

分类号: TM888

单位代码: 10335

密 级:

学 号: 21935006

浙江大学

硕士学位论文



中文论文题目: 基于物质点法的流体表面张力模拟

英文论文题目: Simulation of fluid surface tension
based on material point method

申请人姓名: 袁淳

指导教师: 杨勋年

专业名称: 应用数学

研究方向: 物理模拟

所在学院: 数学科学学院

论文提交日期: 二〇一九年四月

基于物质点法的流体表面张力模拟



论文作者签名: _____

指导教师签名: _____

论文评阅人 1:	陈某某	教授	XX 大学
评阅人 2:	张某某	教授	XX 大学
评阅人 3:	李某某	教授	XX 大学
评阅人 4:	吴某某	教授	XX 大学
评阅人 5:	杨某某	教授	XX 大学

答辩委员会主席:	郑某某	教授	浙江大学
委员 1:	郑某某	教授	浙江大学
委员 2:	刘某某	教授	浙江大学
委员 3:	程某某	教授	浙江大学
委员 4:	杨某某	教授	浙江大学
委员 5:	吴某某	教授	浙江大学

答辩日期: _____ 二〇一九年六月

Simulation of fluid surface tension
based on material point method



Author's signature: _____

Supervisor's signature: _____

External Reviewers: _____
Moumou Chen Prof. XX University
Moumou Zhang Prof. XX University
Moumou Li Prof. XX University
Moumou Wu Prof. XX University
Moumou Yang Prof. XX University

Examining Committee Chairperson:
Moumou Zheng Prof. Zhejiang University

Examining Committee Members:
Moumou Zheng Prof. Zhejiang University
Moumou Liu Prof. Zhejiang University
Moumou Chen Prof. Zhejiang University
Moumou Yang Prof. Zhejiang University
Moumou Wu Prof. Zhejiang University

Date of oral defence: _____ June 2019

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

签字日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：

导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

致 谢

感谢我的爹，感谢我的妈，感谢那个亲爱的他。

摘 要

近年来,物质点法在物理仿真领域取得了巨大的进步,从电影行业,地球物理,到材料科学都带来了巨大的革新,展现出其强大的生命力。得益于其算法框架的包容性,多种物理现象能够同时在一个系统里求解,带来一些过去从未成功模拟过的结果。然而,物质点法相较于其他算法,在流体解算上面带来许多不真实的现象,尤其是在流体粘性和表面张力的部分。由于物质点法对于物质的表示形式是粒子,难以定义表面,这给表面张力的模拟带来了巨大的困难,同时因为物质点法本身对时间步长要求严格,为了得到和表面张力耦合稳定的结果,经常在方程中加入阻尼项以及使用隐式格式,而这更加剧了物质点法本身数值粘性,往往使最后的模拟结果不尽如人意。

本文研究了一种在物质点法框架下计算流体表面张力的方法,其主要思想是利用点云重构流体表面用于计算表面张力,同时改进了表面粒子和内部粒子插值格式以及表面张力项的离散格式,提高了辛积分的稳定性,其相比于纯隐式格式降低了数值耗散,缓解了流体粘性問題。本文的主要工作如下:

(一)结合物质点法的计算流水线,提出了新的基于几何迭代法的流体表面重构方法。利用物质点法本身的数据结构,在不增大内存负担的情况下,做到高速并行。同时给出相应的法向估计算法,相比于直接通过梯度或者网格估计法向,计算效率更高。

(二)基于工作(一)改进了表面张力数值计算格式。通过采样重建得到的表面网格,获取表面积分粒子用于计算表面张力,并通过新的数值离散方式,给出的表面张力项易于实现。同时给出嵌入内部粒子的方式,让表面张力能够更快的传导到内部粒子,提高了辛积分格式的稳定性。

关键词: 物质点法; 流体模拟; 表面张力; 表面重构

Abstract

In recent years, material point method(MPM) have made tremendous progress in physical simulation. From film industry, geophysics, to material science, MPM have been bringing huge innovations for these industries, showing its power among simulation techniques. Thanks to the universality of the MPM, various physical phenomena can be coupled in a system, realizing some that have never been successfully simulated in previous techniques. However, in fluid simulation, more artifacts appeared when we compare to other fluid simulation method, especially some unwanted fluid viscous and unreal surface tension effects. Since the continuum represented by some particles in MPM, it is difficult to define a feasible surface of particle clouds, which brings great difficulties to the simulation of surface tension. At the same time, because the material point method itself has strict requirements on the timestep, in order to obtain a stable coupling with surface tension, damping terms are often added to the equations and implicit schemes used, which intensifies the numerical viscosity of MPM fluid and makes simulation results unsatisfactory.

This paper studies a method of simulating the surface tension of fluids under the framework of the material point method. The main idea is to use point clouds to reconstruct the surface of the particle represented fluid. At the same time, we propose a surface-inner particle connecting method and a discrete method for surface tension calculation, improving the stability of symplectic integral. Of course, symplectic scheme have less numerical dissipation than implicit scheme, this feature modify the viscosity of MPM fluid. The main work of this paper is as follows:

1. We propose a novel surface reconstruction method based on progressive-iterative approximation. In our novel method, we can utilize the data structure of the material point method to achieve high speed parallel computation without increasing the burden of memory. We also propose a normal estimate method, which is more efficient than directly calculate gradient and smoother than calculated by mesh.

2. Based on our previous work, we improve the numerical scheme of surface tension. We sample the surface mesh reconstructed from point clouds to obtain surface integral particles. With this surface integral particles, we design a new and easily implemented

discrete method for the surface tension calculation. At the same time, we propose a inserted particle method for surface tension conduction to inner particle. This method improve the stability for symplectic scheme.

Keywords: Material point method; Fluid simulation; surface tension; surface reconstruction

图目录

表 目 录

目 录

1 绪论

1.1 问题背景

计算机模拟几乎在所有的工程领域的一个重要工具,从土木工程、地球物工程、生物医学工程这些大型工程领域,到服装设计、家具设计这些生活周边,再到游戏行业,电影特效这些文化领域处处都有计算机模拟的身影。计算机模拟不单单只是解决一些极其复杂难以实验,难以分析的问题,它也在替代一些昂贵耗时的物理实验,给设计人员一些基础的直观信息。

目前主要的模拟方法以及一些商业化的工业模拟都是在有限元^[2]的框架下实现的。尽管有限元方法已经理论成熟,并且早已经在学术界和工业界广泛的使用,但是当用来求解一些大形变问题,拓扑变化问题时有限元便显得捉襟见肘,虽然这些问题可以使用重新网格化或者自适应网格来解决,但是网格处理本身就是一个复杂的问题,最后由此带来的便是算法的鲁棒性问题,以及算法的性能也会大打折扣。

在电影特效行业以及游戏行业,经常遇到的问题便是大形变以及多物理场耦合问题,这也导致传统的有限元方法难以应对。然而,电影特效与游戏行业本身对模拟精度的要求并不是那么高,对于这些对精度没有太高要求的问题,人们转而不再单纯的使用网格来表示物体,更灵活的粒子表示方法进入了学者们的视野。近年来,在电影特效界,首当其冲粒子表示方法的便是物质点法,该方法很大程度解决了有限元大形变以及拓扑变化难的缺点,同时由于其加入了空间背景网格参与求解,使其对物理场建模有着极大的灵活性,在应对复杂多样的物理现象时,常常可以使用物质点法作为切入点。但是,在流体模拟中,物质点法相对于专门为流体计算设计的方法并不完善,效果也相差甚远,如流体粘性,表面张力等效应都没有得到很好的表现,而这也使得多物理场模拟在涉及流体的时候,往往并不能很全面的表现出流体的特性,或给人以不真实感。

1.2 相关工作

1.2.1 物质点法

Walt Disney 工作室在 2013 年成功对雪的模拟^[2]将物质点法带到了图形学领域,同时在《冰雪奇缘》这部电影中大放异彩。此时的物质点法在图形学领域还仅仅是用于模拟沙泥这类粘状的物质,但是其方便多物理场耦合的特性已经吸引了众多学者的注意,从 Ram 模拟粘性的牙膏、海绵、泡沫^[2]这类符合物质点法特性的材料,到 Chengfanfu 等人提出了 APIC,在物质点法中加入仿射项^[2]改进物质点法,

使其能够模拟粘性较低的流体，再到 Gergely 等人改进塑性的屈服模型^[2]，沙粒模拟也加入到物质点法能够模拟的范畴。之后 Wang 将有限元与物质点法相结合，让弹性体^[2]能够方便地实现破碎效果，同时改进物质点法本身的数值粘性带来的碰撞不真实感。在有了这么多的针对单种物质模拟的基础工作之后，人们开始着手于多物理场，如在 2017 年 Tampubolu 等人的^[2]对水和沙子耦合的模拟，Chengfanfu 等人对布料沙子以及头发织物^[2]的模拟，这些复杂的场景带来了惊人的视觉效果。在 2019 年，物质点法开始向着“生活”迈进，Mengyuan 等人^[2]首先将热转换、弹塑性、气体压强、流体压强、流体蒸发这些现象耦合在一起，成功的在计算机中模拟了烤曲奇饼干、烤面包、夹心面包、水果风干等效果，展现出物质点法框架巨大的包容性。除了发掘物质点法优势之外，也有学者在改进物质点法本身的劣势，比如物质点法模拟流体时，经常表现出数值粘性，给人一种胶水质感。ChuYuan 等^[2]人针对流体所展现的粘性，继续改进物质点法的粒子网格对流过程，通过推广 APIC 算法，使物质点法下的流体展现出更多细节，同时更好的保留速度信息，大大的降低了流体本身的粘性。Yu 等人^[2]针对物质点法流体和固体耦合时，由固体表面计算时带来的数值粘性做出了改进，成功的解决了流体流过固体表面出现粘连的问题。

1.2.2 流体表面张力模拟

流体表面张力存在于两种混溶的流体或在固体和气体的相交面上，其大小正比于接触面的曲率。通常我们在模拟大规模的流体时，表面张力相比于其他力来说对流体运动影响极小，因此人们经常不考虑表面张力，此时人们经常称这种流体为无表面流体。然而在小尺度流体模拟时，表面积体积比足够大，那么表面张力带来的影响将不可忽视。

尽管物质点法在弹性体，多物理场，无表面流体模拟都取得了很大进展，但是其针对流体表面张力并没有很多工作，最早尝试往物质点法添加表面张力是 2012 年由 L.Chen 等人^[2]提出的的广义插值物质点法 (GIMP)，他们使用连续体表面 (CSF) 对表面张力建模，通过空间的质量梯度来计算表面张力。然而，物质点法的空间质量分布非常粗糙，在他们的实际实验，存在较大的不真实感。之后便是 2020 年的 David 等人^[2]，他们认为直接使用辛格式积分需要很小的时间步长才能保证系统的稳定，于是使用隐式法来耦合表面张力与流体内部压强。然而物质点法每一帧的控制网格都有所差别，于是隐式法每一帧都有重建一个较大的稀疏矩阵的要求，且难以预判该稀疏矩阵的格式，这也给性能提升带来了巨大的挑战。

