模拟大粘性流体的拉伸形态。该场景的测试结果进一步被重构为网格并进行了渲染，渲染结果如图 4-28 所示。



图 4-28 重构了表面网格后的大粘性流体材料。

Figure 4-28 High viscosity incompressibility fluids are rendered as meshes.

4.7.2 更大规模的测试



图 4-29 溃坝场景。

Figure 4-29 Dam break.

为了验证本文对偶粒子法模拟大规模复杂场景时的稳定性，并展示其视觉质量，本小节使用基于策略 C 的对偶粒子法在条件更复杂、更大规模的场景中进行了测试。

图 4-29 使用策略 C 模拟了包含有 50 万粒子的流体溃坝场景，并基于测试后的粒子数据生成了网格。在该测试场景中，流体出现了较为稳定地薄层形态。



图 4-30 坠落的水杯。

Figure 4-30 Dropping cup.

图 4-30 使用策略 C 并结合流固耦合算法^[172]模拟了装满水的水杯在摔在地面上的过程。水杯中的水被甩出，并展现出了非常自然的薄层形态。该场景中构成流体的实粒子数量为 50 万个，虚粒子的数量变化范围为 57 万到 170 万。



图 4-31 多个流体的坠落。
Figure 4-30 Many dropping fluids.

图 4-31 使用策略 C 模拟了包含 150 万实粒子的场景。该场景中有多个由流体构成鱼坠落在地面上，产生了丰富的流体小尺度细节。该测试场景中，虚粒子数量变化范围为 180 万到 410 万个。

图 4-32 使用策略 C 模拟多个流体撞击固体石像鬼（Gargoyle）后溅起水花的过程。该场景包含有 150 万个粒子，流体的人工粘性的控制系数被设置为 0.05。渲染过程中将粒子的速度映射为了颜色。石像鬼固体具有相对复杂的形态，流体撞击后也相应地产生了尺度多变、形态多样流体溅射。该场景中的构成流体实粒子数量为 150 万个，虚粒子的数量变化范围为 180 万到 620 万个。

4.8 本章总结

本章的研究工作首次提出了基于对偶粒子的不可压缩 SPH 流体投影计算方法，该方法可以有效抑制 SPH 流体的拉伸不稳定问题，而且能够显著提高流体小尺度运动细节模拟的稳定性。本章首先讨论了 SPH 流体产生拉伸不稳定问题的原因，并提出了使用对偶粒子方法来抑制拉伸不稳定问题的策略。对偶粒子法包含携带有速度的实粒子和作为压强采样点的虚粒子。本章演示了如何使用这两种粒子离散压强泊松方程与梯度算子，并实现不可压缩性流体仿真。本章的实验证明了对偶粒子法能够有效克服 SPH 法的拉伸不稳定问题，并可稳定地模拟流体的溅射、拉丝、薄层等小尺度细节。此外，本文也研究了不同虚粒子分布对不可压缩计算收敛性与稳定性的影响，并提出了基于 GPU 的空间自适应虚粒子快速生成方法，以高效高质量地实现稳定的流体仿真。

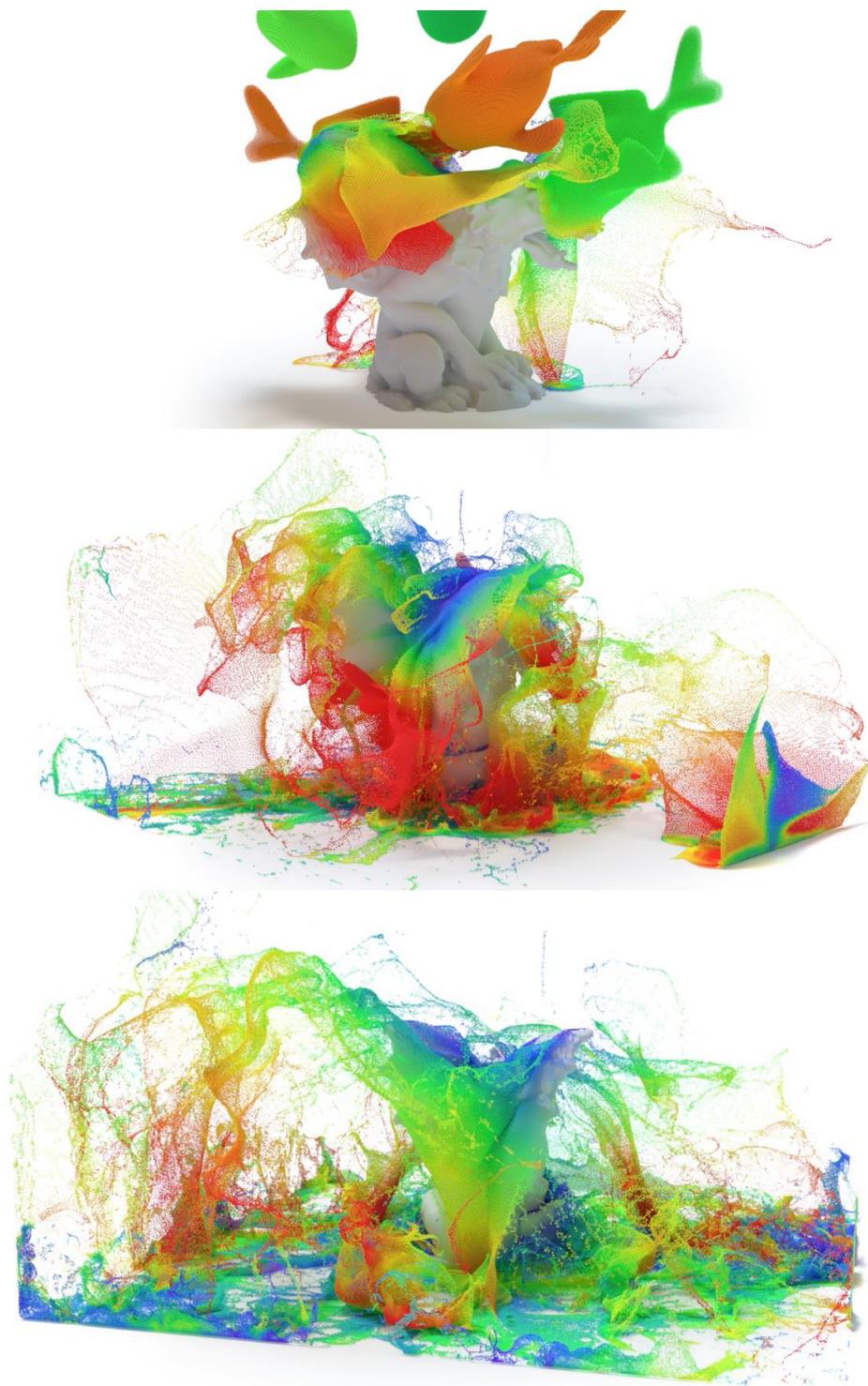


图 4-32 流体与固体石像鬼相撞的过程。

Figure 4-31 Collision between solid gargoyle and fluids.

第 5 章 基于对偶粒子法的通用流体仿真引擎

5.1 引言

本文上一章提出的对偶粒子法中包含有两部分粒子系统，分别为直接表示流体的实粒子，以及用于辅助仿真计算的虚粒子。在上一章研究中发现，通过调整虚粒子采样的策略，可以显著改变流体仿真方法的特性。比如：将实粒子与虚粒子重合放置，该方法会退化为标准的 SPH 法；若采用策略 B 生成虚粒子，则可以在不引入过多虚粒子的前提下提高迭代收敛速度与仿真稳定性；若采用策略 C 生成的虚粒子，该方法就能非常有效地抑制 SPH 流体的拉伸不稳定问题。如果进一步将虚粒子视为欧拉网格，并适当修改标准 SPH 法的离散化方案，则原对偶粒子法会进一步演变为以 PIC、FLIP 法为代表的粒子网格混合法^[35]。因此基于本文的对偶粒子方法，理论上可以建立一种同时兼容标准 SPH 法、对偶粒子法、粒子网格混合法等多类粒子方法的通用流体仿真引擎框架（图 5-1）。

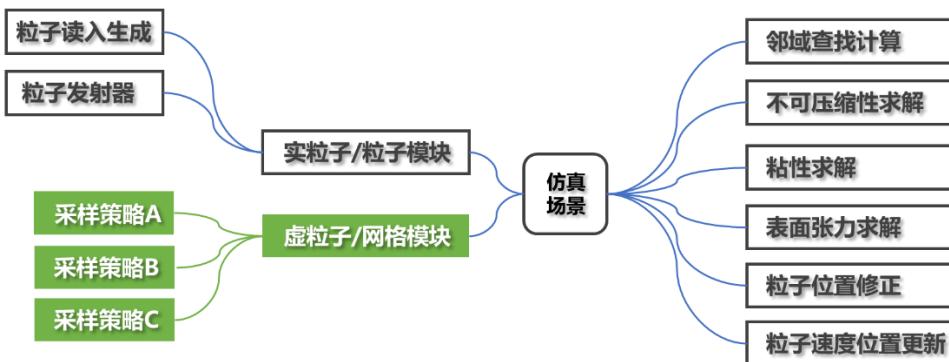


图 5-1 基于对偶粒子的流体仿真引擎框架。

Figure 5-1 A framework for dual-particle based fluid simulation engine.

为推进本文研究工作的实际应用，推动流体仿真研究的发展，本章将基于前文讨论的理论和方法，构建一种基于对偶粒子法的高效、通用流体仿真引擎框架。该流体引擎包含有多种基于粒子法的流体仿真相关算法。为了保证流体仿真具有较好的计算效率与易用性，本文根据 GPU 硬件特性对算法的框架、仿真策略、数据结构等方面进行了针对性的设计，使该引擎能够充分利用 GPU 来加速流体仿真计算。

由于该仿真引擎能够较容易地实现粒子网格混合法的构建，因此易于在完全相同的测试条件下研究对比粒子网格混合法与本文对偶粒子法之间的差异。在本章的研究中发现，相较于 PIC 法、FLIP 法等粒子网格混合法^[35]，本文对偶粒子法模拟的流体，其数值粘性更小且粒子分布更加均匀。针对粒子网格混合法中粒子分布不均匀的问题，本文使用 SPH 粒子位置修正技术^[91]改善了此类方法的粒子分布，提高其仿真稳定性。

5.2 流体仿真引擎的框架设计

流体仿真较为复杂的计算任务。完成一项完整的流体仿真任务，常需要用到多种不同计算步骤，其执行顺序、数据传输的方式等都需要进行合理的配置；而且仿真计算中需要使用到多种不同的硬件资源及其功能，是否能够合理调配使用各硬件资源也会直接影响仿真计算的效率。此外，仿真引擎中也需要包含数据输入输出、仿真过程的实时渲染以及获取用户的交互信息等功能，如何协调这些过程与仿真计算也直接影响了流体仿真引擎的可用性。

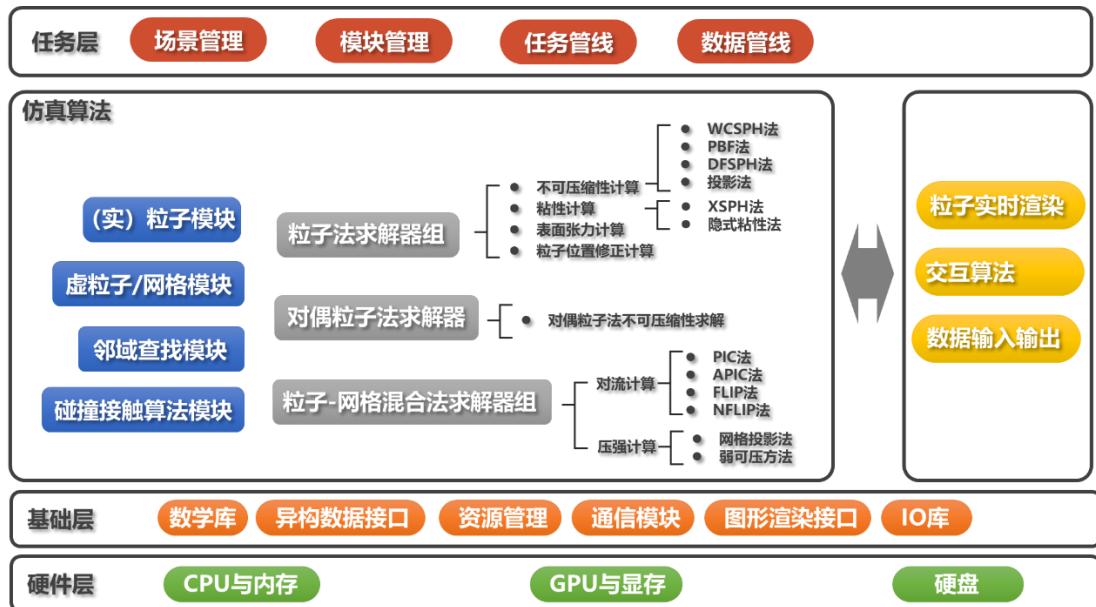


图 5-2 流体仿真引擎系统架构图。

Figure 5-2 Fluid simulation engine system architecture.

基于上述考虑，本章设计了如上图所示的流体仿真引擎架构。该流体仿真引擎的软硬件模块可以划分为 4 个层，其中包括有 CPU、GPU 构成的硬件平台层、核心仿真算法层、以及为核心算法层提供通用的数学计算和所需功能 API 接口的基础层，在核心算法层之上是用于配置仿真任务的任务层，该层中的模块用于实现仿真场景搭建、数据的读入输出、算法模块调配与仿真中间数据的传输控制等。

在核心算法层中，包含有两部分算法，一部分是直接模拟流体运动的求解器，另一部分为协助求解器计算的模块，比如粒子生成、网格生成、邻域查找、碰撞等模块。本文的流体仿真引擎中包含有较多的仿真算法以及相关模块，执行仿真任务时常需要使用到不同求解器与模块，其执行顺序与数据管线的配置，则需要使用任务层来调配。此外仿真过程的渲染、仿真数据的输出、以及用户的交互信息也会在任务层中进行协调。为了保证仿真计算的效率，引擎的主要计算任务都是在 GPU 上实现的，因此也需要根据 GPU 特性来设计求解器算法与相关的数据结构。

5.2.1 仿真场景

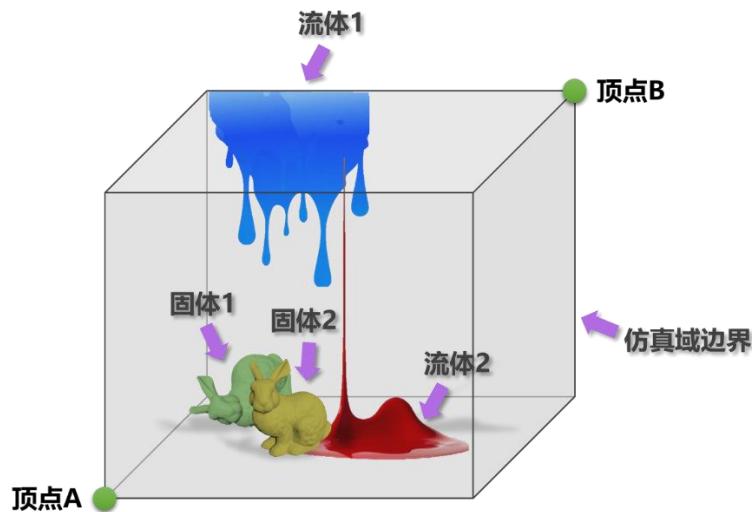


图 5-3 仿真场景。
Figure 5-3 Scenes for fluid simulations.

如上图所示，多数的流体仿真场景可能包括的对象有：仿真域（问题域）边界、一个或多个流体、一个或多个能够与流体发生碰撞的固体。仿真域边界即是材料仿真所在空间的最外层边界，只有仿真域空间内的粒子与固体边界会参与仿真计算。仿真域的边界一般为立方体，可由两个对角顶点的坐标来定义（图 5-3 中的顶点 A 与顶点 B），当引擎检测到有粒子穿过仿真域边界时，需要将粒子位置修正回仿真域。

仿真场景中需要使用到与流体产生碰撞作用的固体，本文引擎中流固边界实现方法有如下两种：

1. **固体边界法。**流体仿真引擎也可以直接读取有向距离场模型或面片形式的模型作为固体边界。而碰撞模块会用于检测流体是否与固体边界发生碰撞。当碰撞发生时，可采用如下算法来修正粒子的位置和速度。

$$\begin{aligned} & \text{IF } (\mathbf{u}_i^*)_{\mathbf{n}} > 0 \\ & \text{THEN } \mathbf{u}_i^{new} = [-(\mathbf{u}_i^*)_{\mathbf{n}} + \beta(\mathbf{u}_i^*)_{\mathbf{f}}]e^{\delta t} \end{aligned} \quad (5-1)$$

上式中的 \mathbf{n} 为与粒子 i 碰撞的固体边界法向量， $(\mathbf{u}_i^*)_{\mathbf{n}}$ 为这一法线方向的速度， $(\mathbf{u}_i^*)_{\mathbf{f}}$ 为法线的垂直方向（边界切向方向）的速度， e 为自然常数。 β 为不大于 1 的整数。当 β 取 0 时，仿真域边界附近的流体就会被施加无滑移边界条件；当 β 取值为 1 时，仿真域边界附近的流体就会被施加自由滑移边界条件。该方法尽管能够避免流体穿入到固体中，但在流固边界附近由于邻域缺失问题的影响，会导致边界附近流体粒子分布非常不均匀。

2. **幽灵固态粒子。**固态粒子法是在固体所占据区域内均匀的填充幽灵固态粒子的方法。这种方法能够有效地避免流固边界附近因粒子邻域缺失而导致的流体

粒子分布不均匀问题。该种边界施加方法经常会发生流体粒子穿入到固体等错误的情况，为了避免该问题，可在流固边界位置设置有向距离场边界，将幽灵固态粒子与固体边界法结合使用。

在引擎中可以存在多种不同材料的粒子，比如流体粒子、固态幽灵粒子等。所有粒子均可通过读入有限距离场模型文件的方式生成粒子。具体实现方式为：先读入有向距离模型文件，然后在模型内部采用等间距或泊松分布的方式填充粒子，并赋予粒子质量、材料类型、速度等属性。其中泊松分布采样粒子在某些场景中也较为必要，比如模拟大粘性流体的拉伸时，采用泊松分布的流体粒子更不容易在拉伸状态下被拉断。泊松盘采样的粒子则可通过 Bridson 提出的方法^[173]来获得，但这一方法只能在 CPU 上实现且时间复杂度较高，因此只能在仿真开始之前的预处理阶段生成。在开始仿真之后，该泊松盘采样的粒子才会被传入到 GPU 的内存中用于计算。

流体粒子也可通过添加粒子发射器的方式生成。粒子发射器会在每间隔一定的仿真时间后，在仿真场景中某特定位置添加若干个新的流体粒子。发射器生成粒子后需要在 GPU 上重新分配内存（显存），以避免过频繁的 CPU 与 GPU 之间的数据交换影响仿真效率。

5.2.2 场景节点图与管线

为说明本文流体仿真引擎的场景搭建方法与仿真任务的执行方式，本节在仿真引擎上搭建了基于本文 SPH 流体的 SIMPLE 迭代算法，测试如图 5-4 所示的场景。该场景中的粒子数量为 5.4 万，SIMPLE 迭代次数选择为 5 次，粘性系数为 $5 \times 10^4 Pa \cdot s$ ，粒子间距为 0.005m，平滑距离为 0.0125m，时间步长为 1ms。



图 5-4 使用本文 SPH 流体的 SIMPLE 迭代算法模拟冰淇淋。

Figure 5-4 An ice-cream is modeled by our SIMPLE iterative algorithm.

图 5-4 中场景中包含有两个对象：用于产生冰淇淋流体的发射器，用于表示托盘的固体边界。该场景中使用到了多种求解器与模块，包括粒子邻域查找模块、

基于 SPH 法的 SIMPLE 迭代求解器、粒子位置更新模块、以及流体与固体的碰撞模块。该场景的节点图如下图所示。

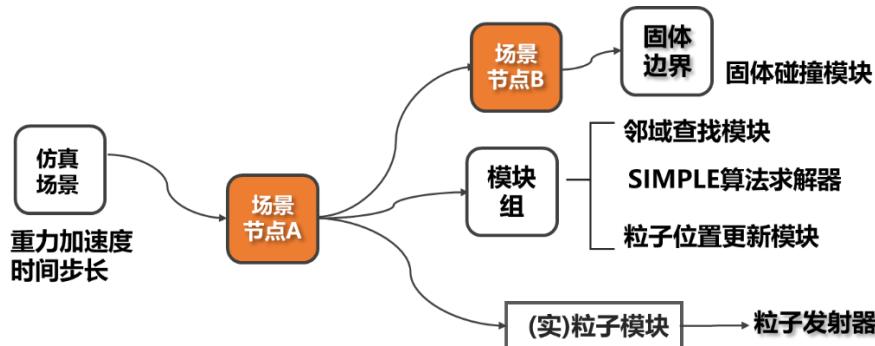


图 5-5 使用本文流体仿真引擎模拟的冰淇淋场景节点图。

Figure 5-5 Node diagram of the ice cream scenario in our fluid simulation engine.

在仿真开始后，流体仿真引擎会从仿真场景的根节点开始遍历其各个子节点，在场景节点 A 中，存储有流体粒子的速度、位置、属性等信息，同时它有另一个子节点——存有固体边界与固体碰撞模块的场景节点 B。流体在模拟过程中待执行的求解器与模块均挂载在场景节点 A 上。当仿真开始后，该模拟任务执行的时序图如下图所示。

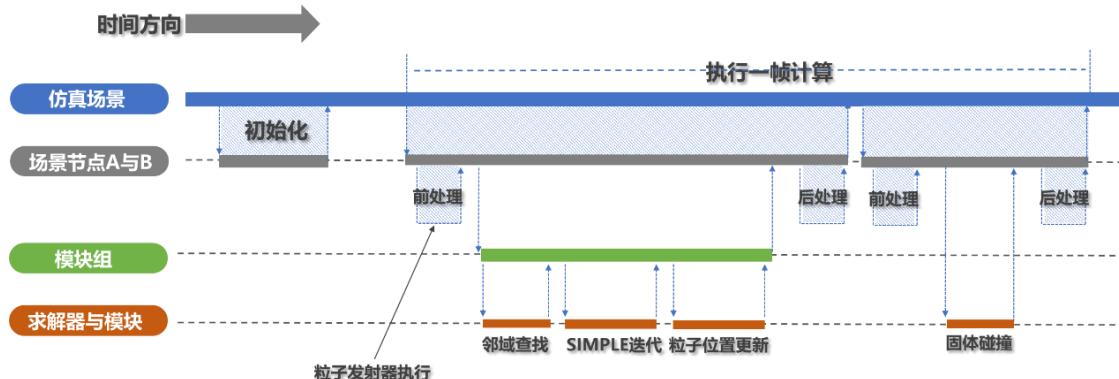


图 5-6 使用本文流体仿真引擎模拟的冰淇淋场景（图 6.3 场景）时序图。

Figure 5-6 Timing diagram of the ice cream scenario (Figure 6.3) in our fluid simulation engine.

仿真开始后，引擎会先执行场景节点的初始化过程，然后再开始后续的仿真计算。在仿真的每一帧计算中，场景节点 A 与节点 B 上挂载的求解器与模块都会按照预设顺序被执行。在场景节点的模块计算开始前与计算完毕后，需要进行相应地前处理计算与后处理计算，比如图 5-4 场景中的流体粒子发射器，就需要在场景节点 A 前处理阶段执行。

5.2.3 数据结构

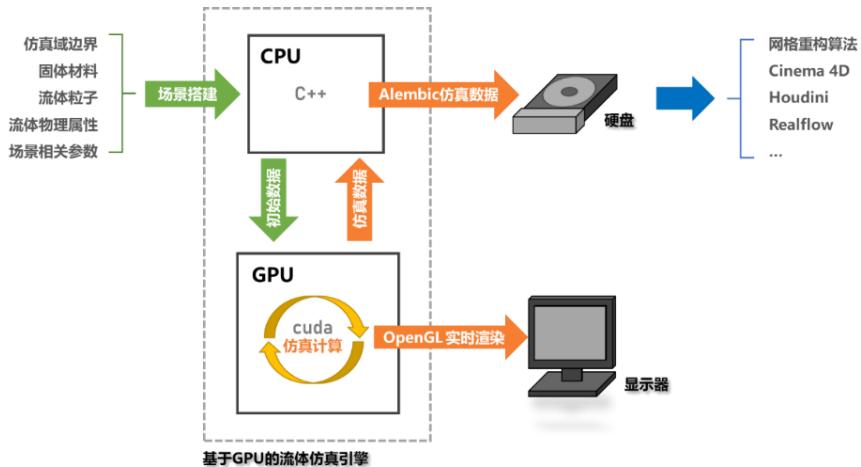


Figure 5-7 An application example of our fluid simulation engine.

流体仿真过程中包含有大量可并行的计算，非常适于使用 GPU 硬件来提高引擎的计算效率。但对于直接面向用户的一些接口和功能，如参数设置、场景数据导入等，则更适于在 CPU 上实现。因此流体引擎的设计要充分考虑结合 CPU 与 GPU 的特性。基于上述考虑，本文流体仿真引擎应用示例如上图所示。

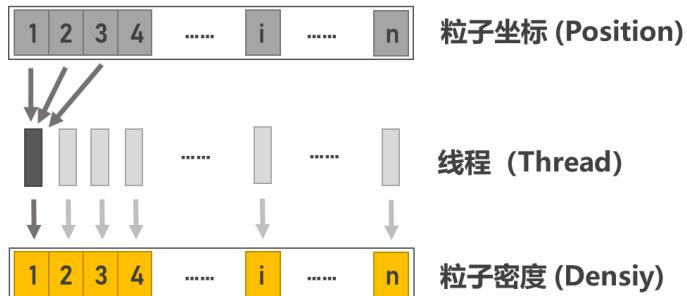


图 5-8 基于 GPU 的粒子质量密度计算。

Figure 5-8 Calculation of particle mass density based on GPU.

为了便于实现在 GPU 上的并行计算，引擎中粒子、虚粒子或网格的各参数以分别独立的数组方式存储于 GPU 的显存中，这种存储方式易于实现在 GPU 上的并行计算。此处以粒子的质量密度计算为例对这一过程进行说明。如图 5-8 所示，密度求解器首先会经由场景节点读取该场景中存有粒子位置的数组，并开辟出一定量的线程对这一数组进行计算，线程的编号与密度数组索引相对应。密度求解步骤开始之后，每个线程会根据粒子的邻域列表来查找粒子坐标数组中对应元素，并基于式(2-40)进行各粒子质量密度的计算，计算结果会依据线程的编号分别存储于密度数组对应位置中。

5.2.4 邻域查找

在 SPH 法中，流体的粘性计算、不可压缩性计算等都需要使用到粒子的邻域，即：每个粒子的相邻粒子。因此在每一帧仿真的最初阶段，均需要进行粒子邻域查找。而粒子邻域查找计算的时间与空间复杂度较高，因此合理的粒子邻域查找策略以及粒子的邻域列表的存储方式，对于提高引擎整体的计算效率至关重要^[68]。

若通过遍历全部仿真域中粒子的方式来查找各粒子的邻域，会导致邻域查找计算的时间复杂度过高 ($O(n^2)$)，而严重影响仿真的效率。因此可在邻域查找计算前先使用空间均匀网格根据粒子位置对粒子进行分组，在邻域查找计算时只查询特定分组中粒子，即可快速加速粒子邻域查找过程^[68]。

由于流体运动的不可预测性，每个粒子的邻域数量并不确定，而如果每个粒子都使用单独的数组来存储其邻域，会引入大量内存分配、释放等操作，不仅影响计算效率，而且难以在 GPU 上高效地实现。为了避免过于频繁的内存分配操作，本文采用了 ArrayList 方式存储各粒子的邻域(图 5-9)。粒子的邻域对象中，包含有三个数组：用于存储所有粒子的邻域粒子编号的数组 C，存储各粒子的邻域在数组 C 中位置的数组 A，以及存储粒子邻域数量的数组 B。求解器在执行过程中，只需要简单的计算即可获取已完成查找的粒子 i 的邻域索引。

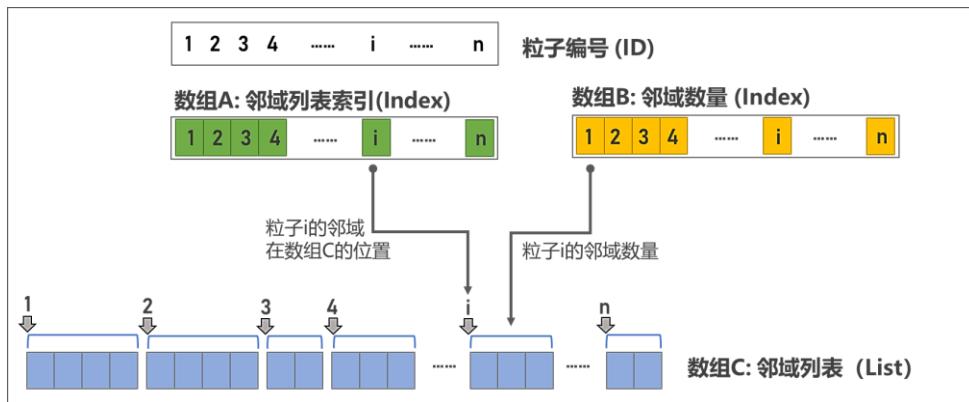


图 5-9 粒子邻域的存储方式。
Figure 5-9 Particle neighborhoods data structure.