当然，在非物质点法上，有许多针对流体表面张力设计的算法。在图形学领域，Akinci 等人^[2]改进 SPH 算法，在模拟稳定态的表面张力取得了很好的效果。Wang 等人^[2]直接计算流体与固体的接触角，在模拟最后的接触结果时，能够达到逼真的

效果。在 2020 年, Hui 等人^[2]通过检测表面粒子, 然后在每一个表面粒子局部的重建二次曲面, 用于计算每个表面粒子上的表面张力, 这种方法很好的捕捉了表面张力的各种效果, 然而其计算代价也是高昂的, 同时 Hui 在文章中也提到, 文章中使用的探测一个粒子是否是表面粒子的算法并不鲁棒, 可能会引发一些意想不到的问题, 同时对于检测粒子是否是表面粒子本身就是一件困难的事情, Marcos 等人^[2]在 2016 年对探测表面粒子这件事情有着一些讨论。除了使用粒子法, 也有学者直接使用网格法来模拟流体, 在 2016 年, Fang 等人^[2]提出使用边界元的方法来模拟流体, 这样的求解策略假定了流体内部的速度场是无旋无散, 当然在单纯的模拟小尺度的流体时, 如此假定是合适的。这种方法也在 2020 年由 Huang 等人^[2]再次使用用来模拟磁流体, 该工作做出了许多非常逼真炫酷的效果。在计算数学领域, 学者们针对流体表面张力项提出求解策略, 2020 年 Takuya 等人^[2]使用表面网格和内部粒子来描述水滴, 通过表面网格计算表面张力同时通过内部粒子计算流体压强并将其耦合, 并通过不断调整网格顶点位置来改善网格质量, 以此提高表面张力计算的准确性, 然而调整网格质量也给网格拓扑变化以及大变形带来了麻烦, Huang 等人也在文章中说明该方法并未考虑水滴的拓扑变化。当然, 也有学者将流体表面张力单独提取出来计算, 实际上流体表面张力对应了一个曲面上的拉普拉斯贝尔特拉米算子^[2], 当前学者们针对点云表示的曲面构造对应的离散算子。在 2013 年, Rongjie 等人^[2]提出在每一点局部的重建 Delauney 网格来构造拉普拉斯算子, Diego 等人^[2]使用局部径向基函数方法构造, Liang 等人^[2]提出局部构造二次曲面以及使用二次拟合的策略来构造。当然, 由于以上算子构造方式都是以局部的方式给出, 这也损失了拉普拉斯方程的对称性。

1.2.3 表面重构

从前文中可以发现, 为了计算表面张力, 我们需要给出流体的表面表示, 而为了在粒子表示的流体上计算表面张力, 我们便需要定义出粒子表示流体的表面。在三维重建的领域, 学者们已经针对表面重构做了许多的工作。表面重建分成两种思路, 第一种是现实曲面重建, 显式重建可以直接处理没有附加信息的点云, 其思路主要是基于 Delauney 三角化理论。Amenta 等人^[2]在 1998 年结合三维 Voronoi 图和 Delauney 三角剖分提出三维的 Crust 算法, 他们在 2001 年对其改进提出 Power Crust 算法, 其借鉴了 α 形的思想, 使用加权 Voronoi 图改进, 提高了对尖锐特征以及高噪声点云的处理能力。这类显式重建方法都涉及到 Delauney 三角剖分, 需要较大的计算开销。另一类算法则是使用隐式曲面来拟合点云, 大多数隐式曲面重建算法都与 Blinn 于 1982 年^[2]的思想相似, 他们都使用局部基函数来重构隐式曲面。在 1991 年, Muraki 等人^[2]提出使用 Blobby 建模方法, 这也是一种类似径向基函数的做法。

Hoppe 等人在 1992 年^[2]提出使用局部符号距离场来重构隐式曲面。Carr^[2]同样使用局部径向基函数,结合优化技巧给出了一种快速计算的流程。2019 年, Huang 等人^[2]提出变分隐式曲面法,该方法对稀疏非均匀点集有着较好的效果与计算性能。

由于点云数据一半非常庞大,因此在构造隐式曲面时,所对应的优化问题都有着非常庞大的规模,如此构造的求解矩阵也是非常庞大的。为了降低内存消耗, Morse 等人^[2]使用局部紧的径向基函数插值方式,大大减少了计算代价与内存消耗。同样,为了降低内存消耗, Jun 等人在 2016 年^[2]使用自适应八叉树 T 样条来进一步降低内存消耗。Yusuf 等人^[2]结合步进迭代逼近与局部紧支撑基函数,进一步加快了求解速度。

1.3 研究内容

本文主要研究了如何在物质点法框架下计算表面张力,并提出了一个快速曲面重构算法。过去的有许多针对表面张力模拟的算法,但也是由于其针对性,尽管对于表面张力模拟效果较好,但是难以在该框架下求解其他现象,导致其应用的局限性。同时近年来的也开始发展出许多普适的模拟算法,比如改进的 SPH,尽管其本身就是从流体算法发展而来,但是其对表面张力的模拟并没有很好的支持。同时近年来一个针对 SPH 改进的算法,被称为移动最小二乘再生核算法^[2],不过该方法也没有考虑表面张力,只是将弹性体纳入求解框架。另一个方法便是近期火热的物质点法,尽管物质点法已经能够模拟很多现象,且已经能够轻易的将许多物理方程耦合求解,使其都容纳到一个框架之下,极大的展现了它的普适性但是其对于流体,特别是流体表面张力研究甚少。本文便从物质点法出发,改进物质点法计算流程,将流体表面张力模拟也融合进物质点法的求解框架之中。

在解决物质点法模拟表面张力的过程中,本文还基于 Yusuf 等人的步进迭代逼近^[2],改进了隐式曲面拟合算法,让整个算法有更高的并行效率,同时又更好的收敛性,使其相较原有算法至少达到三个数量级的加速,同时大大降低内存消耗。特别的,我们发现该隐式曲面拟合算法的计算流程与物质点法部分的计算流程机器类似,这也启发我们利用物质点法的数据结构来计算隐式曲面,经过分析,该计算过程不会带来额外的内存消耗。

在计算表面张力时,为了获取隐式曲面的法向,我们提出了一个更高效的法向逼近算法,该算法相较于直接计算函数梯度有着更低的复杂度,同时更易于底层并行。

在耦合物质点法的内部压强和表面张力之时,我们还给出了一个提高数值稳定性的技巧,即将表面粒子嵌入到内部粒子,这样可以更直接的将表面张力传导到内部粒子与之耦合,相较于直接通过背景网格耦合两者,嵌入内部粒子的方法提高了

计算的稳定性。

本文的主要工作总结如下：

(一) 改进了一个新的点云拟合隐式曲面的算法，同时提出一个相应的快速法向计算方法，并将相应算法耦合进物质点法框架中。

(二) 给出了物质点法模拟表面张力的计算流水线，并给出了相应的数值离散形式，同时在实现上使用嵌入内部粒子技巧提高了数值稳定性。

本文的章节内容安排如下：

第一章，绪论，主要介绍了物理模拟的背景以及近期有关物质点法，表面张力，表面重构三方面的相关工作，同时引出文章需要解决的问题。最后总结了文章的研究内容以及文章结构。

第二章，连续介质力学，给出了文章所需要的基本理论，从欧拉视角和拉格朗日视角来描述我们求解的问题，以此对应物质点法的网格法与粒子法结合，给出了方程离散的基础。同时给出了表面张力的形式，这部分内容也引导了我们在物质点法中如何计算表面张力。

第三章，表面重建，主要给出了如何构造一个粒子表示的流体表面。同时给出我们的隐式曲面拟合算法，并与之前的文章进行性能对比试验来说明我们方法的优势。最后介绍了快速法向计算方法，在理论上分析了快速法向算法能够提高直接梯度计算的性能。

第四章，数值方法，主要介绍了如何将第一章给出的连续控制方程离散化，同时给出了嵌入粒子法以及一些工程修正方法，保证求解方程的稳定性。最后将第三章的表面重建方法耦合到物质点法中，总结出对应的计算流水线。

第五章，总结与展望，给出了本文存在的局限性，以及提出相应的问题以及进一步研究的方向。

1.4 本章小结

本章首先介绍了物理模拟在各种场景的应用以及存在的问题，之后介绍了前人在物质点法、表面张力模拟、表面重建三个方面的工作，并由此引出本文所解决的问题。之后阐明了并总结了本文的研究内容，并给出了各章内容的概要。

2 连续介质力学

2.1 引言

连续介质力学是本文构建控制方程的理论基础。基于连续介质力学，我们刻画流体以及流体表面，并从两个视角来描述物体运动，以此构建动力学方程。为了构建完整的控制方程，我们还需要给出本构模型，即形变与应力的关系，这里我们给出对应的流体的弱可压缩模型以及表面张力模型。

2.2 欧拉视角下的动力学

欧拉视角，即物理量的定义域为 \mathbb{R}^3 ，对物体的刻画由空间中的密度场给出。与拉格朗日视角的区别我们将在 2.3 小节中给出。

2.2.1 欧拉视角下的质量守恒定律

记 $\rho : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}$ 为空间中随时间变化的密度场， $v : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}^3$ 为空间中随时间变化的速度场，此处默认向量为列向量。

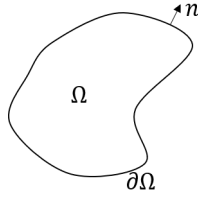


图 2.1 区域符号示例

现在考虑一个闭区域 Ω ， Ω 的边界记为 $\partial\Omega$ ，边界上的外法向记为 n ，同时假定 ρ 和 v 在 Ω 的一个开邻域内足够光滑。根据质量守恒，我们有 Ω 上质量的变化率为 $\partial\Omega$ 上质量的流出率和流入率之和。即

$$\begin{aligned} \frac{d}{dt} \int_{\Omega} \rho(x, t) dx &= - \int_{\partial\Omega} \rho v \cdot n ds \\ \int_{\Omega} \frac{\partial}{\partial t} \rho(x, t) dx &= - \int_{\partial\Omega} \text{div}(\rho v) dx \end{aligned}$$

如果 ρ 在 \mathbb{R}^3 上都能达到足够光滑，那么由 Ω 的任意性，可知

$$\frac{\partial}{\partial t} \rho(x, t) + \text{div}(\rho v) = 0 \quad (2.1)$$

2.2.2 欧拉视角下的动量守恒定律

现在考虑闭区域 Ω 上的动量变化。根据连续介质力学中的牛顿运动定律^[2]，闭区域 Ω 上的动量变化可以分为三部分，第一部分为物质流入流出 Ω 导致的动量变

化, 第二部分为作用在 Ω 边界上的力导致的动量变化, 第三部分为作用在 Ω 内部物质上的力 (一般为重力) 产生的作用变化。即

$$\frac{d}{dt}\bigg|_{t=t_0} \int_{\Omega} \rho v dx = - \int_{\partial\Omega} \rho v (v \cdot n) ds + \int_{\partial\Omega} \sigma \cdot n ds + \int_{\Omega} \rho g dx \quad (2.2)$$