5.3 粒子网格混合法求解器

求解器是流体仿真引擎的核心部分，流体的不可压缩性、粘性、表面张力等特性的模拟计算都需要使用求解器来实现。由于本文仿真引擎中流体使用粒子来表示，因此流体相关求解器适于使用 SPH 法来实现。此外，为了扩展本文流体仿真引擎的功能，同时进一步探究对偶粒子法与粒子网格混合法的差异性，本文的流体引擎中引入了基于粒子网格混合法的不可压缩性仿真方法。

SPH 法相关的求解算法，在本文前几章已有较为详细的介绍，而粒子网格混

合法的讨论尚未涉及。因此本节将会对流体仿真引擎中的粒子网格混合法求解器进行讨论与介绍。

5.3.1 粒子网格混合法概述

粒子网格混合法是一种结合欧拉视角网格与拉格朗日视角粒子方法（图 5-10）。在粒子网格混合法中，其压强投影计算过程与欧拉网格方法中的投影计算过程是完全相同的；其粒子主要用于追踪流体的位置，同时携带流体的速度等物理量。

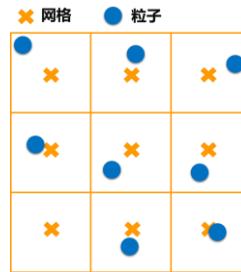


图 5-10 粒子网格混合法。

Figure 5-10 Hybrid particle-grid method

5.3.2 粒子网格混合法中的对流计算

参考 Fei 等人的研究^[35]，若根据粒子与网格之间速度插值方法做区分，可以将粒子网格混合法划分为三类：PIC 法、FLIP 法、NFLIP 法，以下是这三种方法的讨论及其实现方式。

(1) Particle In Cell (PIC)。这类方法中，速度和动量会直接在粒子与网格之间进行插值^[82]。经过 P2G (Particle-to-grid, 粒子到网格的插值) 插值后，其速度不会再被保留在粒子上；类似地，经过 G2P (Grid-to-particle, 网格到粒子的插值) 插值后，网格上的信息也会被删除。由于这种方会引入严重的数值耗散，因此更高阶精度的插值方法被陆续引入到 PIC 方法中，以减小插值过程中引入的耗散，例如 APIC (Affine PIC) 方法^[84]、Poly-PIC 方法^[85]等。此处以标准的 PIC 法为例，P2G 与 G2P 过程的计算公式为^[35]：

$$\text{PIC P2G:} \quad m_g \mathbf{u}_g^n = \sum_p w_{gp} m_p \mathbf{u}_p^n \quad (5-2)$$

$$\text{PIC G2P:} \quad \begin{cases} \mathbf{u}_p^{n+1} = \sum_g w_{gp} \mathbf{u}_g^* \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \delta t \sum_g w_{gp} \mathbf{u}_p^* \end{cases} \quad (5-3)$$

其中 m_p 为粒子的质量， w 为粒子网格混合法中插值函数^[4]， \mathbf{u}_g^* 为经过不可压缩性计算后得到的无速度散度误差的网格速度， \mathbf{x} 为粒子的位置。

(2) Fluid-Implicit-Particles (FLIP)。 FLIP 法是一种对网格速度与粒子速度

进行混合的技术。该方法中粒子上的高频速度场在 G2P 过程中被部分保留下来，以降低 PIC 法中因频繁在粒子与网格间插值而引入的数值耗散^[86]。FLIP 法的 P2G 与 G2P 过程的方程如下。

$$\text{FLIP P2G:} \quad m_g \mathbf{u}_g^n = \sum_p w_{gp} m_p \mathbf{u}_p^n \quad (5-4)$$

$$\text{FLIP G2P:} \quad \begin{cases} \mathbf{u}_p^{n+1} = \sum_g w_{gp} \mathbf{u}_g^* + \alpha \left(\mathbf{u}_g^n - \sum_p w_{gp} m_p \mathbf{u}_p^n \right) \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \delta t \sum_g w_{gp} \mathbf{u}_p^* \end{cases} \quad (5-5)$$

其中， α 为控制 FLIP 与 PIC 之间速度混合强度的系数，该系数决定了保留在粒子上高频速度的比例，如果 α 非常接近于 1，那么上述方法可以有效地降低流体仿真过程中数值耗散，但与此同时粒子的运动也会更加的混乱^[84]。

(3) **Naturally-modified FLIP (NFLIP)**。参考 Fei 等人的研究^[35]，NFLIP 的 G2P 与 P2G 操作可以用下述方程表示^[4]。

$$\text{NFLIP P2G:} \quad m_g \mathbf{u}_g^n = \sum_p w_{gp} m_p \mathbf{u}_p^n \quad (5-6)$$

$$\text{NFLIP G2P:} \quad \begin{cases} \mathbf{u}_p^{n+1} = \sum_g w_{gp} \mathbf{u}_g^* + \alpha \left(\mathbf{u}_g^n - \sum_p w_{gp} m_p \mathbf{u}_p^n \right) \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \delta t \sum_g w_{gp} \mathbf{u}_p^{n+1} \end{cases} \quad (5-7)$$

与标准的 FLIP 法相比较，NFLIP 最主要的区别是使用 \mathbf{u}_p^{n+1} 来更新粒子位置，而非 \mathbf{u}_p^* 。由于粒子上的高频速度也同时参与了粒子位置的更新，因此相比于 FLIP 法，NFLIP 可以进一步降低数值耗散。若上式中的系数 α 取值设置为 1 时，则在 G2P 过程中，网格向粒子插值的不再是速度场，而是速度场的增量，其本质为不可压缩性求解获得的压强力。

5.3.3 粒子网格混合法中的不可压缩性

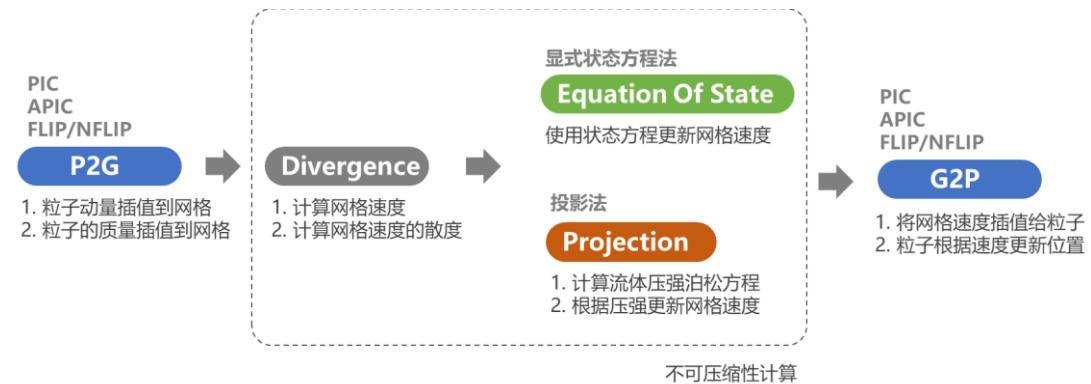


图 5-11 粒子网格混合法的实现方式。

Figure 5-11 The implementation of hybrid particle-grid method.

粒子网格混合法求解器中包含有两部分计算（如上图所示），第一部分计算为对流计算，即上节中介绍的 P2G 与 G2P 操作；第二部分计算为不可压缩性计算，不可压缩性计算的实现方法有两类，第一类是基于网格的投影计算^[70]；第二类是采用显式状态方程的计算。以下是这部分算法实现方式的介绍。

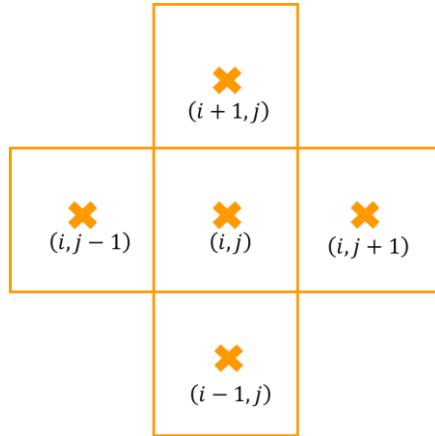


图 5-12 二维网格的索引。

Figure 5-12 Index of a 2-d Eulerian grid.

(1) **投影法：**网格投影法实现流程与第 4 章讨论的 SPH 投影法基本相同，即首先根据压强泊松方程求解出压强，然后根据压强更新流体速度。以 2 维空间为例，基于网格的速度散度计算方法如下：

$$\nabla_{(i,j)} \cdot \mathbf{u} = \frac{1}{2} \left(\frac{\mathbf{u}_{(i+1,j)} - \mathbf{u}_{(i-1,j)}}{\Delta x} + \frac{\mathbf{u}_{(i,j+1)} - \mathbf{u}_{(i,j-1)}}{\Delta x} \right) \quad (5-8)$$

其中 Δx 为网格间距；为便于理解，本小节中下标 (i,j) 表示网格的索引（图 3-12）。

基于网格的压强拉普拉辛离散化方法如下：

$$\nabla_{(i,j)} \cdot \nabla p = \left(\frac{4p_{(i,j)} - p_{(i+1,j)} - p_{(i-1,j)} - p_{(i,j+1)} - p_{(i,j-1)}}{\Delta x^2} \right) \quad (5-9)$$

施加了狄雷克利边界条件后，基于上述的离散化方法即可实现压强泊松方程的求解，求解出压强后即可更新网格上的流体速度：

$$\mathbf{u}_{(i,j)}^{new} = \begin{pmatrix} \frac{p_{(i+1,j)} - p_{(i-1,j)}}{2\Delta x} \\ \frac{p_{(i,j+1)} - p_{(i,j-1)}}{2\Delta x} \end{pmatrix} \quad (5-10)$$

(2) **显式状态方程法：**该求解器算法的设计借鉴了 Tampubolon 等人^[174]的方法。尽管投影法能够获得较为精确的计算结果和较好的稳定性，但计算需要进行迭代，代码实现相对复杂^[70]。相较之下，基于显式状态方程的方法则更为简单^[174]。这一方法中无需计算压强泊松方程，只需如下公式即可直接获得网格上的流体压强。

$$p_{(i,j)}^{new} = E(\nabla_{(i,j)} \cdot \mathbf{u}) \quad (5-11)$$

上式中的 E 为流体的刚度系数， $\nabla_{(i,j)} \cdot \mathbf{u}$ 为网格位置上流体的速度散度。该方法中的刚度系数取值过大或过小都可能造成算法的不稳定，需要根据仿真场景选取合适的值。经过上述方法获得了网格上的压强后，即可根据公式(5-10)计算满足不可压缩性条件的网格速度，然后将网格速度插值回粒子，完成粒子的速度和位置的更新，使流体粒子的速度满足无散度约束条件。

然而，由于上述两种不可压缩性求解方法中不包含流体密度的修正项，因此在模拟过程中很容易发生体积丢失的问题。本文中 4.4.1 节中用于消除漂移误差的方法可以改善这一问题。当网格上的流体密度大于流体静止时的密度 ρ_0 时，网格上会产生相应抵抗压缩的压强。在显式状态方程法中与投影法中，均可使用如下的方程来补偿速度散度，可避免因密度误差积累导致流体发生严重的体积丢失问题。

$$(\nabla_{(i,j)} \cdot \mathbf{u})^{new} = (\nabla_{(i,j)} \cdot \mathbf{u})^{old} + \kappa \frac{\max(\rho_{(i,j)} - \rho_0, 0)}{\rho_0 \Delta t} \quad (5-12)$$

其中 κ 为一常数，网格密度 $\rho_{(i,j)}$ 可直接使用原对偶粒子法中的虚粒子密度来计算。

由于粒子网格混合法中执行对流计算的主体是粒子，因此基于 SPH 粒子的粘性计算、表面张力计算、与固体边界之间的碰撞的相关计算等，均可直接施加于粒子网格混合法模拟的流体之上，并可以产生稳定的结果。

5.4 虚粒子采样方法与稀疏空间网格

由于本文上一章提出的对偶粒子框架具有较好的可扩展性，因此在本章流体仿真引擎将以对偶粒子方法作为仿真部分的核心框架，通过设置不同的虚粒子采样方法实现不同方法下的流体仿真，获得不同的流体运动特性。参考本文 4.5 节的讨论，三种虚粒子采样策略对应的仿真方法讨论如下：

第一种是使实粒子与虚粒子的位置重合，即共位置策略（策略 A）。使用这种采样策略时，对偶粒子方法完全等价于标准的 SPH 法。此时可适当调整粒子邻域查找模块，避免因虚粒子的存在而引入额外计算资源。

第二种是通过微调虚粒子位置，获得更均匀分布的策略，即：粒子微移策略（策略 B）。使用此策略时，流体会在不引入过多虚粒子的情况下，获得比标准的 SPH 法更稳定的仿真结果，而且压强计算的收敛速度也相对更快。

第三种是在实粒子附近生成网格形态的虚粒子，即：空间自适应策略（策略 C）。此时对偶粒子法具有较好地稳定性，可以稳定地模拟拉伸形态的流体，而且该种仿真策略可适配于粒子网格混合法，在不花费太多的存储空间的情况下作为 PIC、FLIP、NFLIP 法的背景网格。相比于采用八叉树的方法，本文对偶粒子策略 C 生成的空间稀疏网格结构不存在较难处理的 T-Junction 问题^[13, 15, 175]，因此无需对 4.5.2 节中粒子网格混合法进行修改，即可直接与之适配。

需要注意的是，策略 C 生成的空间稀疏结构网格并不能根据其索引直接判断虚粒子位置，因此查找这类网格的邻域网格时，也需要用到 SPH 法中的邻域查找算法。

5.5 粒子网格混合法的研究

在实现流体仿真引擎的搭建后，搭配使用不同求解器构建流体仿真场景的操作就会变得较为简洁，相应地也可更合理、更直接地对比研究不同流体仿真方法的特性。本节将使用上述的流体仿真方法，在完全相同的条件下对比研究本文对偶粒子法与粒子网格混合法之间的区别；由于本文的流体仿真引擎中很容易实现不同仿真算法的组合，因此在本节的研究中，使用了 SPH 法粒子位置修正技术（Particle Shifting）改善了在粒子网格混合法中的粒子分布不理想的问题。

5.5.1 对偶粒子法与粒子网格混合法的对比研究

基于上文的讨论与分析可知，本文采用策略 C 的对偶粒子法与 PIC、FLIP 等粒子网格混合法^[35]相比较，两者具有一定的相似性，即：对偶粒子法策略 C 中的虚粒子类似于粒子网格混合法中的欧拉网格。然而这两类方法在本质上是非常不同的。本小节在相同条件下使用两组方法模拟了流体的运动，并对测试结果进行了研究讨论，分析导致其仿真效果不同的原因。

图 5-13 使用本文流体仿真引擎模拟了流体溃坝场景，测试中粒子网格混合法与对偶粒子法所使用的各项参数条件保持了一致，如重力加速度、粒子与固体边界的碰撞实现方法、下落时的初速度等。此外所有方法的收敛阈值均设置为最大误差的 10^{-4} ；粒子网格混合法的网格分辨率与对偶粒子法的虚粒子分辨率相同；所有方法均没有使用人工粘性。测试结果显示粒子网格混合法模拟流体运动过程中，粒子的分布普遍不够均匀，而且表现出了较大的数值粘性。尽管将 NFLIP 的系数 α 设置为 1.0 后，能够观察到其数值粘性有所降低，但其粒子的运动变得非常的杂乱^[35]。相比较之下，策略 C 获得了非常均匀的粒子分布，流体运动过程中也展示出了更为稳定的薄层形态。

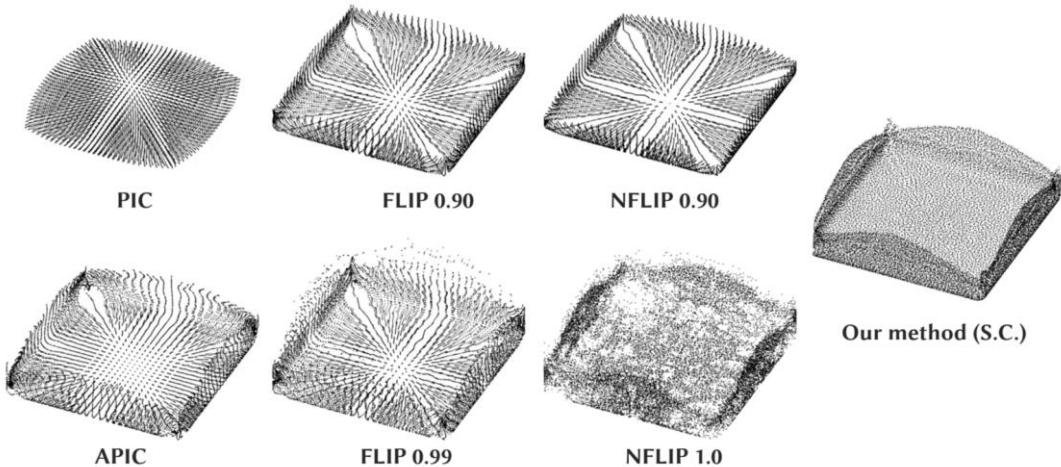


图 5-13 策略 C 与粒子网格混合法的对比。

Figure 5-13 Comparison of our method with S.C. and hybrid particle-grid methods.

理论上粒子网格混合法不存在 SPH 法中的拉伸不稳定问题，但上述场景的测试结果显示，对偶粒子法模拟的流体，其粒子分布更均匀，流体的薄层形态能够更稳定地维持。导致两类方法模拟结果相差较大的原因是多方面的，具体如下。

(1) 速度散度计算不同。对偶粒子法中，不需要使用类似于 P2G 将速度或动量从粒子插值到网格上的操作，其速度散度的计算结果虽然存储在虚粒子上，但速度散度是直接使用实粒子计算的，本质上为实粒子的散度（式(4-14)）；而在粒子网格混合法中，流体速度首先需要插值到网格上，然后在网格上计算获得速度散度，因此其本质为网格速度的散度。

(2) 压强梯度的计算不同。在对偶粒子法中，实粒子位置上压强的梯度是由虚粒子直接计算获得的（式(4-19)），然后直接用来更新实粒子的速度；而在粒子网格混合法中，压强梯度完全被定义在了网格上，并用来更新网格上的速度，待投影计算完成后才会将网格上的速度通过 G2P 过程插值回粒子。计算过程不仅繁琐，而且可能会引入耗散。

(3) 压强泊松方程系数矩阵的元素计算方法不同。在对偶粒子法中，虚粒子携带有密度、体积等物理量，这与标准 SPH 法中的粒子较为相似。因此使用虚粒子离散的压强泊松方程系数矩阵中的元素，均包含有流体体积等相关信息（式(4-11)）。而在粒子网格混合法中，其压强泊松方程使用欧拉网格投影法实现，远离流体边界的网格对应的元素均为相同值，其中包含的信息少于对偶粒子方法（式(5-9)）。

图 5-14 为标准 SPH 投影法、对偶粒子法、粒子网格混合模拟流体时的管线图对比。相比较之下，本文的对偶粒子法在操作步骤方面要比粒子网格混合法更简便。

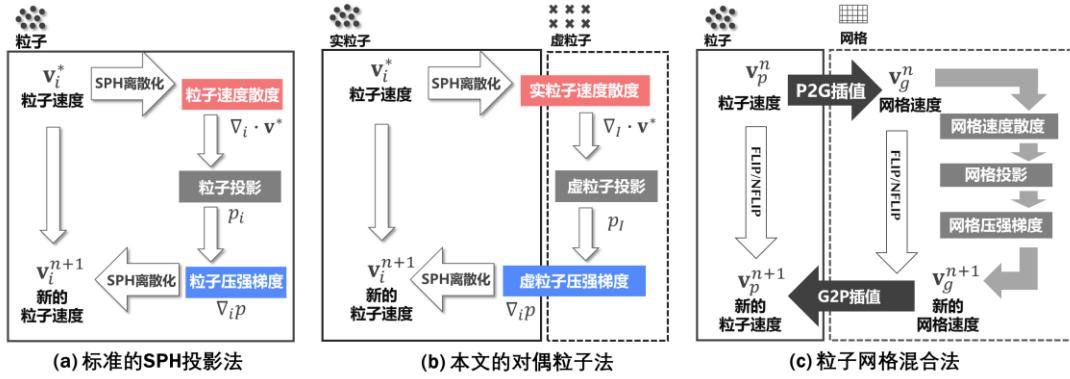


图 5-14 标准 SPH 投影法、本文的对偶粒子法与粒子网格混合法的管线图（Pipeline）对比。

Figure 5-14 The pipeline of the original SPH projection method, our dual-particle approach and the hybrid particle-grid method.

5.5.2 粒子-网格混合法中粒子杂乱问题的改进方法

在上一节中的对比测试中（图 5-16），可发现粒子网格混合法中粒子分布是较为杂乱的。这是由于粒子网格混合法中缺少粒子之间直接的相互作用，导致粒子相聚过近时粒子间不能产生合适的排斥力。该问题在模拟小规模场景时会表现的更为明显。为改善这一问题，Ando 等人提出在粒子之间施加微弱弹簧力（Weak spring force）^[176]，但这一方法只能在粒子间提供有限的斥力，不能立即改善粒子分布不均问题；而若将该力调大，则可能会引起粒子间的震荡。Um 等人则提出，在粗网格之上叠加使用更高分辨率的网格的方法，该方法使用高分辨率网格来修正粒子间的位置以改善粒子分布^[177]。但这一方法依然不在粒子间产生直接斥力，因此其效果有赖于细网格的高分辨率。

SPH 法中，改善粒子分布的方法已经较为成熟，其中较为常用的方法是粒子位置修正技术（Particle Shifting）^[91, 156, 178, 179]，该技术通过菲克定律（Fick Law）来实现粒子位置的修正，即粒子总是从高浓度位置流向低浓度位置。该方法中粒子的修正量计算可视为能量优化问题，其中一种常用的能量函数为：

$$E_i = \frac{1}{2} \frac{\|\mathbf{x}_i - \mathbf{x}_i^{old}\|^2}{d_0^2} + \frac{\lambda}{4} \left(\frac{\rho_i}{\rho_0} - 1 \right)^2 \quad (5-13)$$

上式中的 λ 为常数。使用上述公式作为能量函数的方法中，粒子每次迭代的位置修正量即为：

$$\delta \mathbf{x}_i = \zeta \nabla_i E \quad (5-14)$$

其中， ζ 为控制粒子位置修正量取值的系数。

不同的流体仿真算法具有不同的特性，基于本文的流体仿真引擎可以将两种相差较大的流体算法相互结合使用，通过取长补短的方式明显改善流体的仿真质量。因此在本文流体仿真引擎中，可以将 SPH 法中的粒子位置修正技术（Particle

shifting) [91, 156, 178, 179] 引入到粒子网格混合法中来解决 PIC 法、FLIP 法等方法中粒子分布不均的问题。

图 5-16 为使用本文流体仿真引擎模拟的粒子网格混合法流体，该场景在本文流体仿真引擎中的搭建方式如图 5-15 所示。在该场景中，流体粒子采用等间距均匀采样的方式进行初始化。图 5-15(a)展示的结果表明，PIC 法^[180]、APIC 法^[84]、FLIP 法^[181]、NFLIP^[35]法模拟的流体粒子会逐渐变得不均匀（图 5-16(a)）。而在使用了 SPH 法中的粒子位置修正技术之后，流体粒子的分布被明显地改善（图 5-16(b)）。

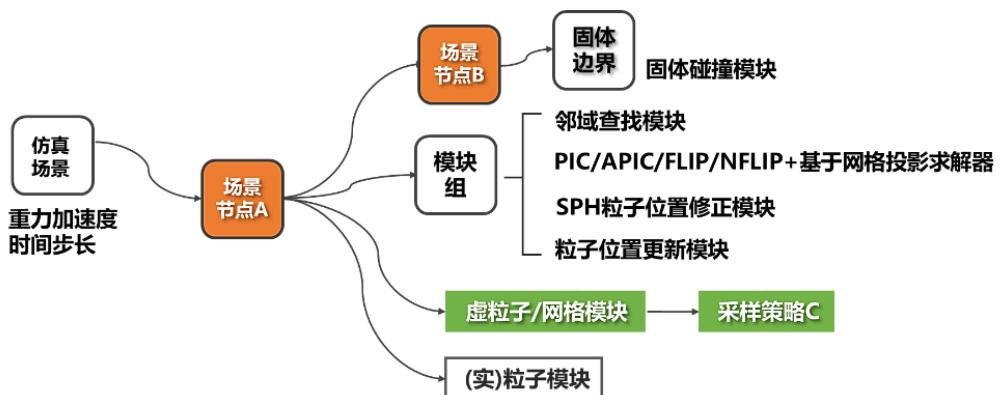


图 5-15 粒子网格混合法测试场景（图 5-16）的节点图。

Figure 5-15 The node diagram of the hybrid particle-grid method (Figure 5-13).

5.6 本章总结

本章基于对偶粒子方法的架构设计了能够兼容标准 SPH 法、对偶粒子法以及粒子网格混合法的通用粒子法流体仿真引擎。本章介绍了该引擎的架构、仿真场景的搭建方法、仿真任务的实现流程。由于对偶粒子法中的策略 C 的虚粒子采样方法类似于具有空间稀疏结构的欧拉网格，因此可以将其作为粒子网格混合法中的背景网格，而且相比于常用的八叉树网格，策略 C 获得的网格不存在 T-Junction 的问题。基于这种通用的仿真引擎便于实现不同流体模拟方法的对比，因此本章对比研究了对偶粒子法与粒子网格混合法的区别。在测试中发现，由于对偶粒子法使用了完全不同的离散化与插值方案，因此获得的流体粒子的分布更加均匀。为改善粒子网格混合法中粒子分布不理想的问题，本节引入了 SPH 法中的粒子修正技术，并将其与粒子网格混合法结合，有效地改善了粒子网格混合法的仿真质量。

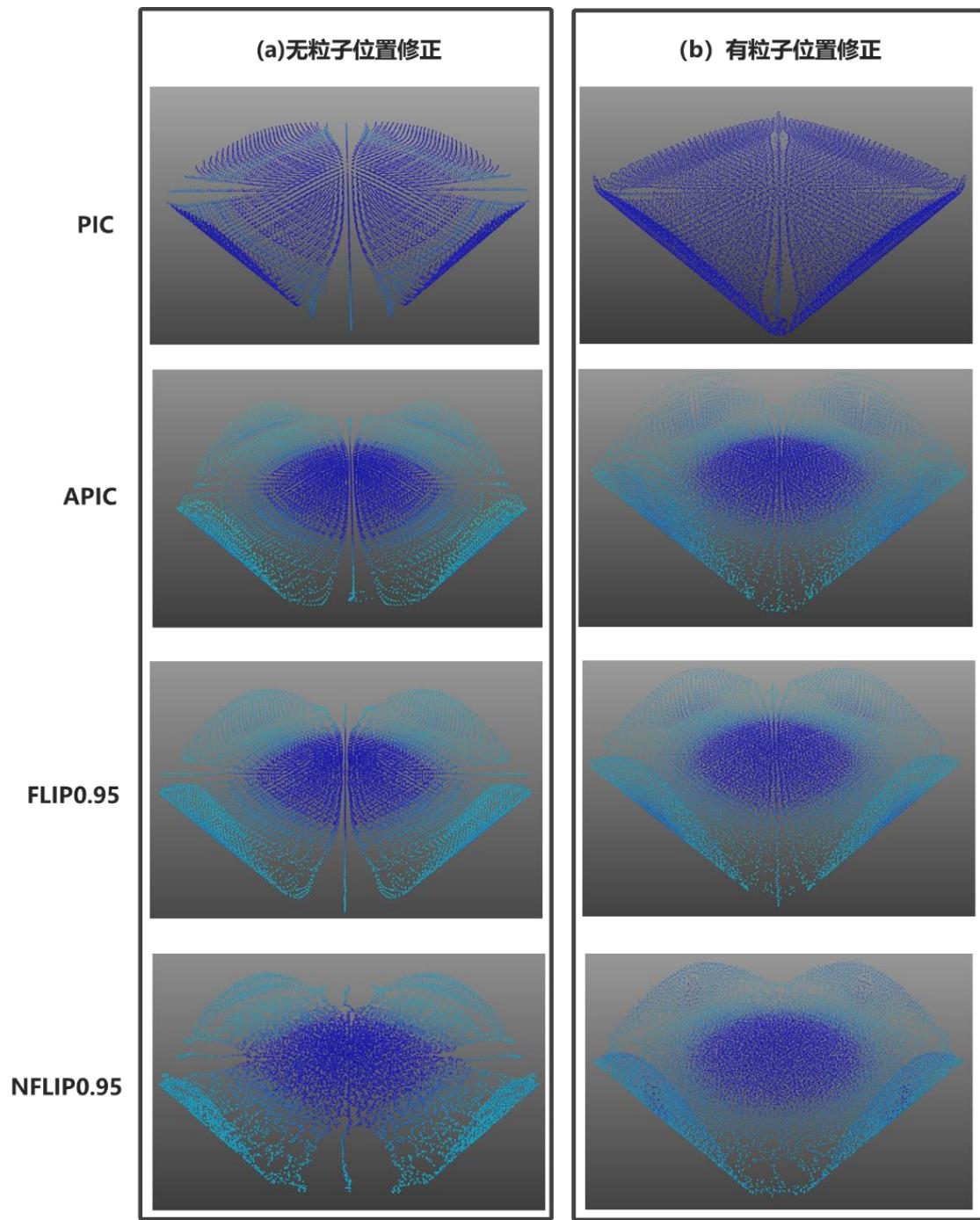


图 5-16 粒子网格混合法模拟的流体在使用粒子修正技术之前与之后的对比。

Figure 5-16 Comparison of fluids simulated by hybrid particle-grid method before and after using particle shifting technique.