在 (2.2) 式中, g 为重力加速度, σ 为 Cauchy 应力^[2]。特别的, σ 是一个三阶对称矩阵, 其对称性来源于角动量守恒^[2], 我们将在 2.4 节中从另一个角度说明其对称性。

由于 (2.2) 式中 Ω 选择的任意性, 我们有

$$\frac{\partial}{\partial t}\bigg|_{t=t_0} (\rho v) = -\operatorname{div}(\rho v^T v) + \operatorname{div}(\sigma) + \rho g \quad (2.3)$$

矩阵函数散度 div 定义如下:

$$\begin{aligned} \operatorname{div} : C^1(\mathbb{R}^3; \mathbb{R}^{3 \times 3}) &\rightarrow C^1(\mathbb{R}^3; \mathbb{R}^3) \\ \operatorname{div}(A)_i &:= \sum_j \partial_j A_{ij} \end{aligned}$$

(2.2) 式变形为

$$\frac{\partial}{\partial t}\bigg|_{t=t_0} (\rho v) + \operatorname{div}(\rho v^T v) = \operatorname{div}(\sigma) + \rho g \quad (2.4)$$

2.3 拉格朗日视角下的动力学

在上一节中, 我们根据质量守恒和动量守恒得到了两个方程 (方程 (2.1) 与方程 (2.4)), 非常重要的一点是, 上述方程并没有使用任何有关自然状态—材料在不施加外力的静止状态—的信息。在之后的章节中, 我们将假定, $t=0$ 是处于自然状态, 并且只考虑 $t \geq 0$ 的情况。

假定在自然状态下 ($t=0$), 材料占据的空间为 $\Omega_0 \subset \mathbb{R}^3$, 此时也称 Ω_0 为参考构型。对于任意给定的 $t \in (0, +\infty)$, 此时材料占据的空间为 $\Omega_t \subset \mathbb{R}^3$, 相对于参考构型 Ω_0 , 称 Ω_t 为当前构型。

拉格朗日视角和欧拉视角的区别主要是函数的定义域。在上一节中, 我们所有函数的定义域都在 \mathbb{R}^3 上, 而本节我们将把视角限制在材料上, 例如对于任意时刻 t , 有 Ω_t 上的实值函数, $f(\cdot, t) : \Omega_t \rightarrow \mathbb{R}$ 。

2.3.1 形变映射, 形变梯度, 速度

为了描述材料相对于参考构型发生的形变, 我们引入形变映射

$$\phi : \Omega_0 \times [0, +\infty) \rightarrow \mathbb{R}^3$$

, 形变映射同时也给出了物体的运动轨迹。为了方便, 我们记 $\phi_t(x) = \phi(x, t)$, 实际上我们还假定 $\{\phi_t(x) : t \in \mathbb{R}\}$ 构成了一个单参数变换群, 并且关于 t 至少二阶光滑, 关于 X 至少有连续的一阶导数。

有了形变映射, 我们可以引入对局部形变的刻画, 即形变梯度 $F(X, t) := \frac{\partial \phi_t(X)}{\partial X}$ 。实际上, 形变梯度 $F(X, t)$ 还可以视为 X 处切空间的映射, 即

$$F(X, t) : T_X \Omega_0 \rightarrow T_x \Omega_t$$

对于任意的 $\partial_X \in T_X \Omega_0$, $\partial_x = F(X, t)[\partial_X] \in T_x \Omega_t$, 即形变梯度将 X 点的切向量 ∂_X 经过平移起始点, 旋转方向, 拉伸长度后得到 x 点的切向量 ∂_x , 如图 2.2 所示。

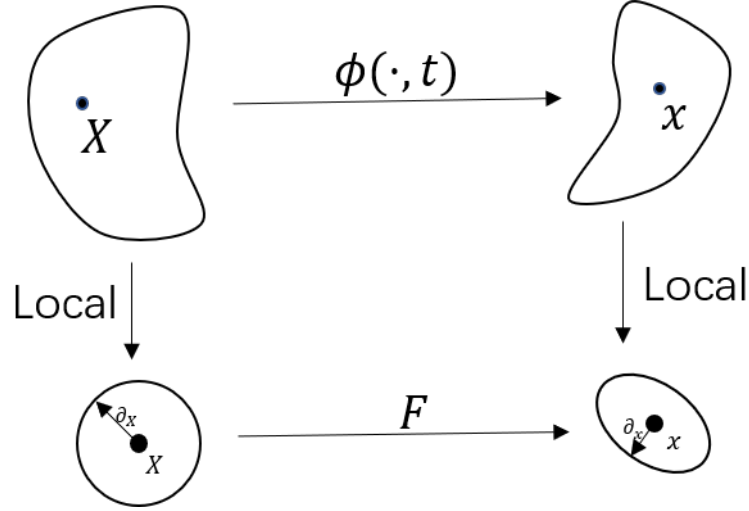


图 2.2 形变与形变梯度

同样的, 形变映射作为单参数变换群, 自然的导出速度的定义 $V(X, t) := \frac{\partial \phi_t(X)}{\partial t}$ 。值得注意的是, $V(X, t)$ 的定义域是在 Ω_0 上, 其值域是一个起始点在 Ω_t 上的一个三维列向量, 其与欧拉视角下定义的速度关系为 $V(X, t) = v(\phi_t(X), t)$, 如果记 $x = \phi_t(X)$, 则 $V(X, t) = v(x, t)$ 。

考察 F 关于时间的一阶导

$$\begin{aligned} \frac{d}{dt} F(X, t) &= \frac{d}{dt} \frac{\partial}{\partial X} \phi_t \\ &= \frac{\partial}{\partial X} \frac{d}{dt} \phi_t \\ &= \frac{\partial}{\partial X} V(X, t) \\ &= \frac{\partial v(\phi_t(X), t)}{\partial x} \frac{\partial \phi_t}{\partial X} \\ &= \frac{\partial v(x, t)}{\partial x} \frac{\partial \phi_t}{\partial X} \\ &= \nabla v \cdot F \end{aligned}$$

整理后我们得到 F 和 v 的关系

$$\dot{F} = \nabla v \cdot F \quad (2.5)$$

2.3.2 拉格朗日视角下的质量守恒

在引入了形变 ϕ_t 之后,很自然的有 $\hat{\Omega}_t = \{\phi_t(X) : X \in \hat{\Omega}_0 \subset \Omega_0\}$ (简记为 $\phi_t(\hat{\Omega}_0)$), 且 $\hat{\Omega}_t \subset \Omega_t$, 根据质量守恒, 形变不会影响质量, 即

$$\int_{\hat{\Omega}_0} \rho(x, 0) dx = \int_{\hat{\Omega}_t} \rho(x, t) dx = \int_{\phi_t(\hat{\Omega}_0)} \rho(x, t) dx$$

如果我们将 $\hat{\Omega}_t$ 的质量记为 $M[\hat{\Omega}_t]$, 则有 $\frac{d}{dt} M[\hat{\Omega}_t] = 0$ 。而

$$\begin{aligned} \frac{d}{dt} \int_{\phi_t(\hat{\Omega}_0)} \rho(x, t) dx &= \frac{d}{dt} \int_{\hat{\Omega}_0} \rho(\phi_t(X), t) \det(F(X, t)) dX \\ &= \int_{\hat{\Omega}_0} \frac{d}{dt} [\rho(\phi_t(X), t) \det(F(X, t))] dX \\ &= 0 \end{aligned}$$

记 $R(X, t) := \rho(\phi_t(X), t)$, $J(X, t) := \det(F(X, t))$, 则根据 $\hat{\Omega}_0$ 的任意性, 有 $\frac{d}{dt} [R(X, t) J(X, t)] = 0$, 即

$$R(X, t) J(X, t) = R(X, 0) \quad (2.6)$$

(2.6) 式即为拉格朗日视角下的质量守恒方程。

2.3.3 拉格朗日视角下的动量守恒

在拉格朗日视角下, $\hat{\Omega}_0$ 的动量变化可以表示为作用在 $\hat{\Omega}_0$ 边界上的力以及作用在 $\hat{\Omega}_0$ 内部的力之和。即

$$\left. \frac{d}{dt} \right|_{t=t_0} \int_{\hat{\Omega}_0} R(X, 0) V(x, t) dX = \int_{\partial \hat{\Omega}_0} P \cdot N dS + \int_{\hat{\Omega}_0} R(X, 0) g dX \quad (2.7)$$

上式实际上对于任意的 $\hat{\Omega}_0$ 成立, 因此有以下

$$R(X, 0) \frac{\partial}{\partial t} V(X, t) = \text{DIV}(P) + Rg \quad (2.8)$$

此处 DIV 是 Ω_0 上的散度算子, 为了与 Ω_t 上进行区分记为大写。其中 P 是应力的另一种表述形式, 称为 **First Piola-Kirchhoff** 应力^[2], 其满足 $\int_{\partial \hat{\Omega}_0} P \cdot N dS = \int_{\partial \hat{\Omega}_t} \sigma \cdot n ds$ 我们将在下一节本构关系中更详细的给出 P 的计算方式以及与 σ 的关系。

下面我们再次从拉格朗日视角推导欧拉视角下的动量守恒, 来证明两者的等价

性。

$$\begin{aligned}
\frac{d}{dt}\Big|_{t=t_0} \int_{\hat{\Omega}_0} R(X, 0) V(x, t) dX &= \frac{d}{dt}\Big|_{t=t_0} \int_{\hat{\Omega}_0} R(X, t) v(\phi_t(X), t) J(X, t) dX \\
&= \int_{\hat{\Omega}_0} \frac{d}{dt}\Big|_{t=t_0} [\rho(\phi_t(X), t) v(\phi_t(X), t) J(X, t)] dX \\
&= \int_{\hat{\Omega}_0} \frac{d}{dt}\Big|_{t=t_0} [\rho(\phi_t(X), t) v(\phi_t(X), t) J(X, t)] dX \\
&= \int_{\hat{\Omega}_0} v(\phi_{t_0}(X), t_0) \frac{d}{dt}\Big|_{t=t_0} [\rho(\phi_t(X), t) J(X, t)] \\
&\quad + \rho(\phi_{t_0}(X), t_0) J(X, t_0) \frac{d}{dt}\Big|_{t=t_0} v(\phi_{t_0}(X), t_0) dX \\
&= \int_{\hat{\Omega}_0} v(\phi_{t_0}(X), t_0) \frac{d}{dt}\Big|_{t=t_0} R(X, 0) \\
&\quad + \rho(\phi_{t_0}(X), t_0) J(X, t_0) \frac{d}{dt}\Big|_{t=t_0} v(\phi_t(X), t) dX \\
&= \int_{\hat{\Omega}_0} \rho(\phi_{t_0}(X), t_0) J(X, t_0) \frac{d}{dt}\Big|_{t=t_0} v(\phi_t(X), t) dX \\
&= \int_{\hat{\Omega}_{t_0}} \rho(x, t_0) \frac{d}{dt}\Big|_{t=t_0} v(\phi_{t,t_0}(x), t) dx
\end{aligned}$$