第6章 总结与展望

6.1 本文总结

SPH 法是计算机图形学领域中实现流体仿真的主要技术之一。虽然发展多年，但 SPH 法依然存在着数值计算不稳定问题，导致其模拟大粘性流体、流体的拉伸状态等场景时稳定性较差。为了进一步拓展 SPH 法的应用潜力，推动物理仿真技术的发展，本文从数值计算稳定性方面对 SPH 法展开研究。本文每章内容的总结如下：

在第一章中，本文首先介绍了 SPH 流体的研究意义，讨论了 SPH 法模拟流体粘性、不可压缩性研究的发展及其存在的问题，然后简要说明了本文的研究内容与组织结构。

在第二章中，本文介绍了连续介质力学中有关流体不可压缩性、粘性方面的基础理论，然后介绍了使用 SPH 法实现流体仿真的基本方法及其离散化方法。

在第三章中，本文研究了 SPH 法中粘性与不可压缩性之间的冲突问题，并首次提出了适用于粘性不可压缩 SPH 流体的 SIMPLE 迭代算法。该方法对不可压缩性与粘性计算进行总体迭代，可逐渐减轻两者之间的相互干扰，使大粘性流体能够产生更锐利、更丰富的表面细节，更稳定地保持形态。此外，本章也研究了非牛顿流体的高效仿真方法，实现了基于 Cross 模型的剪切变稀流体、剪切增稠流体在大时间步长下的稳定仿真。

在第四章中，本文研究了 SPH 流体的拉伸稳定性问题，首次提出了可有效抑制拉伸不稳定问题的对偶粒子架构，并构建了基于对偶粒子的投影计算方法，该方法可大幅度提高流体小尺度运动细节模拟的稳定性。研究中发现，作为压强采样点的虚粒子，其分布的均匀性对于提升 SPH 流体的数值稳定性方面非常重要。因此，本章设计了基于 GPU 快速生成空间紧凑自适应分布虚粒子的方法，以高效高质量地实现稳定的流体仿真。

在第五章的研究中，本文基于对偶粒子法提出了一种通用的流体仿真引擎。该仿真引擎可兼容标准 SPH 法、对偶粒子法与粒子网格混合法，文章介绍了该流体仿真引擎中的实现架构、仿真场景的搭建方式与仿真管线的设计，并探讨了 PIC、FLIP、NFLIP 等粒子网格混合法求解器的实现方法。研究中发现，对偶粒子方法中的空间自适应虚粒子生成策略，非常适合作为粒子网格混合法的背景网格。由于粒子网格混合法与本文提出的对偶粒子法有一定相似性，本章使用该仿真引擎对比研究了两者的区别，试验发现粒子网格混合法中的粒子分布非常不均匀。为改善此问题，本节引入了 SPH 法中的粒子修正技术，并将其与粒子网格混合法结合，有效地改善了粒子网格混合法的仿真质量。这一研究凸显了开发通用粒子仿真引擎的必要性。

6.2 未来展望

本文对 SPH 流体中不可压缩性与粘性间的干扰，以及拉伸不稳定等数值不稳定问题展开了研究，并取得了较好的进展。但是在本文的研究工作中尚存在着一些问题，有待于进一步完善。首先，基于 SIMPLE 迭代的策略尽管能够提升大粘性流体的仿真质量，但由于其迭代收敛较慢，而且每次迭代的计算量较大，导致计算较为耗时；第二，基于对偶粒子法的研究，虽然能够有效提升 SPH 流体的拉伸稳定性，但该方法无法实现核函数归一化，导致其计算精度不够高；第三，为了获得较好的仿真质量，对偶粒子法计算过程中需要引入较多的虚粒子，而这也会导致仿真效率较差；第四，本文第五章提出的通用流体仿真引擎，现阶段只能用于仿真流体材料，能够应对的仿真场景过于局限。针对本文研究中尚未解决的问题，并扩展本文研究的深度与广度，我们计划在未来的研究工作中朝着以下几方面继续探索：

(1) 针对基于 SPH 法的 SIMPLE 迭代算法收敛速度较慢的问题，可尝试通过结合 Projective Dynamics 方法^[108, 182]，改进 SIMPLE 迭代策略，以及加速其收敛速度。如果能够构建出粘性不可压缩性流体的统一求解器，理论上可以完全避免粘性与不可压缩性之间的冲突，同时有效提高算法的计算速度。

(2) 仿真计算中不同求解步骤之间的相互干扰问题是普遍存在的，比如粘弹性流体中，粘性、弹性、不可压缩性之间也存在着相互干扰问题。而本文提出的 SIMPLE 迭代策略，理论上可以扩展到更广泛的领域中，解决更复杂场景与材料模拟的数值稳定性问题。

(3) 由于对偶粒子法在模拟流体时，会引入大量的虚粒子，这导致粒子邻域查找过程较为耗时。如果能够针对虚粒子的空间特性对邻域查找算法进行优化，比如结合八叉树网格、空间哈希表等技术，则可以大幅度提高对偶粒子法的仿真效率。

(4) 对偶粒子法的研究中，现阶段压强泊松方程与动量方程的离散化方法使用的均为零阶一致性方案。如果在离散化过程中使用更高阶精度的计算方法，比如引入核归一化条件，结合移动最小二乘法、RBF 法等，理论上可以进一步提升对偶粒子法的数值计算稳定性。

(5) 对偶粒子法已被证明能够有效抑制 SPH 法流体的拉伸不稳定问题，而且相比于粒子网格混合法计算过程更加的简洁、稳定，因此可以考虑将其进一步扩展到粘性、弹性、塑性、边界处理等多方面的研究工作中，进而建立起一套完整、高效、稳定、全新的物理仿真体系，并以此为基础建立更完善的物理仿真引擎。

参考文献

- [1] Alduán I, Tena A , Otaduy M A . . DYVERSO: A Versatile Multiphase Position-Based Fluids Solution for VFX [J]. Computer Graphics Forum, 2016, 36(6): [10.1111/cgf.12992].
- [2] Ueda K, Fujishiro I. Adsorptive SPH for directable bleeding simulation [Z]. Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry - VRCAI '15. 2015: 9-16.10.1145/2817675.2817684
- [3] Hu Y, Li T-M, Anderson L, et al. Taichi [J]. ACM Transactions on Graphics, 2019, 38(6): 1-16.
- [4] Stomakhin A, Schroeder C, Chai L, et al. A material point method for snow simulation [J]. ACM Transactions on Graphics, 2013, 32(4).
- [5] Boyd L, Bridson R. MultiFLIP for energetic two-phase fluid simulation [J]. ACM Transactions on Graphics, 2012, 31(2): 1-12.
- [6] 肖祥云, 杨旭波. 基于物理及数据驱动的流体动画研究 [J]. 软件学报, 2020, 31(10): 15.
- [7] Chui Y P, Heng P A. A Particle-based Modeling Framework for Thrombo-emboli Simulation [J]. Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry - VRCAI 2012.
- [8] Neal E. Seymour, Anthony G. Gallagher, Sanziana A. Roman, et al. Virtual Reality Training Improves Operating Room Performance [J]. Annas of Surgery, 2002, 236(4): 458-64.
- [9] Huang L, Hädrich T, Michels D L. On the accurate large-scale simulation of ferrofluids [J]. ACM Transactions on Graphics, 2019, 38(4): 93.
- [10] Gissler C, Peer A, Band S, et al. Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling [J]. ACM Transactions on Graphics, 2019, 38(1): 5.
- [11] Barreiro H, García-Fernández I, Alduán I, et al. Conformation constraints for efficient viscoelastic fluid simulation [J]. ACM Transactions on Graphics, 2017, 36(6): 1-11.
- [12] Gotoh H, Khayyer A. Current achievements and future perspectives for projection-based particle methods with applications in ocean engineering [J]. Journal of Ocean Engineering and Marine Energy, 2016, 2(3): 251-78.
- [13] Xiao Y, Chan S, Wang S, et al. An adaptive staggered-tilted grid for incompressible flow simulation [J]. ACM Transactions on Graphics, 2020, 39(6): 171.
- [14] Chen Y-L, Meier J, Solenthaler B, et al. An extended cut-cell method for sub-grid liquids tracking with surface tension [J]. ACM Transactions on Graphics, 2020, 39(6): 169.

- [15] Goldade R, Wang Y, Aanjaneya M, et al. An adaptive variational finite difference framework for efficient symmetric octree viscosity [J]. ACM Transactions on Graphics, 2019, 38(4): 94.
- [16] Macklin M, Müller M. Position based fluids [J]. ACM Transactions on Graphics, 2013, 32(4): 104.
- [17] Winchenbach R, Akhunov R, Kolb A. Semi-analytic boundary handling below particle resolution for smoothed particle hydrodynamics [J]. ACM Transactions on Graphics, 2020, 39(6): 173.
- [18] Zhai X, Hou F, Qin H, et al. Fluid Simulation with Adaptive Staggered Power Particles on GPUs [J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 26(6): 2234-46.
- [19] Chen Y, Li W, Fan R, et al. GPU Optimization for High-Quality Kinetic Fluid Simulation [J]. IEEE Transactions on Visualization and Computer Graphics, 2021: [10.1109/TVCG.2021.3059753].
- [20] Hu Y, Fang Y, Ge Z, et al. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling [J]. ACM Transactions on Graphics, 2018, 37(4): 150.
- [21] Nakanishi R, Nascimento F, Campos R, et al. RBF liquids: an adaptive PIC solver using RBF-FD [J]. ACM Transactions on Graphics, 2020, 39(6): 170.
- [22] Fang Y, Qu Z, Li M, et al. IQ-MPM: an interface quadrature material point method for non-sticky strongly two-way coupled nonlinear solids and fluids [J]. ACM Transactions on Graphics, 2020, 39(4): 51.
- [23] Müller M, Schirm S, Teschner M, et al. Interaction of fluids with deformable solids [J]. Computer Animation and Virtual Worlds, 2004, 15(34): 159-71.
- [24] Keiser R, Adams B, Gasser D, et al. A Unified Lagrangian Approach to Solid-Fluid Animation; proceedings of the Eurographics Symposium on Point-Based Graphics, F, 2005 [C]. The Eurographics Association.
- [25] Cornelis J, Ihmsen M, Peer A, et al. IISPH-FLIP for incompressible fluids [J]. Computer Graphics Forum, 2014, 33(2): 255-62.
- [26] Hong J-M, Lee H-Y, Yoon J-C, et al. Bubbles alive [J]. ACM Transactions on Graphics, 2008, 27(3): 1.
- [27] Thürey N, Sadlo F, Schirm S, et al. Real-time simulations of bubbles and foam within a shallow-water framework.; proceedings of the Acm Siggraph/eurographics Symposium on Computer Animation ACM, F, 2007 [C].
- [28] Losasso F, Talton J, Kwatra N, et al. Two-way coupled SPH and particle level set fluid simulation [J]. IEEE Trans Vis Comput Graph, 2008, 14(4): 797-804.
- [29] Lee H, Han S. Solving the Shallow Water equations using 2D SPH particles for interactive applications [J]. The Visual Computer, 2010, 26(6-8): 865-72.
- [30] Kang N, Sagong D. Incompressible SPH using the Divergence-Free Condition [J]. Computer Graphics Forum, 2014, 33(7): 219-28.

- [31] Shi J, Li C, Wang C, et al. An advanced hybrid smoothed particle hydrodynamics–fluid implicit particle method on adaptive grid for condensation simulation [J]. Computer Animation and Virtual Worlds, 2020, 31(6): 1546-4261.
- [32] Shao X Q, Mei P, Chen W X. Real-time simulation of realistic fluid animation based on stable SPH-SWE numerical model [J]. Acta Physica Sinica, 2021, 70(23).
- [33] 邵绪强, 梅鹏, 陈文新. 基于稳定性 SPH-SWE 数值模型的真实感流体动画实时模拟 [J]. 物理学报, 2021, 70(23): 12.
- [34] Liu G R, Liu M B. Smoothed particle hydrodynamics: a meshfree particle methods [M]. 2003.
- [35] Fei Y, Guo Q, Wu R, et al. Revisiting integration in the material point method: a scheme for easier separation and less dissipation [J]. ACM Transactions on Graphics, 2021, 40(4): 109.
- [36] Akbay M, Nobles N, Zordan V, et al. An extended partitioned method for conservative solid-fluid coupling [J]. ACM Transactions on Graphics, 2018, 37(4): 86.
- [37] Ren B, Li C, Yan X, et al. Multiple-Fluid SPH Simulation Using a Mixture Model [J]. ACM Transactions on Graphics, 2014, 33(5): 171.
- [38] Yang T, Chang J, Ren B, et al. Fast multiple-fluid simulation using Helmholtz free energy [J]. ACM Transactions on Graphics, 2015, 34(6): 201.
- [39] Yan X, Jiang Y-T, Li C-F, et al. Multiphase SPH simulation for interactive fluids and solids [J]. ACM Transactions on Graphics, 2016, 35(4): 79.
- [40] Yang T, Chang J, Lin M C, et al. A unified particle system framework for multiphase, multi-material visual simulations [J]. ACM Transactions on Graphics, 2017, 36(6): 224.
- [41] Chang Y, Bao K, Liu Y, et al. A particle-based method for viscoelastic fluids animation [M]. 2009.
- [42] Kim J H, Lee J. Stable and Anisotropic Freezing Framework With Interaction Between IISPH Fluids and Ice Particles [J]. IEEE Access, 2021, 9: 146097-109.
- [43] Jiang Y, Lan Y. A Dynamic Mixture Model for Non-equilibrium Multiphase Fluids [J]. Computer Graphics Forum, 2021, 40(7): 85-95.
- [44] 廖惟博, 赵霄洋, 张繁, 等. 基于红外物理学和流体动力学的海面船只红外尾迹真实感绘制 [J]. 计算机辅助设计与图形学学报, 2017, 29(7): 8.
- [45] 郑顾平, 张萌, 邵绪强, 等. 真实感水力侵蚀地形建模与绘制 [J]. 图学学报, 2020, 41(2): 9.
- [46] Xu X, Yu P. A technique to remove the tensile instability in weakly compressible SPH [J]. Computational Mechanics, 2018, 62(5): 963-90.
- [47] Chalk C M, Pastor M, Peakall J, et al. Stress-Particle Smoothed Particle Hydrodynamics: An application to the failure and post-failure behaviour of slopes [J]. Computer Methods in Applied Mechanics and Engineering, 2020, 366.
- [48] Vacondio R, Altomare C, De Leffe M, et al. Grand challenges for Smoothed

- Particle Hydrodynamics numerical schemes [J]. Computational Particle Mechanics, 2021, 8(3): 575-88.
- [49] Larionov E, Batty C, Bridson R. Variational Stokes: A Unified Pressure-Viscosity Solver for Accurate Viscous Liquids [J]. ACM Transactions on Graphics, 2017, 36(4): 101.1-.11.
- [50] Lucy L B. A numerical approach to the testing of the fission hypothesis [J]. The Astrophysical Journal, , 1977, 8(12): 1013-24.
- [51] Gingold R A, Monaghan J. Smoothed Particle Hydrodynamics - Theory and Application to Non-Spherical Stars [J]. mnras, 1977, 181: 375-89.
- [52] Joseph J. M. Smoothed Particle Hydrodynamics [J]. Annual Review of Astronomy & Astrophysics, 1992.
- [53] Su Z Y, Wang G Z, Wang Y K, et al. Numerical simulation of dynamic catastrophe of slope instability in three Gorges reservoir area based on FEM and SPH method [J]. Natural Hazards, 2022, 111(1): 709-24.
- [54] Lyu H G, Sun P N. Further enhancement of the particle shifting technique: Towards better volume conservation and particle distribution in SPH simulations of violent free-surface flows [J]. Applied Mathematical Modeling, 2022, 101: 214-38.
- [55] Peng X, Yu P, Chen G, et al. Development of a Coupled DDA-SPH Method and its Application to Dynamic Simulation of Landslides Involving Solid-Fluid Interaction [J]. Rock Mechanics and Rock Engineering, 2020, 53(1): 113-31.
- [56] Hammani I, Marrone S, Colagrossi A, et al. Detailed study on the extension of the δ -SPH model to multi-phase flow [J]. Computer Methods in Applied Mechanics and Engineering, 2020, 368: 113189.
- [57] Desbrun M, Gascuel. M P. Smoothed particles: A new paradigm for animating highly deformable bodies. [J]. Proceedings of the Eurographics Workshop, 1996: 61-7.
- [58] Xu T, Wu W, Wu E. Real-time generation of smoothed-particle hydrodynamics-based special effects in character animation [J]. Computer Animation and Virtual Worlds, 2014, 25(2): 185-98.
- [59] Huang C, Zhu J, Sun H, et al. Parallel-optimizing SPH fluid simulation for realistic VR environments [J]. Computer Animation and Virtual Worlds, 2015, 26(1): 43-54.
- [60] Stanton M, Humberston B, Kase B, et al. Self-refining games using player analytics [J]. ACM Transactions on Graphics, 2014, 33(4): 73.
- [61] Cavelan A, Cabezón R M, Grabarczyk M, et al. A Smoothed Particle Hydrodynamics Mini-App for Exascale [Z]. Proceedings of the Platform for Advanced Scientific Computing Conference. 2020: 1-11.10.1145/3394277.3401855
- [62] Nie X, Hu Y, Su Z, et al. Fluid Reconstruction and Editing from a Monocular Video

- based on the SPH Model with External Force Guidance [J]. Computer Graphics Forum, 2021, 40(6): 62-76.
- [63] Schoentgen A, Poulin P, Darles E, et al. Particle-based Liquid Control using Animation Templates [J]. Computer Graphics Forum, 2020, 39(8): 79-88.
- [64] García-Feal O, Crespo A J C, Gómez-Gesteira M. VisualSPHysics: advanced fluid visualization for SPH models [J]. Computational Particle Mechanics, 2021.
- [65] Li H, Ren H, Qiu S, et al. Physics-Based Simulation of Ocean Scenes in Marine Simulator Visual System [J]. Water, 2020, 12(1): 215.
- [66] Hassaballa H M, Aly A M, Abdelnaim A. Interactive fluid flow simulation in computer graphics using incompressible smoothed particle hydrodynamics [J]. Computer Animation and Virtual Worlds, 2020, 31(7): 1546-4261.
- [67] Ihmsen M, Orthmann J, Solenthaler B, et al. SPH Fluids in Computer Graphics [J]. Eurographics Proceedings, 2014.
- [68] Koschier D, Bender J, Solenthaler B, et al. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids [J]. Eurographics Proceedings, 2019.
- [69] Bo R, Yang X Y, Lin M C, et al. Visual Simulation of Multiple Fluids in Computer Graphics: A State-of-the-Art Report [J]. Journal of Computer Science & Technology, 2018, 33(003): 431-51.
- [70] Bridson R. Fluid Simulation For Computer Graphics [M]. CRC Press, 2014.
- [71] Wicke M, Stanton M, Treuille A. Modular bases for fluid dynamics [J]. ACM Transactions on Graphics, 2009, 28(3): 1-8.
- [72] Cho J, Ko H-S. Geometry-Aware Volume-of-Fluid Method [J]. Computer Graphics Forum, 2013, 32(2pt3): 379-88.
- [73] Batty C, Bridson R. Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids [J]. Eurographics/acm Siggraph Symposium on Computer Animation 2008.
- [74] Takahashi T, Lin M C. A Geometrically Consistent Viscous Fluid Solver with Two - Way Fluid - Solid Coupling [J]. Computer Graphics Forum, 2019, 38(2): 49-58.
- [75] Song O, Shin H, Ko H. Stable But Nondissipative Water [J]. ACM Transactions on Graphics, 2005, 24(1): 81.
- [76] Ferstl F, Ando R, Wojtan C, et al. Narrow Band FLIP for Liquid Simulations [J]. Computer Graphics Forum, 2016, 35(2): 225-32.
- [77] Lyu L, Cao W, Wu E, et al. Extended Narrow Band Weighted MultiFLIP for Two-Phase Liquid Simulation [Z]. The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry. 2019: 1-8.10.1145/3359997.3365685
- [78] Yang L, Li S, Hao A, et al. Hybrid Particle-grid Modeling for Multi-scale Droplet/Spray Simulation [J]. Computer Graphics Forum, 2014, 33(7): 199-208.
- [79] Ram D, Gast T, Jiang C, et al. A Material Point Method for Viscoelastic Fluids,

- Foams and Sponges [J]. Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, 2015.
- [80] Stomakhin A, Schroeder C, Jiang C, et al. Augmented MPM for phase-change and varied materials [J]. ACM Transactions on Graphics, 2014, 33(4): 1-11.
- [81] Su H, Xue T, Han C, et al. A unified second-order accurate in time MPM formulation for simulating viscoelastic liquids with phase change [J]. ACM Transactions on Graphics, 2021, 40(4): 1-18.
- [82] Harlow F H. The particle-in-cell method for numerical solution of problems in fluid dynamics [R]: Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1962.
- [83] Sulsky D, Zhou S J, Schreyer H L. Application of a particle-in-cell method to solid mechanics. [J]. Computer Physics Communications, 1995, 87(1): 236-52.
- [84] Jiang C, Schroeder C, Selle A, et al. The affine particle-in-cell method [J]. ACM Transactions on Graphics, 2015, 34(4): 1-10.
- [85] Fu C, Guo Q, Gast T, et al. A polynomial particle-in-cell method [J]. ACM Transactions on Graphics, 2017, 36(6): 1-12.
- [86] Brackbill J U, Ruppel H M. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions [J]. Journal of Computational physics, 1986, 65(2): 314-43.
- [87] Lai W M. Introduction to Continuum Mechanics [M]. Introduction to continuum mechanics, 2009.
- [88] Yan H, Wang Z, He J, et al. Real-time fluid simulation with adaptive SPH [J]. Computer Animation and Virtual Worlds, 2009, 20(2-3): 417-26.
- [89] Reinhardt S, Huber M, Eberhardt B, et al. Fully asynchronous SPH simulation [Z]. Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation. 2017: 1-10.10.1145/3099564.3099571
- [90] Bender J, Koschier D. Divergence-Free SPH for Incompressible and Viscous Fluids [J]. IEEE transactions on visualization and computer graphics, 2017, 23(3): 1193-206.
- [91] He X, Wang H, Wang G, et al. A variational staggered particle framework for incompressible free-surface flows [J]. arXiv, 2020, 2001.09421.
- [92] Solenthaler B, Pajarola R. Predictive-corrective incompressible SPH [J]. ACM Transactions on Graphics, 2009, 28(3): 40.
- [93] Ma Q, Zhou Y, Yan S. A review on approaches to solving Poisson's equation in projection-based meshless methods for modelling strongly nonlinear water waves [J]. Journal of Ocean Engineering and Marine Energy, 2016, 2: 279–99.
- [94] Matthias Müller C D, Gross M . . Particle-Based Fluid Simulation for Interactive Applications [J]. Acm Siggraph/eurographics Symposium on Computer Animation ACM, 2003.
- [95] Becker M, Teschner M. Weakly compressible SPH for free surface flows [J].

- Eurographics/ ACM SIGGRAPH Symposium on Computer Animation 2007.
- [96] He X, Liu N, Li S, et al. Local Poisson SPH For Viscous Incompressible Fluids [J]. Computer Graphics Forum, 2012, 31(6): 1948-58.
- [97] Bender J, Koschier D. Divergence-free smoothed particle hydrodynamics [Z]. Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation - SCA '15. 2015: 147-55.10.1145/2786784.2786796
- [98] Chorin A J. The numerical solution of the Navier-Stokes equations for an incompressible fluid [J]. Bulletin of the American Mathematical Society, 1967, 73(6): 928-31.
- [99] Cummins S J, Rudman M. An sph projection method. [J]. Journal of Computational Physics, 1999, 152(2): 584-607.
- [100]Raveendrany K, Wojtan C, Turk G. Hybrid Smoothed Particle Hydrodynamics [J]. Acm Siggraph/eurographics Symposium on Computer Animation 2011.
- [101]Ihmsen M, Cornelis J, Solenthaler B, et al. Implicit Incompressible SPH [J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(3): 426-35.
- [102]He X, Liu N, Wang G, et al. Staggered meshless solid-fluid coupling [J]. ACM Transactions on Graphics, 2012, 31(6): 149.
- [103]Takahashi T, Dobashi Y, Nishita T, et al. An Efficient Hybrid Incompressible SPH Solver with Interface Handling for Boundary Conditions [J]. Computer Graphics Forum, 2017, 37(1): 313-24.
- [104]He X, Wang H, Zhang F, et al. Robust Simulation of Sparsely Sampled Thin Features in SPH-Based Free Surface Flows [J]. ACM Transactions on Graphics, 2014, 34(1): 189.
- [105]Nair P, Tomar G. An improved free surface modeling for incompressible SPH [J]. Computers & Fluids, 2014, 102: 304-14.
- [106]Yang S, He X, Wang H, et al. Enriching SPH Simulation by Approximate Capillary Waves; proceedings of the Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, F, 2016 [C].
- [107]Müller M, Macklin M, Chentanez N, et al. Detailed rigid body simulation with extended position based dynamics; proceedings of the Computer Graphics Forum, F, 2020 [C]. Wiley Online Library.
- [108]Kee M H, Um K, Jeong W, et al. Constrained projective dynamics: real-time simulation of deformable objects with energy-momentum conservation [J]. ACM Transactions on Graphics (TOG), 2021, 40(4): 1-12.
- [109]Weiler M, Koschier D, Bender J. Projective fluids [Z]. ACM SIGGRAPH Motion in Games ACM, 2016. 2016: 55-60
- [110]Sin F, Bargteil A W, Hodgins J K. A Point-based Method for Animating Incompressible Flow [J]. Eurographics/ ACM SIGGRAPH Symposium on Computer Animation 2009.
- [111]Goes F D, Wallez C, Huang J. Power particles: an incompressible fluid solver