这里 $\phi_{t,t_0} = \phi_t \cdot \phi_{t_0}^{-1}$ ，带入 (2.6) 中就有

$$\int_{\hat{\Omega}_{t_0}} \rho(x, t_0) \frac{d}{dt}\Big|_{t=t_0} v(\phi_{t,t_0}(x), t) dx = \int_{\hat{\Omega}_{t_0}} \operatorname{div}(\sigma) dx + \int_{\hat{\Omega}_{t_0}} \rho g dx$$

这里由于 $\hat{\Omega}_{t_0}$ 为 Ω_t 内的任意闭区域，因此

$$\rho(x, t_0) \frac{d}{dt}\Big|_{t=t_0} v(\phi_{t,t_0}(x), t) = \operatorname{div}(\sigma) + \rho g$$

其中

$$\frac{d}{dt}\Big|_{t=t_0} v(\phi_{t,t_0}(x), t) = \frac{\partial}{\partial t}\Big|_{t=t_0} v(x, t) + \sum_i \frac{\partial \phi_{t,t_0}^i}{\partial t} \cdot \frac{\partial}{\partial x_i} v(x, t_0) = \frac{\partial}{\partial t}\Big|_{t=t_0} v(x, t) + v^T \nabla v$$

这里 ϕ_{t,t_0}^i 为 ϕ_{t,t_0} 的第 i 个分量。此处引入材料时间导数 $\frac{D}{Dt} f := \frac{\partial}{\partial t} f + v \nabla f$ ，将其代入上式中整理得

$$\rho \frac{Dv}{Dt} = \operatorname{div}(\sigma) + \rho g \tag{2.9}$$

该方程实际与 (2.4) 式等价，由于其更简洁的表述形式，我们之后将称 (2.9) 式为欧拉视角下的动量守恒。

$$\frac{\partial}{\partial t}(\rho v) + \operatorname{div}(\rho v^T v) = \operatorname{div}(\sigma) + \rho g$$

可考察方程的左端项

$$\begin{aligned}
 \frac{\partial}{\partial t}(\rho v) + \operatorname{div}(\rho v^T v) &= \rho \frac{\partial}{\partial t} v + v \frac{\partial}{\partial t} \rho + \rho v^T \nabla v + v \operatorname{div}(\rho v) \\
 &= \rho \left(\frac{\partial}{\partial t} v + v^T \nabla v \right) + v \left(\frac{\partial}{\partial t} \rho + \operatorname{div}(\rho v) \right) \quad \text{第二项为质量守恒} \\
 &= \rho \frac{Dv}{Dt}
 \end{aligned}$$

带入即知, (2.4) 式是 (2.8) 式的展开。

2.4 本构关系

在前两节中, 我们在动量守恒方程中引入了应力项, 其决定了速度如何随着时间变化。本节我们将给出如何通过形变来给出应变张量, 其中应变张量和形变的函数关系被称为本构关系。事实上, 本构关系直接决定了我们模拟的物体会表现出何种特质。

流体被认为是一种不可压缩的物质, 用数学的语言就是说其形变梯度 F 满足 $\det(F) \equiv 1$, 然而在实践中, 为了更好的配合物质点法的使用, 我们从要求 $\det(F) \equiv 1$ 转变为使用形变能量密度来惩罚形变对体积的影响, $W(F) := \frac{\lambda}{2}(\det(F) - 1)^2$ 。可以发现, 对任意的 $F \in \mathbb{R}^{3 \times 3}, U \in SO_3$, 始终有 $W(UF) = W(F)$, 该特性被称为旋转不变性, 根据诺特定理^[7], 如果能量密度是旋转不变的, 那么所得到的系统是角动量守恒的。

在有了能量密度函数, 我们推导能量密度变化量和形变梯度的变化量之间的关系

$$\begin{aligned}
 \delta W(F) &= \frac{\lambda}{2} \delta(\det(F) - 1)^2 \\
 &= \lambda(\det(F) - 1) \delta(\det(F) - 1) \\
 &= \lambda(\det(F) - 1) \delta(\det(F))
 \end{aligned}$$

为了计算 $\delta \det(F)$, 首先考察 $\left. \frac{d}{dt} \right|_{t=0} \det(F + tI)$, 由矩阵的特征多项式得

$$\det(F + \epsilon I) = \epsilon^n + \epsilon^{n-1} \operatorname{Tr}(F) + \dots + \det(F)$$

则等式两边同时除 ϵ^n 得

$$\det\left(\frac{1}{\epsilon} F + I\right) = 1 + \frac{1}{\epsilon} * \operatorname{Tr}(F) + \dots + \frac{1}{\epsilon^n} \det(F)$$

此时将 $t \neq 0$ 代入有

$$\det(tF + I) = 1 + t \operatorname{Tr}(F) + \dots + t^n \det(F)$$

同时验证 $t = 0$ 可知上式恒成立。因此 $\frac{d}{dt}\big|_{t=0} \det(tF + I) = \text{tr}(F)$ ，现在对于任意的 t_0 以及 $\mathbb{R}^{3 \times 3}$ 中通过 $A(t_0)$ (假设 $A(t_0)$ 可逆) 的一条光滑路径 $A(t)$ ，我们有对应的通过 I 的路径 $A(t_0)^{-1}A(t)$ ，其对应的切向量为 $A(t_0)^{-1}\frac{d}{dt}\big|_{t=t_0} A(t)$ ，则

$$\begin{aligned} \frac{d}{dt}\big|_{t=t_0} \det(A(t)) &= \lim_{h \rightarrow 0} \frac{\det(A(t_0 + h)) - \det(A(t_0))}{h} \\ &= \det(A(t_0)) \lim_{h \rightarrow 0} \frac{\det(A(t_0)^{-1}A(t_0 + h)) - 1}{h} \\ &= \det(A(t_0)) \text{tr}(A(t_0)^{-1} \frac{d}{dt}\big|_{t=t_0} A(t)) \\ &= \det(A(t_0)) A(t_0)^{-T} : \frac{d}{dt}\big|_{t=t_0} A(t) \end{aligned} \quad (2.10)$$

这里 $A : B := \sum_{i,j} A_{ij} B_{ij}$ 。

故 $\delta \det(F) = \det(F) F^{-T} : \delta F$ ，代入 $\delta W(F)$ 之中即有

$$\delta W(F) = \lambda(\det(F) - 1) \det(F) F^{-T} : \delta F$$

，其中 $P = \lambda(\det(F) - 1) \det(F) F^{-T}$ 被称为 **First Piola-Kirchhoff** 应力，其与 **Cauchy** 应力 σ 的关系^[2] 为

$$J\sigma = PF^T$$

我们将在本小节最后给出该关系式的证明，代入可知柯西应力 $\sigma = \lambda(\det(F) - 1)I$ 。

这里的 $W(F)$ 实际上只给出了物体的抗压缩性的性质，我们还需要给出表面张力的能量密度函数。根据^[2]，表面能实际和物体表面积成正比，表面张力为表面能的梯度。那么我们首先可以形变和表面能量的关系

$$S(\phi_t) = k \int_{\phi_t(\partial\Omega_0)} ds \quad (2.11)$$

为了更好的计算其梯度，我们对 (2.10) 式做一些处理。此处引入一些微分流形的记号^[2]，记 f^* 为 f 导出的拉回映射， f_* 为推前映射， $ds, d\tilde{s}$ 分别为曲面 $\partial\Omega_{t_0}, \partial\Omega_t$ 上的面积二形式，其为空间体积形式 dv 内乘法向的结果，即 $ds(X, Y) = dv(n, X, Y)$ ，我们记 $ds = dv \odot n$ ， $\tilde{p} = \phi_{t_0, t}(p)$ ，则对任意的 $t \in [0, t_0)$ ， $\tilde{X}, \tilde{Y} \in T_{\tilde{p}}\Omega_t$ 有如下

$$\begin{aligned} (\phi_{t_0, t}^* ds)(\tilde{X}, \tilde{Y}) &= (\phi_{t_0, t}^*(dv \odot n))(\tilde{X}, \tilde{Y}) \\ &= ((\phi_{t_0, t}^* dv) \odot (\phi_{t_0, t}^* n))(\tilde{X}, \tilde{Y}) \\ &= \det(F(x; t_0, t)) dv(\langle F^{-1}(x; t_0, t)n, \tilde{n} \rangle \tilde{X}, \tilde{Y}) \\ &= \det(F(x; t_0, t)) \langle F^{-1}(x; t_0, t)n, \tilde{n} \rangle dv(\tilde{n}, \tilde{X}, \tilde{Y}) \\ &= \det(F(x; t_0, t)) \langle n, F^{-T}(x; t_0, t)\tilde{n} \rangle dv(\tilde{n}, \tilde{X}, \tilde{Y}) \end{aligned} \quad (2.12)$$

上式中 $F(x; t_0, t) = \frac{\partial \phi_{t_0, t}(x)}{\partial x}$, \tilde{n} 为 $T_p \Omega_t$ 单位法向, $\langle \cdot, \cdot \rangle$ 为空间内积, 容易验证 $F^{-T}(x; t_0, t)\tilde{n}$ 与 $T_p \Omega_0$ 垂直, 故 $\langle n, F^{-T}(x; t_0, t)\tilde{n} \rangle = \|F^{-T}(x; t_0, t)\tilde{n}\|$, 即 $\|F^{-T}(x; t_0, t)\tilde{n}\|n = F^{-T}(x; t_0, t)\tilde{n}$, 那么有

$$(\phi_{t_0, t}^* ds) = \det(F(x; t_0, t)) \|F^{-T}(x; t_0, t)\tilde{n}\| d\tilde{s}$$

则代入表面能表达式有

$$\begin{aligned} S(\phi_{t_0}) &= k \int_{\partial \Omega_{t_0}} ds \\ &= k \int_{\phi_{t_0, t}(\partial \Omega_t)} ds \\ &= k \int_{\partial \Omega_t} \phi_{t_0, t}^* ds \\ &= k \int_{\partial \Omega_t} \det(F(x; t_0, t)) \|F^{-T}(x; t_0, t)\tilde{n}\| d\tilde{s} \end{aligned}$$

至此, 我们得到表面能的另一个表达形式

$$S(\phi_{t_0}) = k \int_{\partial \Omega_t} \det(F(x; t_0, t)) \|F^{-T}(x; t_0, t)\tilde{n}\| d\tilde{s} \quad (2.13)$$

(2.21) 式将是表面张力离散化的基础。

最后, 我们给出 $J\sigma = PF^T$ 的证明,

$$\begin{aligned} \int_{\partial \hat{\Omega}_t} \sigma \cdot n ds &= \int_{\phi_t(\partial \hat{\Omega}_0)} \sigma \cdot n ds \\ &= \int_{\partial \hat{\Omega}_0} \phi_t^*(\sigma \cdot n)(\phi_t^* ds) \\ &= \int_{\partial \hat{\Omega}_0} \sigma \cdot n J \|F^{-T} N\| dS \\ &= \int_{\partial \hat{\Omega}_0} \sigma F^{-T} \cdot F^T n J \|F^{-T} N\| dS \\ &= \int_{\partial \hat{\Omega}_0} J \|F^{-T} N\| \sigma F^{-T} \cdot \frac{1}{\|F^{-T} N\|} N dS \\ &= \int_{\partial \hat{\Omega}_0} J \sigma F^{-T} \cdot N dS \end{aligned}$$

而 $\int_{\partial \hat{\Omega}_0} P \cdot N dS = \int_{\partial \hat{\Omega}_t} \sigma \cdot n ds$, 因此 $\int_{\partial \hat{\Omega}_0} J \sigma F^{-T} \cdot N dS = \int_{\partial \hat{\Omega}_0} P \cdot N dS$, 由于上式对任意 $\partial \hat{\Omega}_0$ 成立, 则 $J \sigma F^{-T} = P$ 得 $J \sigma = P F^T$ 。

2.5 本章小结

本章基于连续介质力学得到了一系列的控制方程

$$\frac{\partial}{\partial t} \rho(x, t) + \operatorname{div}(\rho v) = 0 \quad \text{欧拉视角下的质量守恒方程}$$

$$R(X, t)J(X, t) = R(X, 0) \quad \text{拉格朗日视角下的质量守恒方程}$$

$$\rho \frac{Dv}{Dt} = \operatorname{div}(\sigma) + \rho g \quad \text{欧拉视角下的动量守恒方程}$$

$$R(X, 0) \frac{\partial}{\partial t} V(X, t) = \operatorname{DIV}(P) + Rg \quad \text{拉格朗日视角下的质量守恒方程}$$

以上四个方程分别从两种视角分别等价的刻画了质量守恒和动量守恒，这两种视角也会在物质点法离散化的过程中体现出来。其次，我们还给出了形变梯度与速度的关系

$$\dot{F} = \nabla v \cdot F$$

该方程将会在物质点法如何获得下一时刻的形变梯度中起到关键作用。最后，我们给出了弱可压缩能量 $W(F) = \lambda(J - 1)^2$ 来刻画流体的特性，该能量也给出了 **First Piola-Kirchhoff** 应力和 **Cauchy** 应力的计算方式。其次我们给出了表面能的计算方法

$$S(\phi_{t_0}) = k \int_{\partial\Omega_{t_1}} \det(F(x; t_0, t_1)) \|F^{-T}(x; t_0, t_1) \tilde{n}\| d\tilde{s}$$

该计算形式下，积分域不随着 ϕ_{t_0} 变化而变化，为我们离散计算以及变分都带来了便利，而表面张力将由表面能变分获得，这一步我们将在离散化的过程中来计算。

3 表面重建

3.1 引言

为了计算表面张力，我们需要一种能够从点云中获取表面和表面法向的方法。本文使用的方法是使用粒子水平集方法^[2]获得一个粗糙的一阶连续隐式曲面，并使用 **MarchingCube**^[2] 算法重构表面网格。然而该网格的质量只能做到一阶连续，计算得到的法向在面片之间并不连续，难以应用在表面张力所需的法向计算。本文在该网格上进行泊松圆盘采样，然后结合已经存在的 **IPIA** 算法，提出一种新的快速重建算法以此获得一个二阶光滑的隐式曲面，然后基于该隐式曲面来计算表面点云的法向。

3.2 粒子水平集方法

记 \mathcal{V} 为点云集合， $v \in \mathcal{V}$ 为粒子，记 $d(x, v) \in \mathbb{R}$ 为空间上 x 的点到粒子 v 的距离，则 $\mathcal{S}(v) = \{x \in \mathbb{R}^3 : d(x, v) = r\}$ 为半径为 r ，圆心在 v 的球，如图??左所示，蓝色圆圈为二位情况的 $\mathcal{S}(v)$ 。

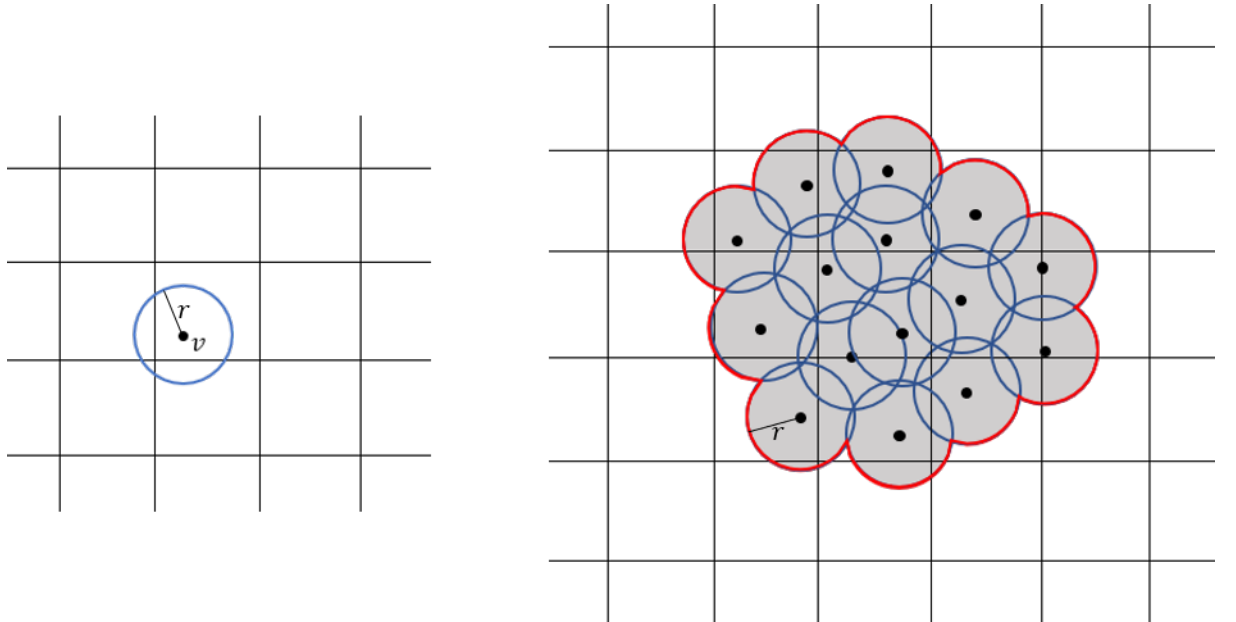


图 3.1 二维情况 $\mathcal{S}(v)$ (左) 与 $\mathcal{S}(\mathcal{V})$ (右)

记 $d(x, \mathcal{V}) := \min\{d(x, v) : v \in \mathcal{V}\}$ ，则 $\mathcal{S}(\mathcal{V}) = \{x \in \mathbb{R}^3 : d(x, \mathcal{V}) = r\}$ 为点云的外延边界。

如图??右所示，红色的外延边界为所求 $\mathcal{S}(\mathcal{V})$ 。该红色边界初步确定了该点云的

边界，灰色部分界定了物体内部。下一步我们将从利用红色边界和背景网格提取出一个粗糙的网格。

3.3 水平集网格提取

Marching cube 通常用于三维标量场的水平集的可视化，本文使用 Marching cube 算法来获取水平集的三角网格，在本文中我们的水平集为 $\mathcal{S}(\mathcal{V}) = \{x \in \mathbb{R}^3 : d(x, \mathcal{V}) = r\}$ 。

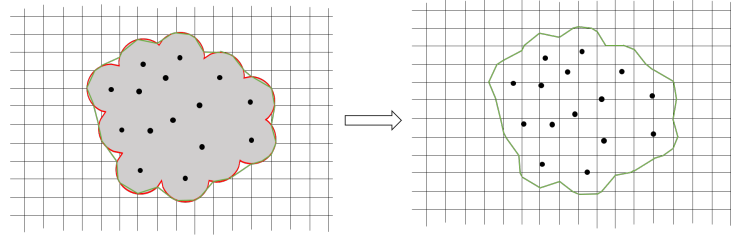


图 3.2 二维点云轮廓提取图示

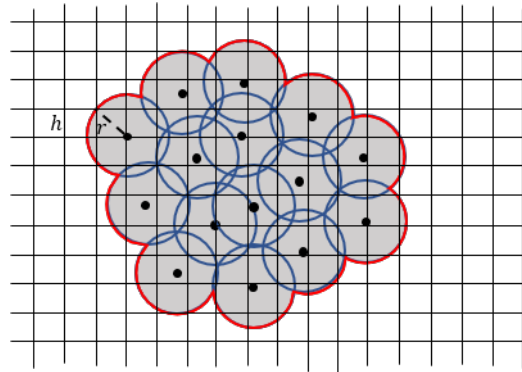


图 3.3 二维情况的 r 与 h 的关系

本小节我们的目标只是提取出一个大致的点云的表面轮廓，如图??所示。我们首先将空间离散化成一个个均匀的立方体，我们称之为背景网格，并对每一个正方体格点放置一个标志符，默认为 0，此处记立方体宽度为 h ，为了保证球能够覆盖到一个足够大的面积，此处我们选择 $r = \sqrt{3}h$ ，二维情况为 $r = \sqrt{2}h$ ，具体如图??所示。本文为了节省内存，具体实现使用了八叉树数据结构。之后对每一个粒子操作，将粒子周围的格点做标记，如果格点距离该粒子半径不超过 r ，则将格点标记为 1，如此下来，所有的正方体格点都被标记为两种状态。之后我们对正方体按格点状态进行分类，由于每个格点有两种状态，每个正方体有 8 个格点，如此一来便有 $2^8 = 256$ 种状态，在合并旋转和对称的情况后，可以简化成 15 种，可以分类成如图??所示。之后我们再将网格顶点位置调整到合适的位置，并建立三角网格数据结构。具体的算法描述见 Algorithm ??。

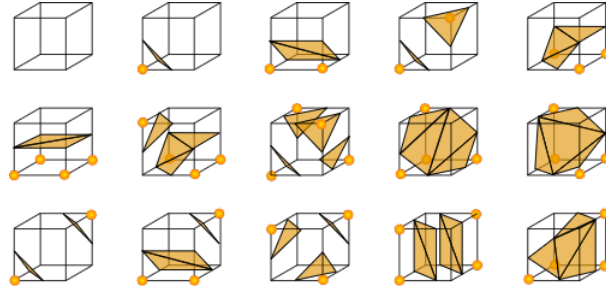


图 3.4 Marching cube 列表

Algorithm 1 Marching cube**Input:** 点云集合 \mathcal{V} , 立方体宽度 h **Output:** 三角网格位置集合 V , 面片集合 F