- based on power diagrams[C] [J]. SIGGRAPH 2015, 2015, 34.
- [112]赵建旺, 陈晨阳, 丰芸芸, 等. 复杂场景流固耦合的高效模拟 [J]. 计算机辅助设计与图形学学报, 2019, 31(8): 10.
- [113]Monaghan J J. On the problem of penetration in particle methods [J]. Journal of Computational Physics,, 1989, 82: 1–15.
- [114]Koschier D, Bender J, Solenthaler B, et al. A Survey on SPH Methods in Computer Graphics; proceedings of the Computer Graphics Forum, F, 2022 [C]. Wiley Online Library.
- [115]Bodin K, Lacoursiere C, Servin M. Constraint fluids [J]. IEEE Trans Vis Comput Graph, 2012, 18(3): 516-26.
- [116]Andrade L F D S, Sandim M, F P. SPH Fluids for Viscous Jet Buckling [J]. Sibgrapi Conference on Graphics IEEE, 2014.
- [117]张雅斓, 班晓娟, 王笑琨, 等. 剪切变稀型 SPH 非牛顿流体的预测-校正方法 [J]. 计算机辅助设计与图形学学报 2017, 29(5): 906-13.
- [118]张雅斓, 班晓娟, 徐衍睿, 等. 面向非牛顿流体仿真的边界处理方法 [J]. 计算机辅助设计与图形学学报, 2019, 31(8): 9.
- [119]Takahashi T, Dobashi Y, Fujishiro I, et al. Implicit Formulation for SPH-based Viscous Fluids [J]. Computer Graphics Forum, 2015, 34(2).
- [120]Cruickshank J O, Munson B R. Viscous fluid buckling of plane and axisymmetric jets [J]. Journal of Fluid Mechanics, 2006, 113(-1).
- [121]Habibi M, Maleki M, Golestanian R, et al. Dynamics of liquid rope coiling [J]. Phys Rev E Stat Nonlin Soft Matter Phys, 2006, 74(6 Pt 2): 066306.
- [122]Weiler M, Koschier D, Brand M, et al. A Physically Consistent Implicit Viscosity Solver for SPH Fluids [J]. Computer Graphics Forum, 2018, 37(2): 145-55.
- [123]Peer A, Teschner M. Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion [J]. IEEE Trans Vis Comput Graph, 2017, 23(12): 2656-62.
- [124]Fries T-P M, Hermann-Georg;. Classification and Overview of meshfree method [J]. Institut für Wissenschaftliches Rechnen, 2004.
- [125]Takahashi T, Batty C. Monolith: a monolithic pressure-viscosity-contact solver for strong two-way rigid-rigid rigid-fluid coupling [J]. ACM Transactions on Graphics, 2020, 39(6): 176.
- [126]Takahashi T, Batty C. FrictionalMonolith: a monolithic optimization-based approach for granular flow with contact-aware rigid-body coupling [J]. ACM Trans Graph, 2021, 40(6): Article 206.
- [127]Goldade R, Aanjaneya M, Batty C. Constraint bubbles and affine regions: reduced fluid models for efficient immersed bubbles and flexible spatial coarsening [J]. ACM Transactions on Graphics, 2020, 39(4).
- [128]Schechter H, Bridson R. Ghost SPH for animating water [J]. ACM Transactions

- on Graphics, 2012, 31(4): 61.
- [129] Si W, Qin J, Chen Z, et al. Thin-Feature-Aware Transport-Velocity Formulation for SPH-Based Liquid Animation [J]. IEEE Transactions on Multimedia, 2018, 20(11): 3033-44.
- [130] Si W, Liao X, Qian Y, et al. Versatile numerical fractures removal for SPH-based free surface liquids [J]. Computers & Graphics, 2019, 81: 26.
- [131] 袁志勇, 徐标, 廖祥云. 基于 SPH 方法的快速逼真流体表面张力仿真 [J]. 计算机学报, 2019, 42(9): 14.
- [132] Harada T, Koshizuka S, Kawaguchi Y. Smoothed Particle Hydrodynamics in Complex Shapes [Z]. SCGG '07: Proceedings of the 23rd Spring Conference on Computer Graphics. Budmerice Castle, Slovakia. 2007: 191-7.10.1145/2614348.2614375
- [133] Ihmsen M, Akinci N, Gissler M, et al. Boundary handling and adaptive time-stepping for PCISPH [Z]//BENDER J, ERLEBEN K, TESCHNER M. Proceedings of the Seventh Workshop on Virtual Reality Interactions and Physical Simulations. Copenhagen, Denmark; The Eurographics Association 2010.10.2312/PE/vrphys/vrphys10/079-088
- [134] Akinci N, Ihmsen M, Akinci G, et al. Versatile rigid-fluid coupling for incompressible SPH [J]. ACM Transactions on Graphics, 2012, 31(4): 62.
- [135] Band S, Gissler C, Peer A, et al. MLS pressure extrapolation for the boundary handling in divergence-free SPH [Z]. Proceedings of the 14th Workshop on Virtual Reality Interactions and Physical Simulations. Delft, The Netherlands; Eurographics Association. 2018: 55–63
- [136] Fujisawa M, Miura K T. An Efficient Boundary Handling with a Modified Density Calculation for SPH [J]. Computer Graphics Forum, 2015, 34(7): 155-62.
- [137] Chang Y, Liu S, He X, et al. Semi - analytical Solid Boundary Conditions for Free Surface Flows [J]. Computer Graphics Forum, 2020, 39(7): 131-41.
- [138] Cornelis J, Bender J, Gissler C, et al. An optimized source term formulation for incompressible SPH [J]. The Visual Computer, 2019, 35(4): 579-90.
- [139] Reinhardt S, Krake T, Eberhardt B, et al. Consistent shepard interpolation for SPH-based fluid animation [J]. ACM Transactions on Graphics, 2019, 38(6): 189.
- [140] Chen X-S, Li C-F, Cao G-C, et al. A moving least square reproducing kernel particle method for unified multiphase continuum simulation [J]. ACM Transactions on Graphics, 2020, 39(6): 176.
- [141] Zhu B, Lee M, Quigley E, et al. Codimensional non-Newtonian fluids [J]. ACM Transactions on Graphics, 2015, 34(4): 1-9.
- [142] Monaghan. Smoothed particle hydrodynamics [J]. Reports on Progress in Physics, 2005.
- [143] Peer A, Ihmsen M, Cornelis J, et al. An Implicit Viscosity Formulation for SPH Fluids [J]. ACM Transaction on Graphics, 2015, 34(4): 114:1-:10.

- [144] Patankar S. Numerical heat transfer and fluid flow. [J]. CRC press, 2018.
- [145] Shao S, Lo E Y. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface [J]. Advances in water resources, 2003, 26(7): 787-800.
- [146] Khayyer A, Gotoh H. Enhancement of stability and accuracy of the moving particle semi-implicit method [J]. Journal of Computational Physics, 2011, 230(8): 3093-118.
- [147] Bhattacharya H, Gao Y, Bargteil A W. A Level-Set Method for Skinning Animated Particle Data [J]. IEEE Trans Vis Comput Graph, 2015, 21(3): 315-27.
- [148] Cruickshank J, Munson B R. Viscous fluid buckling of plane and axisymmetric jets [J]. Journal of Fluid Mechanics, 1981, 113: 221-39.
- [149] Cross M M. Rheology of non-newtonian fluids: A new flow equation for pseudoplastic systems [J]. Journal of Colloid Science, 1965, 20(5): 417-37.
- [150] Belytschko T, Xiao S. Stability analysis of particle methods with corrected derivatives [J]. Computers & Mathematics with Applications, 2002, 43(3–5): 329-50.
- [151] Swegle J W, Hicks D L, Attaway S. Smoothed particle hydrodynamics stability analysis [J]. Journal of computational physics, 1995, 116(1): 123-34.
- [152] Ihmsen M, Cornelis J, Solenthaler B, et al. Implicit incompressible SPH [J]. IEEE Trans Vis Comput Graph, 2014, 20(3): 426-35.
- [153] Lyu H-G, Sun P-N, Huang X-T, et al. On removing the numerical instability induced by negative pressures in SPH simulations of typical fluid–structure interaction problems in ocean engineering [J]. Applied Ocean Research, 2021, 117: 102938.
- [154] Sun P, Colagrossi A, Marrone S, et al. Multi-resolution Delta-plus-SPH with tensile instability control: Towards high Reynolds number flows [J]. Computer Physics Communications, 2018, 224: 63-80.
- [155] Sun P, Colagrossi A, Marrone S, et al. The δ plus-SPH model: Simple procedures for a further improvement of the SPH scheme [J]. Computer Methods in Applied Mechanics and Engineering, 2017, 315: 25-49.
- [156] Xu R, Stansby P, Laurence D. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach [J]. Journal of computational Physics, 2009, 228(18): 6703-25.
- [157] Randles P, Libersky L. Normalized SPH with stress points [J]. International Journal for Numerical Methods in Engineering, 2000, 48(10): 1445-62.
- [158] Chen D, Huang W, Sloan S W. An alternative updated Lagrangian formulation for finite particle method [J]. Computer Methods in Applied Mechanics and Engineering, 2019, 343: 490-505.
- [159] O'Connor J, Rogers B D. A fluid–structure interaction model for free-surface flows and flexible structures using smoothed particle hydrodynamics on a GPU

- [J]. Journal of Fluids and Structures, 2021, 104: 103312.
- [160]Zhan L, Peng C, Zhang B, et al. A stabilized TL-WC SPH approach with GPU acceleration for three-dimensional fluid-structure interaction [J]. Journal of Fluids and Structures, 2019, 86: 329-53.
- [161]Sun P-N, Le Touze D, Oger G, et al. An accurate FSI-SPH modeling of challenging fluid-structure interaction problems in two and three dimensions [J]. Ocean Engineering, 2021, 221: 108552.
- [162]Akinci N, Akinci G, Teschner M. Versatile surface tension and adhesion for SPH fluids [J]. ACM Transactions on Graphics, 2013, 32(6): 182.
- [163]Yang, Li, XS, et al. A novel surface tension formulation for SPH fluid simulation [J]. VISUAL COMPUT, 2017.
- [164]Chen J, Kala V, Marquez-Razon A, et al. A momentum-conserving implicit material point method for surface tension with contact angles and spatial gradients [J]. ACM Transactions on Graphics, 2021, 40(4): 1-16.
- [165]Liu M, Liu G. Smoothed particle hydrodynamics (SPH): an overview and recent developments [J]. Archives of computational methods in engineering, 2010, 17(1): 25-76.
- [166]Dyka C, Ingel R. An approach for tension instability in smoothed particle hydrodynamics (SPH) [J]. Computers & structures, 1995, 57(4): 573-80.
- [167]Chen J, Beraun J. A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems [J]. Computer Methods in Applied Mechanics and Engineering, 2000, 190(1-2): 225-39.
- [168]Fries T P, Belytschko T. Convergence and stabilization of stress - point integration in mesh - free and particle methods [J]. International Journal for Numerical Methods in Engineering, 2008, 74(7): 1067-87.
- [169]Chalk C, Pastor M, Peakall J, et al. Stress-Particle Smoothed Particle Hydrodynamics: An application to the failure and post-failure behaviour of slopes [J]. Computer Methods in Applied Mechanics and Engineering, 2020, 366: 113034.
- [170]Tsuruta N, Khayyer A, Gotoh H. A short note on dynamic stabilization of moving particle semi-implicit method [J]. Computers & Fluids, 2013, 82: 158-64.
- [171]Batty C, Bertails F, Bridson R. A fast variational framework for accurate solid-fluid coupling [J]. ACM Transactions on Graphics, 2007, 26(99).
- [172]He X, Wang H, Wu E. Projective Peridynamics for Modeling Versatile Elastoplastic Materials [J]. IEEE Trans Vis Comput Graph, 2018, 24(9): 2589-99.
- [173]Bridson R. Fast Poisson disk sampling in arbitrary dimensions [J]. SIGGRAPH sketches, 2007, 10(1): 1.
- [174]Tampubolon A P, Gast T, Klár G, et al. Multi-species simulation of porous sand and water mixtures [J]. ACM Transactions on Graphics, 2017, 36(4): 1-11.
- [175]Ando R, Batty C. A practical octree liquid simulator with adaptive surface

- resolution [J]. ACM Transactions on Graphics, 2020, 39(4).
- [176] Ando R, Tsuruno R. A particle-based method for preserving fluid sheets; proceedings of the Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation, F, 2011 [C].
- [177] Um K, Baek S, Han J. Advanced Hybrid Particle-Grid Method with Sub-Grid Particle Correction [J]. Computer Graphics Forum, 2014, 33(7): 209-18.
- [178] Lind S J, Xu R, Stansby P K, et al. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves [J]. Journal of Computational Physics, 2012, 231(4): 1499-523.
- [179] Yang L J, Rakhsha M, Hu W, et al. A consistent multiphase flow model with a generalized particle shifting scheme resolved via incompressible SPH [J]. Journal of Computational Physics, 2022, 458.
- [180] Zhu Y B R. Animating sand as a fluid [J]. ACM Transactions on Graphics, 2005, 24(3).
- [181] Ando R, Thurey N, Tsuruno R. Preserving fluid sheets with adaptively sampled anisotropic particles [J]. IEEE Trans Vis Comput Graph, 2012, 18(8): 1202-14.
- [182] Bouaziz S, Martin S, Liu T, et al. Projective dynamics [J]. ACM Transactions on Graphics, 2014, 33(4): 1-11.

附录一 第三章部分测试场景的单帧平均计算时间统计

本文第三章中部分测试场景的技术统计如下表。其中 s 代表时间单位秒，ms 代表时间单位毫秒，k 代表数量单位千。

场景	粒子	粘性($pa \cdot s$)	表面张力	时间步长	SIMPLE 迭代	粘性计算	压强计算	总时间
图 3-9(a)	22k	5000	0.0005	1.0ms	1	4.1ms	5.2ms	0.036s
图 3-9(b)	22k	5000	0.0005	1.0ms	2	4.1ms	5.2ms	0.045s
图 3-9(c)	22k	5000	0.0005	1.0ms	4	4.1ms	5.2ms	0.064s
图 3-9(d)	22k	5000	0.0005	1.0ms	8	4.1ms	5.2ms	0.11s
图 3-9(e)	22k	5000	0.0005	1.0ms	16	4.1ms	5.2ms	0.19s
图 3-13(d)	36k	5000	0	1.0ms	1	5.3ms	8.2ms	0.061s
图 3-13(e)	36k	5000	0	1.0ms	5	5.3ms	8.2ms	0.11s
图 3-13(f)	36k	5000	0	1.0ms	10	5.3ms	8.2ms	0.18s
图 3-14(e)	5k	5000	0	1.0ms	1	1.0ms	1.7ms	0.011s
图 3-14(f)	5k	5000	0	1.0ms	5	1.0ms	1.7ms	0.020s
图 3-15(a)	36k	0	0	1.0ms	1	5.5ms	0	0.032
图 3-15(b)	36k	300	0	1.0ms	2	5.5ms	4.4ms	0.051
图 3-15(c)	36k	5×10^7	0	1.0ms	2	5.5ms	13.1ms	0.070
图 3-16	136k	3, 100, 5×10^5	0	1.0ms	2	31ms	49ms	0.27s
图 3-17	975k	5×10^4	0	1.2-1.9ms	4	82ms	171ms	0.38s
图 3-19	48k	动态	0	2.0-3.0ms	5	6.3ms	9.1ms	0.15
图 3-20	103k	动态	0	1.0ms	2	17ms	41ms	0.38
图 3-21 左	104k	5×10^6	0	1.0ms	1	16ms	37ms	0.13
图 3-21 右	104k	5×10^6	0	1.0ms	5	16ms	37ms	0.34
图 3-23	462k	400	0.002	1.4-2.2ms	2	65ms	135ms	1.6s
图 3-24	1068 k	200 300 5×10^4 5×10^5	0, 0.0005	1.0ms	5	90ms	203ms	2.4s