- 1: 准备空间网格数据结构 (八叉树), 并记 \mathcal{G} 为格点集合, \mathcal{E} 为立方体的边集合, \mathcal{C} 为立方体集合
- 2: $state(g) \in \{0, 1\}$ 为格点 $g \in \mathcal{G}$ 的状态, 初始化为 0
- 3: $P(e) \in \mathbb{R}^3$ 为边 $e \in \mathcal{E}$ 上的一个顶点
- 4: $r = \sqrt{3}h$
- 5: **for** $v \in \mathcal{V}$ **do**
- 6: **for** $g \in \{g \in \mathcal{G} : \|g - v\| < r\}$ **do**
- 7: $state(g)$ 标记为 1
- 8: **end for**
- 9: **end for**
- 10: **for** $e \in \{e \in \mathcal{E} : e \text{ 两端格点状态相异}\}$ **do**
- 11: 取 e 状态为 1 的顶点记为 g_1 , 状态为 0 的记为 g_0
- 12: 申请栈空间 p_{stack}
- 13: **for** $v \in \{v \in \mathcal{V} : \|g_1 - v\| < r\}$ **do**
- 14: 计算 $\mathcal{S}(v)$ 与 e 的交点并压入 p_{stack} 中
- 15: **end for**
- 16: 在 p_{stack} 中选取离 g_0 最近的点, 并赋值给 $P(e)$, 同时将 $P(e)$ 压入 V 中
- 17: **end for**
- 18: **for** $c \in \mathcal{C}$ **do**
- 19: 匹配 c 所对应的 Marching cube 列表中的状态
- 20: 构造相应的三角面表 f_s , 顶点选为对应 $P(e)$ 在 V 中得索引
- 21: 将 f_s 其压入 F 中
- 22: **end for**

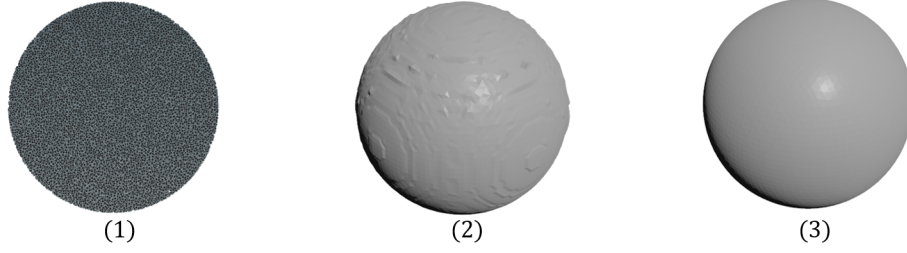


图 3.5 图 (1) 为输入点云, 图 (2) 为直接提取网格结果, 图 (3) 为 LSIPIA 拟合后提取的网格结果

从实验结果的图??(2) 中可以发现, **Marching cube** 算法确实提取出了一个点云的边界轮廓, 但是轮廓只能做到一阶连续, 法向明显不连续, 同时其三角网格质量无法无法保证, 甚至可以明显看出存在一些尖锐地方, 网格质量极差, 上述的问题给我们计算表面张力带来了很大的阻碍。

3.4 LSIPIA

为了解决法向不连续的问题以及网格质量差的问题, 我们使用二阶连续的隐式曲面来逼近三角网格, 因此隐式曲面的法向一阶连续。

3.4.1 隐式曲面的渐进迭代逼近

我们首先给出隐式曲面重建问题, 给定一个无序点集 $\mathcal{V} = \{v_i = (v_i^1, v_i^2, v_i^3)\}_{i=1}^n$ 以及点上的单位法向量 $\mathcal{N} = \{n_i = (n_i^1, n_i^2, n_i^3)\}_{i=1}^n$, 寻找一个函数 $f(x^1, x^2, x^3)$ 使得 $f(x^1, x^2, x^3)$ 的 0 水平集拟合该组无序点集 \mathcal{V} 。以数学的表达形式即为

$$\min_f \sum_{i=1}^n f(v_i^1, v_i^2, v_i^3)^2 \quad (3.1)$$

为了让问题可解, 我们选择 f 为有限个张量积 **B**-样条基函数的线性组合, 即

$$f(x^1, x^2, x^3) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} C_{ijk} B_i(x^1) B_j(x^2) B_k(x^3)$$

这里我们选择二阶 **B**-样条基函数, 其中二阶标准 **B**-样条函数

$$B(x) = \begin{cases} \frac{3}{4} - |x|^2 & 0 \leq |x| < \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - |x|)^2 & \frac{1}{2} \leq |x| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |x| \end{cases}$$

如图??所示, 这里 $B_i(x) := B(\frac{1}{h}(x - x_i))$, 这里的 x_i 本文选取为空间立方体网格格点坐标的一个分量, h 为立方体宽度。则最小化问题 (3.2) 转化成

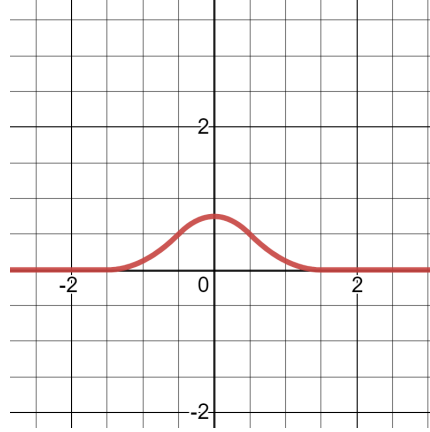


图 3.6 二次标准 B-样条基函数

$$\min_{C_{ijk}} E(C_{111}, C_{112}, \dots, C_{N_x N_y N_z}) := \frac{1}{2} \sum_{i=1}^n f(v_i^1, v_i^2, v_i^3)^2 \quad (3.2)$$

这里 C_{ijk} 张量积 B-样条系数。

我们记三变量 B-样条基函数为 $B_{ijk}(x^1, x^2, x^3) = B_i(x^1)B_j(x^2)B_k(x^3)$, 则上述问题变为

$$\begin{aligned} \min_{C_{ijk}} E(C_{111}, C_{112}, \dots, C_{N_x N_y N_z}) &:= \frac{1}{2} \sum_{r=1}^n \left(\sum_{ijk} C_{ijk} B_{ijk}(v_r^1, v_r^2, v_r^3) \right)^2 \\ &\Leftrightarrow \delta E(\{C_{ijk}\}) = 0 \\ &\Leftrightarrow \delta \left(\frac{1}{2} \sum_{r=1}^n \left(\sum_{ijk} C_{ijk} B_{ijk}(v_r^1, v_r^2, v_r^3) \right)^2 \right) = 0 \\ &\Leftrightarrow \sum_{r=1}^n \sum_{ijk} \sum_{uvw} C_{ijk} B_{ijk}(v_r^1, v_r^2, v_r^3) B_{uvw}(v_r^1, v_r^2, v_r^3) \delta C_{uvw} = 0 \\ &\Leftrightarrow \sum_{r=1}^n \sum_{ijk} C_{ijk} B_{ijk}(v_r^1, v_r^2, v_r^3) B_{uvw}(v_r^1, v_r^2, v_r^3) = 0 \end{aligned} \quad (3.3)$$

我们将 $\{B_{111}, B_{112}, \dots, B_{11N_z}, \dots, B_{1N_y1}, \dots, B_{N_x N_y N_z}\}$ 以字典序排列, 这样我们可配置如下矩阵

$$\mathbf{B} = \begin{bmatrix} B_{111}(v_1^1, v_1^2, v_1^3) & B_{112}(v_1^1, v_1^2, v_1^3) & \dots & B_{N_x N_y N_z}(v_1^1, v_1^2, v_1^3) \\ B_{111}(v_2^1, v_2^2, v_2^3) & B_{112}(v_2^1, v_2^2, v_2^3) & \dots & B_{N_x N_y N_z}(v_2^1, v_2^2, v_2^3) \\ \vdots & \vdots & & \vdots \\ B_{111}(v_r^1, v_r^2, v_r^3) & B_{112}(v_r^1, v_r^2, v_r^3) & \dots & B_{N_x N_y N_z}(v_r^1, v_r^2, v_r^3) \end{bmatrix}$$

同样对于 $\{C_{111}, \dots, C_{N_x N_y N_z}\}$ 我们有

$$\mathbf{C} = \begin{bmatrix} C_{111} \\ \vdots \\ C_{N_x N_y N_z} \end{bmatrix}$$

代入 (3.3) 最后一个式子中，我们可以得到一个紧凑的形式

$$\mathbf{B}^T \mathbf{B} \mathbf{C} = 0$$

然而，在实践中以上线性系统并非有唯一解，并且有平凡解 ($\mathbf{C} = 0$)。为了避免陷入平凡解，我们添加法向扰动。我们利用本小节开头提到的每一点上的法向信息 \mathcal{N} ，我们利用法向取一些偏移点 $\mathcal{Q} = \{q_i = v_i + \epsilon n_i : v_i \in \mathcal{V}, n_i \in \mathcal{N}\}$ 。然后尽可能要求 $f(q_i) = \epsilon, i = 1, 2, 3, \dots, n$ 。据此我们修改 (3.3) 中的优化问题，

$$\begin{aligned} \min_{C_{ijk}} E(C_{111}, C_{112}, \dots, C_{N_x N_y N_z}) &:= \frac{1}{2} \sum_{r=1}^n [(\sum_{ijk} C_{ijk} B_{ijk}(v_r^1, v_r^2, v_r^3))^2 + (\epsilon - \sum_{ijk} C_{ijk} B_{ijk}(q_r^1, q_r^2, q_r^3))^2] \\ &\Leftrightarrow \sum_{r=1}^n (\sum_{ijk} C_{ijk} B_{ijk}(v_r^1, v_r^2, v_r^3) B_{uvw}(v_r^1, v_r^2, v_r^3) - (\epsilon - \sum_{ijk} C_{ijk} B_{ijk}(q_r^1, q_r^2, q_r^3)) B_{uvw}(q_r^1, q_r^2, q_r^3)) \end{aligned} \quad (3.4)$$

$$\mathbf{B} = \begin{bmatrix} B_{111}(v_1^1, v_1^2, v_1^3) & B_{112}(v_1^1, v_1^2, v_1^3) & \dots & B_{N_x N_y N_z}(v_1^1, v_1^2, v_1^3) \\ B_{111}(v_2^1, v_2^2, v_2^3) & B_{112}(v_2^1, v_2^2, v_2^3) & \dots & B_{N_x N_y N_z}(v_2^1, v_2^2, v_2^3) \\ \vdots & \vdots & & \vdots \\ B_{111}(v_r^1, v_r^2, v_r^3) & B_{112}(v_r^1, v_r^2, v_r^3) & \dots & B_{N_x N_y N_z}(v_r^1, v_r^2, v_r^3) \\ B_{111}(q_1^1, q_1^2, q_1^3) & B_{112}(q_1^1, q_1^2, q_1^3) & \dots & B_{N_x N_y N_z}(q_1^1, q_1^2, q_1^3) \\ B_{111}(q_2^1, q_2^2, q_2^3) & B_{112}(q_2^1, q_2^2, q_2^3) & \dots & B_{N_x N_y N_z}(q_2^1, q_2^2, q_2^3) \\ \vdots & \vdots & & \vdots \\ B_{111}(q_r^1, q_r^2, q_r^3) & B_{112}(q_r^1, q_r^2, q_r^3) & \dots & B_{N_x N_y N_z}(q_r^1, q_r^2, q_r^3) \end{bmatrix}$$

如果再记

$$\mathbf{b} = [0, \dots, 0, \epsilon, \dots, \epsilon]^T$$

其中 \mathbf{b} 的前 n 项为 0，后 n 项为 ϵ 。则 (3.4) 式可以写成紧凑的形式

$$\mathbf{B}^T (\mathbf{b} - \mathbf{B} \mathbf{C}) = 0 \quad (3.5)$$

求解该线性系统，我们使用迭代法求解，与 IPIA^[9] 不同，我们的迭代格式为

$$\mathbf{C}^{\alpha+1} = \mathbf{C}^{\alpha} + \Lambda \mathbf{B}^T (\mathbf{b} - \mathbf{B} \mathbf{C}^{\alpha}), \alpha = 0, 1, 2, \dots \quad (3.6)$$

这里 Λ 为

$$\begin{bmatrix} \frac{1}{\sum_{r=1}^n B_{111}(v_r) + B_{111}(q_r)} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sum_{r=1}^n B_{112}(v_r) + B_{112}(q_r)} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sum_{r=1}^n B_{NxNyNz}(v_r) + B_{NxNyNz}(q_r)} \end{bmatrix} \quad (3.7)$$

这里 $B_{ijk}(x) = B_{ijk}(x^1, x^2, x^3)$, 并假设 Λ 可逆 (即 $\forall ijk, \sum_{r=1}^n B_{ijk}(v_r) + B_{ijk}(q_r) \neq 0$), 否则认为有某个节点 ijk 满足 $\sum_{r=1}^n B_{ijk}(v_r) + B_{ijk}(q_r) = 0$, 由于 **B**-样条基函数非负, 则 $B_{ijk}(v_r) = B_{ijk}(q_r) = 0$, 我们可以去掉该节点, 同时去掉对应的 **B** 和 **C** 对应的行, 其并不影响点云附近的隐式曲面构建。所以我们不妨假定 Λ 可逆。有关收敛性的证明, 详见^[2]。本迭代格式相较于 **IPIA** 格式最大的优势就是内存占用少, 可高度并行, 我们从 **Algorithm ??** 来分析这一点。

我们可以发现 **Algorithm ??** 在第 4 行, 第 15 行, 第 22 行对应的循环可高度并行。同时需要的额外内存为储存在每个格点 \mathcal{G} 上的 C_{ijk} , λ_{ijk} , Δ_{ijk} 。而这一点, 我们将在下一节中说明里所需的额外内存可以借用物质点法中的数据结构完成, 也就是说 **LSIPIA** 的优势为可复用物质点法的数据结构, 同时可以高度并行。

图??(3) 为使用 **LSIPIA** 拟合点云后的结果, 可以发现拟合结果明显更光滑。图??更多的实验结果, 图中实验皆选自 Yusuf 等人^[2] 在 2019 年耗时较长的实验, 在原文中最短的兔子实验耗时为 88.24 秒, 其他三个模型均在 900 秒以上。其实验平台处理器为 Intel-core i7 @3.6GHZ, 内存 RAM 为 16GB。本文的实验平台为 Intel-Core i5 @1.6GHZ, 内存 RAM 为 8GB。具体性能如表??, 可以发现在以下四个实验中即使使用了更精细的空间网格以及较为逊色的运行平台, 相比于 **IPIA** 性能依然有三个数量级的提升, 且性能瓶颈主要集中在网格尺寸上。

表 3.1 **LSIPIA** 的性能

模型	数据点个数	背景网格尺寸	耗时 (单位: 秒)
bunny	35947	$128 \times 128 \times 128$	0.157785
budahha	80000	$256 \times 256 \times 256$	0.635985
armadillo	51892	$256 \times 256 \times 256$	0.690507
fertility	241607	$128 \times 128 \times 128$	0.554154

3.4.2 快速法向计算

在完成 **Algorithm ??** 之后, 我们获得了一个二阶连续的隐式曲面。即

$$f(x) = \sum_{ijk} C_{ijk} B_{ijk}(x) \quad (3.8)$$

Algorithm 2 LSIPIA**Input:** 点云位置 \mathcal{V} , 点云法向 \mathcal{N} , 背景网格格点集合 \mathcal{G} , 指定迭代次数 $LoopTimes$ **Output:** 隐式曲面 $f(x) = \sum_{ijk} C_{ijk} B_{ijk}(x)$, 即输出参数值 C_{ijk}

```

1: 初始化  $\forall ijk, C_{ijk} = 0$ 
2: 初始化  $\forall ijk, \lambda_{ijk} = 0$ 
3: 初始化偏移量  $\epsilon$ 
4: for  $r \in \#\mathcal{V}$  do
5:   预计算下标为  $I = \{ijk : v_r \in \text{supp}(B_{ijk})\}$  的张量积 B-样条函数, 储存为  $B_{ijk}(v_r)$ 
6:   for  $ijk \in I$  do
7:      $\lambda_{ijk} \leftarrow \lambda_{ijk} + B_{ijk}(v)$ 
8:   end for
9:    $v_{\epsilon r} \leftarrow v_r + \epsilon n_r$ 
10:  预计算下标为  $I_\epsilon = \{ijk : v_{\epsilon r} \in \text{supp}(B_{ijk})\}$  的张量积 B-样条函数, 储存为  $B_{ijk}(v_{\epsilon r})$ 
11:  for  $ijk \in I_\epsilon$  do
12:     $\lambda_{ijk} \leftarrow \lambda_{ijk} + B_{ijk}(v_{\epsilon r})$ 
13:  end for
14: end for
15: for  $ijk \in \mathcal{G}$  do
16:  if  $\lambda_{ijk} \neq 0$  then
17:     $\lambda_{ijk} \leftarrow \frac{1}{\lambda_{ijk}}$ 
18:  end if
19: end for
20: 初始化  $\forall ijk, \Delta_{ijk} = 0$ 
21: for  $i = 1, \dots, LoopTimes$  do
22:  for  $r \in \#\mathcal{V}$  do
23:    预计算下标为  $I = \{ijk : v_r \in \text{supp}(B_{ijk})\}$  的张量积 B-样条函数, 储存为  $B_{ijk}(v_r)$ 
24:     $res = 0$ 
25:    for  $ijk \in I$  do
26:       $res \leftarrow res + B_{ijk}(v_r)$ 
27:    end for
28:     $\delta_r = 0 - res$ 
29:    for  $ijk \in I$  do
30:       $\Delta_{ijk} \leftarrow \Delta_{ijk} + \delta_r \cdot B_{ijk}(v_r)$ 
31:      重置  $\Delta_{ijk}$  为 0
32:    end for
33:  end for
34: end for

```



图 3.7 LSIPIA 实验图，其中左上角模型为 bunny 模型实验，左上角第一张图为 bunny 点云输入，第二张为 LSIPIA 拟合结果。右上角，左下角，右下角分别为 budahha 模型，armadillo 模型，fertility 模型的实验结果

获取法向我们可以直接计算其梯度 ∇f ，我们首先分析计算一个点上的梯度所需要的代价。首先根据张量积 B-样条函数的定义 $B_{ijk}(x) = B_i(x^1) * B_j(x^2) * B_k(x^3)$ ，如果选取的为二次样条，则在二维情况下其能够覆盖到点 x 的基函数如图??，所以能够覆盖到点 x 的坐标个点为 $I = \{ij : i = i_0 + k, j = i_0 + l, \text{其中 } k, l \in \{0, 1, 2\}\}$ 。

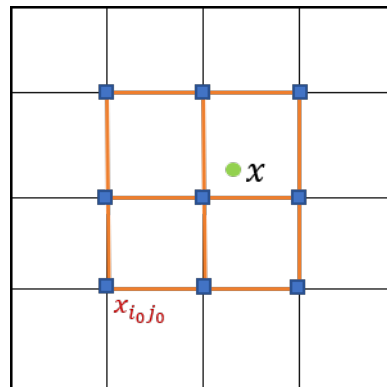


图 3.8 蓝色正方形为其紧支集能够覆盖 x 的自由度， $x_{i_0 j_0}$ 为左下角格点坐标

而

$$\begin{aligned}
\nabla f &= \sum_{ijk} C_{ijk} \nabla B_{ijk}(x) \\
&= \sum_{ijk} C_{ijk} \nabla (B_i(x^1) B_j(x^2) B_k(x^3)) \\
&= \sum_{ijk} C_{ijk} (\partial_i B_i(x^1) \cdot B_j(x^2) B_k(x^3), B_i(x^1) \partial_2 B_j(x^2) \cdot B_k(x^3), B_i(x^1) B_j(x^2) \partial_3 B_k(x^3))
\end{aligned} \tag{3.9}$$

可以发现要计算 x 点的梯度 ∇f , 我们需要计算的有 $B_i(x^r), \partial_r B_i(x^r), i \in \{0, 1, 2\}, r \in \{1, 2, 3\}$, 其中样条阶数越高, 计算 $\partial_r B_i(x^r)$ 代价越大, 并且 $\partial_r B_i(x^r)$ 为分段函数, 故在不同的 r 之间不可能做到单指令多数数据流并行, 同样在拼装 $\nabla B_{ijk}(x)$ 时每一个维度需要单独的计算。基于这些缺点, 我们提出另一个近似法向的计算方法。这里我们使用移动加权最小二乘来逼近格点系数, 我们将会看到在格点张量积 **B**-样条基下, 移动加权最小二乘将有惊人的简单形式。

我们寻求的近似目标函数为

$$f_y(x; c) = f(y) + c^T(x - y) \tag{3.10}$$

能量函数设计如下

$$E_y(c) = \frac{1}{2} \sum_{ijk} B_{ijk}(y) (f_y(x_{ijk}; c) - C_{ijk})^2 \tag{3.11}$$

对上式取极小我们得到

$$\begin{aligned}
0 &= \frac{\partial}{\partial c} E_y(c) \\
&= \frac{\partial}{\partial c} \frac{1}{2} \sum_{ijk} B_{ijk}(y) (f(y) + c^T(x_{ijk} - y) - C_{ijk})^2 \\
&= \sum_{ijk} B_{ijk}(y) (f(y) + c^T(x_{ijk} - y) - C_{ijk})(x_{ijk} - y)^T \\
&= \sum_{ijk} B_{ijk}(y) (f(y) - C_{ijk})(x_{ijk} - y)^T + c^T \sum_{ijk} B_{ijk}(y) (x_{ijk} - y)(x_{ijk} - y)^T
\end{aligned} \tag{3.12}$$

则

$$\sum_{ijk} B_{ijk}(y) (C_{ijk} - f(y))(x_{ijk} - y) = [\sum_{ijk} B_{ijk}(y) (x_{ijk} - y)(x_{ijk} - y)^T] c$$

当然我们要计算 c 我们得到如下式子

$$c = [\sum_{ijk} B_{ijk}(y) (x_{ijk} - y)(x_{ijk} - y)^T]^{-1} \cdot \sum_{ijk} B_{ijk}(y) (C_{ijk} - f(y))(x_{ijk} - y) \tag{3.13}$$

这里 (3.12) 式要成立的条件为 $[\sum_{ijk} B_{ijk}(y)(x_{ijk} - y)(x_{ijk} - y)^T]$ 可逆, 我们接下来证明该矩阵可逆。