附录二 第四章部分测试场景的单帧平均计算时间统计

本文第四章中，部分测试场景的技术统计如下表。其中 s 代表时间单位秒，k 代表数量单位千，M 代表数量单位百万。

场景	方法	实粒子数量	虚粒子数量	压强计算时间	虚粒子生成时间	总计算时间
图 4-15	策略 C	237k	312k-401k	0.18s	0.024s	0.32s
	策略 B	237k	237k	0.16s	0.032s	0.29s
	策略 A	237k	237k	0.16s	-	0.26s
图 4-22	策略 C	72k	97k-600k	0.067s	0.005s	0.12s
	策略 B	72k	72k	0.051s	0.007s	0.084s
	策略 A	72k	72k	0.051s	-	0.086s
图 4-25	策略 C	144k	0.2M-1.0M	0.120s	0.010s	0.20s
	策略 B	144k	144k	0.094s	0.018s	0.18s
	策略 A	144k	144k	0.095s	-	0.16s
图 4-26	策略 C	1.0M	1.8M-6.2M	1.1s	0.061s	1.6s
图 4-27	策略 C	520.2k	0.6M-1.5M	0.43s	0.038s	0.87s
图 4-30	策略 C	505k	0.57M-1.7M	0.81s	0.060s	1.9s
图 4-31	策略 C	1.5M	1.8M-4.1M	1.4s	0.082s	2.4s
图 4-32	策略 C	1.5M	1.8M-6.2M	1.5s	0.096s	2.6s

致 谢

时光如梭，飞逝而去，回念往昔，历历在目，思绪万千，如江河、如残云、如砂砾、如树木、如金石，难以尽叙。此生修行路上，无数人赐予我恩义、赠予我炽烈深情、送予我期待与关心，纵然穷尽我终生，亦无以为报。此生求学之行已近止结，此刻寸阴若岁，欲将心中难以名状之情，用苍白文字描述一二，聊表谢意。

（一）梦想。

自幼我就十分羡慕科研工作者，羡慕他们能够在人类认知的极限之处求索，羡慕他们是推动文明进步的中坚力量。离开校园工作多年之后，这份羡慕之情也愈加强烈。在决定回到学校读博之后，我遇到了改变我生命轨迹的导师——王文成老师。王老师引领我来到了计算机图形学的桃源深处，让我与这构建梦想世界的技术结缘，从此我踏上了“圆梦”之旅。王老师渊博的知识、严谨的治学风格、一丝不苟的工作态度，引领我逐渐深入我的研究方向之中；而在求学之外，王老师随和、温暖而又坚定的态度，总是能够给予我感动和继续前行的力量。感谢王老师无私给予我的一切，能够成为王老师的学生是我此生的荣幸，在此我想向王老师表达我的感谢，祝愿王老师身体健康，事事顺心如意！

天赋异禀、才华横溢、为人随和、内心坚毅，是我梦想中的完美人设。在我博士求索之路上，我遇到了这样一位小说主角般传奇的老师——何小伟老师。何老师教会了我无数的知识和技能，为我指点了科研的方向，也在生活上给我无微不至的关心。他如奔流般的思维，总是能够在研究中另辟蹊径；他宽广无垠的知识和技术储备，总是能解决一切难题；他高深莫测的数学功底，总是让我叹为观止；而他随和、谦虚、温柔而又坚定的性格和处事态度，让我无比的钦佩；与何老师相处的日子里如沐春风般的美好。

“夫子温、良、恭、俭、让以得之。夫子之求之也，其诸异乎人之求之与！”——《论语·学而》。提到温、良、恭、俭、让，就会让我想到另一位大儒——吴恩华老师。感谢儒雅、和蔼的吴恩华老师给予我的帮助和鼓励。

瞻彼淇奥，绿竹如簧，有斐君子，如金如锡，如圭如璧。感谢我的研究生期间的导师——飒爽英姿、风流倜傥、才华横溢、满腹经纶的秦伟平老师。他在我研究生期间教会了我很多，从专业知识和技能，到创新思维、工作学习方法、做汇报的策略和技巧。他让我懂得了什么是侠之大者，为国为民；让我了解了海纳百川，有容乃大；让我见识了什么是知微见著、臻于至善。

此外，也要感谢侯飞老师、刘学慧老师、赵辉老师在科研上对我的点拨！感谢实验室的张丽老师、费腾老师、孙守云老师、张鑫老师，以及研究生部的李晓

云老师、杜慧文老师、张欢老师对我的支持和帮助！感谢在物理仿真方向求索的同道们，任立新、何浩、郭煜中、卢子璇、缪冉、罗旭坤、叶子萌、常悦、夏提、康杨宇轩，大家在科研上相互帮助、不断地砥砺前行，结下了如战友般的情谊！感谢图形和视觉组的同学们，李雷、王少东、张哲、储翊尧、马俊辉、邱枫、谭雪晗、赵冲、马帅、封士明、王驰誉、伊力夏提、刘浩、陈旭辉、申泽宇、王雨浓、李赞、杜列邦，感谢他们对我的无私帮助！也要感谢我的室友——王盛春，感谢他这几年来在生活方面对我的包容与帮助。

感谢我的研究方向——物理仿真，这是一门可以创造“世界”的科学技术，尽管它学习起来不容易，但这个方向中有星辰，有大海，还有我的梦想。

“起初神创造天地，地是空虚混沌。深的灵运行在水面上，神说要有光，就有了光……”——《圣经·创世纪·旧约》。

最后，再次感谢助我圆梦的老师、同学，恩情难忘！

（二）生命。

我出生于一个幸福的大家庭。可能是由于早产，我出生时浑身冰凉，那时的医疗条件有限，我姥姥只能把我贴身包裹在她的衣服里，用她的体温点燃了我的生命。年幼时我体弱多病，几次徘徊于鬼门关外，如果没有我的父母、二姨、三姨、四姨、老姨对我细致入微的照顾，也许我不会活过成年。我明白，是温暖、厚重的亲情，给了我生命。

我的父亲是个善良、记恩不记仇、刀子嘴豆腐心的人。他的话很多，却唯独不会说想念我的话。但我知道，我是他心里最牵挂、最骄傲的那一部分。我的母亲是个善良、要强、乐观、细心的人。她在我的身上投入了她几乎全部的精力，为了照顾体弱多病、性格孤僻的我，她辞去工作，开了幼儿园，还落下了一身的病。然而，她从未给过我心理上的压力，是这份深沉的感情与信任，让我这个内心敏感的孩子能够开心、幸福的成长。

虽然我是独生子，但我有三个比亲妹妹还亲的表妹。她们成熟、细心，总是在牵挂着我，当我在她们面前时，我更像是大她们十几岁的巨型弟弟；

我的爷爷、大爷、大娘、二姑和各位哥哥姐姐，也是我心灵上的另一片港湾，他们也像我父亲一样，不喜欢直接表达感情，但深沉的挂念总是让我安心。

还有我的姨夫们，待我也如亲外甥一般，总是为我排忧，为我解难。

我姥爷是一个倔强而有才华的老头。他手很巧，会做家具、衣服、二胡，甚至缝纫机和自动化的崩爆米花机。在那个青黄不接、有上顿没下顿的年代，他愿意花很多钱，为我的几个姨买昂贵的小提琴、手风琴、脚踏风琴。东北的国企改革后他下岗了，从此生活变得更加困顿。年事已高，依然要支撑起一个小小的自行车修理亭。他的手上沾满了油渍，但我知道，这双脏兮兮的手会画漂亮的油画和水彩画，弹得一手好二胡和笛子；他的嗓音有时候累得沙哑，但我知道，他的

嗓子比童祥苓还要高，还要亮。在我姥爷的世界里，无论多困苦，他的诗和大海从未远去，直到前年，他与世长辞。我时常想起他给我扎的风筝、给我焊的爬犁、还有给我做的世上最好吃的鱼，时常想起他教我画画、指点我弹琴的童年时光，就像一切从未远去。

我的爷爷是个热爱读书的人，即使他的眼睛昏花，记性变得很差，甚至上午的事儿下午就会忘掉，但他依然带着眼镜，一本接一本的阅读着艰深厚重的历史书。尽管我与爷爷在一起的时间很少，但我知道他很挂念我。那一年冬天他离开尘世的那一刻，无论如何都不肯闭上眼睛，直到看到我从外面慌张的赶回来，跑进屋子……

大恩不言谢，他们给我生命，伴我成长。这一份份恩情的厚重我无以为报。我唯有背负着他们的挂念，努力前行。

记得我在学校的某一天，我无意间翻开了我母亲的博客，上面有一首她写的诗：

今天是你返校的日子，这几日纠结就没离开我。

欣慰你在学习的路上，能继续求索。

但从此注定聚少离多。

心中有诸多不舍，牵挂从没有断过。

不敢去车站送你，因为怕自己泪雨滂沱。

自认为很坚强，眼泪总远离我。

今天才知道，原来我也很软弱。

心里有很多话想和你说：

身在异地要爱护自我；

三餐一顿不能错过；

按时休息就不会黑眼窝；

不熬夜头发就不会掉得那么多；

每天记得垫鞋垫，还有经常吃水果；

早起一定要喝杯水；要注意冷暖；别为美把棉服早脱；

你一定会笑老妈如此啰嗦；

会说：我长大了，我会生活。

在我眼里你永远是孩子，你是我永远的牵挂休想逃脱。

还有话对你说：上大学、读硕、读博，我只在梦里才有过。

希望你珍惜好好把握。

我知道，你不会辜负我，你会为自己一生负责。

希望你能不断的振奋自我；

希望在未来的路上不畏坎坷；

希望将来你有美好、幸福的生活。

感谢妈妈，我独自一人身在外地的这些年，我有照顾好自己。我想，我也实现了您曾经的梦想……

（三）成长。

小的时候我不仅学习差，性格也很自卑、孤僻，很幸运的我遇到了王哲老师、王敏老师、田禾老师、程佳老师，她们对我的认可，让我拾起了我的信心，让我更坚定、更自信地走在我的人生路上！

相知无远近，万里尚为邻；数人世相逢，百年欢笑，能得几回又。感谢陪伴我成长、与我分享喜怒哀乐的同学、朋友和曾经的同事；感谢曾帮助我、关心我、指点我的前辈和长辈们。

（四）陪伴。

本科毕业时，我参加一个乐队时遇到一位个子很高很帅的女学霸贝斯手。因缘巧合，我们俩开始了长久的异地恋：她在冰城我在喜都，当她奔赴喜都后我又飞去了胶澳，随后我俩又各自奔向了燕京与鹏城。直到 2017 年的那个国庆节，我们组成了分布式的家庭，从此她是主机，我是从机。深厚的感情和绝对的信任是我俩之间的通信总线与通信协议。间隔 2372 公里我们也从未退烧，因为我们的灵魂已经发生了量子纠缠，她自旋向上我自旋向下，我俩很可能是人类史上最早实用化的量子传输系统。

感谢她在玛雅历第五太阳纪末日的前一年，选择了与我共渡漫漫人生路；感谢她在我身无长物之时，就看透了我无任何修饰、吸收峰为 520nm 的灵魂；感谢她在我处于低能态时，向我打出了一个个短波长光子；感谢她像希格斯场一样，赋予我生活的意义；感谢她在这九百三十亿光年可观测宇宙中拉尼亚凯亚——一本超星系团——银河系——猎户座旋臂的一个不起眼的时空上，谱写了只有我才能听懂得的歌曲……听我说，谢谢你，如果没有你，我就是一个没有主机的从机。

如若问起，我的妻子为何如欧拉方程般完美？那是由于她生长在一个温暖、真诚、正直、朴实的家庭里。在这里，我感谢我的岳父岳母，感谢您们对我的信任和认可；感谢您们对我的包容和理解；感谢您们把我视为亲儿子般的关心照顾。

（五）尾声。

我的博士求学之路上，陪伴我的还有代码、数值算法、流体仿真技术和 SPH 法。这几年的修习，不仅让我了解了很多的专业知识，也让我懂得了，上善若水：

- 处众人之所恶，故几于道。居善地，心善渊，与善仁，言善信，正善治，

事善能，动善时。

——老子

● Be formless, shapeless, like water.

——李小龙

● $\begin{cases} \rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla \cdot \nabla \mathbf{u} + \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$

——纳维斯托克斯

感谢曾滋润我的点点滴滴，愿我也能如水一般，润泽万物而不争。

最后，再次向为我毕业论文评审付出努力的老师与专家表示由衷的感谢！

2022 年 10 月

作者简历及攻读学位期间发表的学术论文与其他相关学术成果

作者简历:

2007 年 9 月——2011 年 7 月，在黑龙江大学电子工程学院获得学士学位。

2011 年 9 月——2014 年 7 月，在吉林大学电子科学与工程学院获得硕士学位。

2018 年 9 月——2022 年 12 月，在中国科学院软件研究所攻读博士学位。

工作经历:

2014 年 8 月——2018 年 8 月，在青岛海信集团青岛海信网络科技股份有限公司任研发工程师。

已发表（或正式接受）的学术论文:

1. Shusen Liu, Xiaowei He, Wencheng Wang, Enhua Wu. Adapted SIMPLE Algorithm for Incompressible SPH Fluids With a Broad Range Viscosity[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 28(9): 3168-3179. (CCF-A)
2. Yue Chang, Shusen Liu, Xiaowei He, Sheng Li, Guoping Wang. Semi-analytical Solid Boundary Conditions for Free Surface Flows[J] Computer Graphics Forum. 2020, 39(7): 131-141. (CCF-B)
3. 刘树森, 何小伟, 王文成, 吴恩华. 光滑粒子流体动力学 (SPH) 流体仿真技术综述. 软件学报, 2022. (已接收)

申请或已获得的专利:

1. 何小伟, 王文成, 刘树森, “一种基于 SPH 的粘性不可压缩流体仿真方法”
专利号: CN201911390333.9;
2. 刘树森, 何小伟, 王文成, “一种拉伸稳定的 SPH 流体仿真方法”, 专利号:
CN202210976684.3;
3. 常悦, 刘树森, 何小伟, 李胜, 汪国平, 一种面向自由表面流的固体边界提
取方法, 专利号: CN202011119179.4。

参加的研究项目及获奖情况:

1. 参加国家重点研发项目: 复杂时变场景的物理仿真关键技术, 项目号: No. 2021YFB1715800;
2. 中国计算机学会计算机辅助设计与图形学专业委员会 (CCF CAD&CG): 优
秀图形开源软件, 项目名: 泛动引擎 Peridyno;
3. 2021 届中科院软件研究所学术年会优秀海报奖;
4. 2020 年中国科学院大学三好学生;

5. 2014 年获得共青团长春市委举办“长春市青年科技创新创业大赛”二等奖。