此处我们记 $A = \sum_{ijk} B_{ijk}(y)(x_{ijk} - y)(x_{ijk} - y)^T$ 。首先我们计算 A 的非对角项即 $A_{uv}, u \neq v$ 其中 $u, v \in \{1, 2, 3\}$, 这里再记 $w \in \{1, 2, 3\}$ 且 $w \notin \{u, v\}$ 。即

$$\begin{aligned}
 A_{uv} &= \sum_{i_1 i_2 i_3} B_{i_1 i_2 i_3}(y)(x_{i_1 i_2 i_3}^u - y^u)(x_{i_1 i_2 i_3}^v - y^v) \\
 &= \sum_{i_u i_v i_w} B_{i_u}(y^u) B_{i_v}(y^v) B_{i_w}(y^w)(x_{i_u i_v i_w}^u - y^u)(x_{i_u i_v i_w}^v - y^v) \\
 &= \sum_{i_w} B_{i_w}(y^w) \left(\sum_{i_u} B_{i_u}(y^u)(x_{i_u i_v i_w}^u - y^u) \right) \left(\sum_{i_v} B_{i_v}(y^v)(x_{i_u i_v i_w}^v - y^v) \right) \\
 &= \sum_{i_w} B_{i_w}(y^w) \left(\sum_{i_u} B_{i_u}(y^u) x_{i_u i_v i_w}^u - y^u \sum_{i_u} B_{i_u}(y^u) \right) \left(\sum_{i_v} B_{i_v}(y^v) x_{i_u i_v i_w}^v - y^v \sum_{i_v} B_{i_v}(y^v) \right)
 \end{aligned} \tag{3.14}$$

这里我们注意到 **B**-样条基函数的单位分解性 $\sum_i B_i(x) = 1$ 以及线性还原性 $\sum_i x_i B_i(x) = x$, 带入到 (3.13) 式中我们有

$$\sum_{i_w} B_{i_w}(y^w) \left(\sum_{i_u} B_{i_u}(y^u) x_{i_u i_v i_w}^u - y^u \sum_{i_u} B_{i_u}(y^u) \right) \left(\sum_{i_v} B_{i_v}(y^v) x_{i_u i_v i_w}^v - y^v \sum_{i_v} B_{i_v}(y^v) \right) = 0$$

因此 $u \neq v$ 时 $A_{uv} = 0$, 即 A 为对角阵。

接下来考察 $u = v$ 时 A_{uv} 的值, 而

$$\begin{aligned}
 A_{uu} &= \sum_{i_1 i_2 i_3} B_{i_1 i_2 i_3}(y)(x_{i_1 i_2 i_3}^u - y^u)^2 \\
 &= \sum_{i_u i_v i_w} B_{i_u}(y^u) B_{i_v}(y^v) B_{i_w}(y^w)(x_{i_u i_v i_w}^u - y^u)^2 \\
 &= \sum_{i_u} B_{i_u}(y^u)(x_{i_u i_v i_w}^u - y^u)^2 \sum_{i_v} B_{i_v}(y^v) \sum_{i_w} B_{i_w}(y^w) \\
 &= \sum_{i_u} B_{i_u}(y^u)(x_{i_u i_v i_w}^u - y^u)^2
 \end{aligned} \tag{3.15}$$

我们将说明 $\sum_{i_u} B_{i_u}(y^u)(x_{i_u i_v i_w}^u - y^u)$ 恒为常数。这里我们注意到均匀 n 阶 **B**-样条导数公式为 $\frac{d}{dy} B_i^n(y) = B_i^{n-1}(y + \frac{1}{2}) - B_i^{n-1}(y - \frac{1}{2})$, 此处假定样条阶数为 n , 同时再记

$B_{i_u}(y^u)$ 为 $B_{i_u}^n(y^u)$, 现在考察 $\sum_{i_u} B_{i_u}(y^u)(x_{i_1 i_2 i_3}^u - y^u)$ 导数

$$\begin{aligned}
& \frac{d}{dy^u} \sum_{i_u} B_{i_u}^n(y^u)(x_{i_1 i_2 i_3}^u - y^u)^2 \\
&= \sum_{i_u} \frac{d}{dy^u} B_{i_u}^n(y^u)(x_{i_1 i_2 i_3}^u - y^u)^2 \\
&= \sum_{i_u} \left[\frac{d}{dy^u} B_{i_u}^n(y^u) \right] (x_{i_1 i_2 i_3}^u - y^u)^2 + B_{i_u}^n(y^u) \frac{d}{dy^u} [(x_{i_1 i_2 i_3}^u - y^u)^2] \\
&= \sum_{i_u} \left[\frac{d}{dy^u} B_{i_u}^n(y^u) \right] (x_{i_1 i_2 i_3}^u - y^u)^2 + 2B_{i_u}^n(y^u)(y^u - x_{i_1 i_2 i_3}^u) \\
&= \sum_{i_u} \left[\frac{d}{dy^u} B_{i_u}^n(y^u) \right] (x_{i_1 i_2 i_3}^u - y^u)^2 \\
&= \sum_{i_u} (B_{i_u}^{n-1}(y^u + \frac{h}{2}) - B_{i_u}^{n-1}(y^u - \frac{h}{2}))(x_{i_1 i_2 i_3}^u - y^u)^2 \\
&= \sum_{i_u} B_{i_u}^{n-1}(y^u + \frac{h}{2})(y^u - x_{i_1 i_2 i_3}^u)^2 - \sum_{i_u} B_{i_u}^{n-1}(y^u - \frac{h}{2})(y^u - x_{i_1 i_2 i_3}^u)^2 \\
&= \sum_{i_u} B_{i_u}^{n-1}(y^u + \frac{h}{2})(y^u + \frac{h}{2} - x_{i_1 i_2 i_3}^u - \frac{h}{2})^2 - \sum_{i_u} B_{i_u}^{n-1}(y^u - \frac{h}{2})(y^u - \frac{h}{2} - x_{i_1 i_2 i_3}^u + \frac{h}{2})^2 \\
&= \sum_{i_u} B_{i_u}^{n-1}(y^u + \frac{h}{2})[(y^u + \frac{h}{2} - x_{i_1 i_2 i_3}^u)^2 + \frac{h^2}{4} - h(y^u + \frac{h}{2} - x_{i_1 i_2 i_3}^u)] \\
&\quad - \sum_{i_u} B_{i_u}^{n-1}(y^u - \frac{h}{2})[(y^u - \frac{h}{2} - x_{i_1 i_2 i_3}^u)^2 + \frac{h^2}{4} + h(y^u - \frac{h}{2} - x_{i_1 i_2 i_3}^u)] \\
&= \{ [\sum_{i_u} B_{i_u}^{n-1}(y^u + \frac{h}{2})(y^u + \frac{h}{2} - x_{i_1 i_2 i_3}^u)^2] + \frac{h^2}{4} - h \sum_{i_u} B_{i_u}^{n-1}(y^u + \frac{h}{2})(y^u + \frac{h}{2} - x_{i_1 i_2 i_3}^u) \} \\
&\quad - \{ [\sum_{i_u} B_{i_u}^{n-1}(y^u - \frac{h}{2})(y^u - \frac{h}{2} - x_{i_1 i_2 i_3}^u)^2] + \frac{h^2}{4} - h \sum_{i_u} B_{i_u}^{n-1}(y^u - \frac{h}{2})(y^u - \frac{h}{2} - x_{i_1 i_2 i_3}^u) \} \\
&= \{ [\sum_{i_u} B_{i_u}^{n-1}(a)(a - x_{i_1 i_2 i_3}^u)^2] - h \sum_{i_u} B_{i_u}^{n-1}(a)(a - x_{i_1 i_2 i_3}^u) \} \\
&\quad - \{ [\sum_{i_u} B_{i_u}^{n-1}(a - h)(a - h - x_{i_1 i_2 i_3}^u)^2] - h \sum_{i_u} B_{i_u}^{n-1}(a - h)(a - h - x_{i_1 i_2 i_3}^u) \} \\
&= \sum_{i_u} B_{i_u}^{n-1}(a)(a - x_{i_1 i_2 i_3}^u)^2 - \sum_{i_u} B_{i_u}^{n-1}(a - h)(a - h - x_{i_1 i_2 i_3}^u)^2 \\
&= 0
\end{aligned} \tag{3.16}$$

从式 (3.15) 的推导中可以看出, 其导数恒为 0, 因此 $A_{uu} = \sum_{i_u} B_{i_u}^n(y^u)(x_{i_1 i_2 i_3}^u - y^u)^2$ 恒为常数。事实上, 在本文中, B-样条阶数为 2, 可以计算得到 $A_{uu} = \frac{h^2}{4}$, 因

此 $A = \frac{h^2}{4}I$ ，代入到 (3.12) 式中有如下

$$c = \frac{4}{h^2} \sum_{ijk} B_{ijk}(y)(C_{ijk} - f(y))(x_{ijk} - y) \quad (3.17)$$

$$f_y(x; c) = f(y) + \frac{4}{h^2} \sum_{ijk} B_{ijk}(y)(C_{ijk} - f(y))(x_{ijk} - y)^T(x - y) \quad (3.18)$$

继续观察 (3.18) 式，再利用 B-样条的线性还原性，可以得到

$$f_y(x; c) = f(y) + \frac{4}{h^2} \sum_{ijk} B_{ijk}(y)C_{ijk}(x_{ijk} - y)^T(x - y) \quad (3.19)$$

故近似法向选取为 $n = \frac{c}{\|c\|}$ 。可以发现，在计算 c 时，我们只要预计算 $B_i(y^u), i, u \in \{1, 2, 3\}$ ，这表明该算法有着更低的内存消耗，在组合法向时，同时向量值 $x_{ijk} - y$ 的计算远比 $(\partial_i B_i(x^1) \cdot B_j(x^2) B_k(x^3), B_i(x^1) \partial_2 B_j(x^2) \cdot B_k(x^3), B_i(x^1) B_j(x^2) \partial_3 B_k(x^3))$ 计算有更高的效率。

图??为二维算例的法向对比实验，可以看出使用拟合方法与直接梯度计算并没有明显的差别。

图??为更细致的比较。可以发现，在曲率比较大的位置，两种梯度计算有偏差，但是依然能捕捉到连续的法向，这对本身就是拟合点云得来的结果并无影响。

3.5 本章小结

本章给出了点云表面重建的方法，首先我们建立了点云的距离场，并使用合适的阈值 r 探查点云大致轮廓，并使用 **Marching cube** 算法来构造一阶连续曲面。为了得到更光滑的结果来获得更光滑的法向，本文使用 **LSIPIA** 方法处理，而实际上我们看到 **LSIPIA** 需要粒子上的法向作为输入，而实际上该法向仅仅是为了 **LSIPIA** 算法陷入平凡解，因此我们只需要估计出大致方向作为输入即可，其估计大致法向的办法将在下一节给出。在得到一个二阶光滑曲面之后，我们将基于此来计算法向，当然这个法向至少是一阶连续的，为了更少的计算代价，我们结合 B-样条基函数的特性，使用移动最小二乘方法给出了一个相比于直接计算梯度更高效的估计办法。

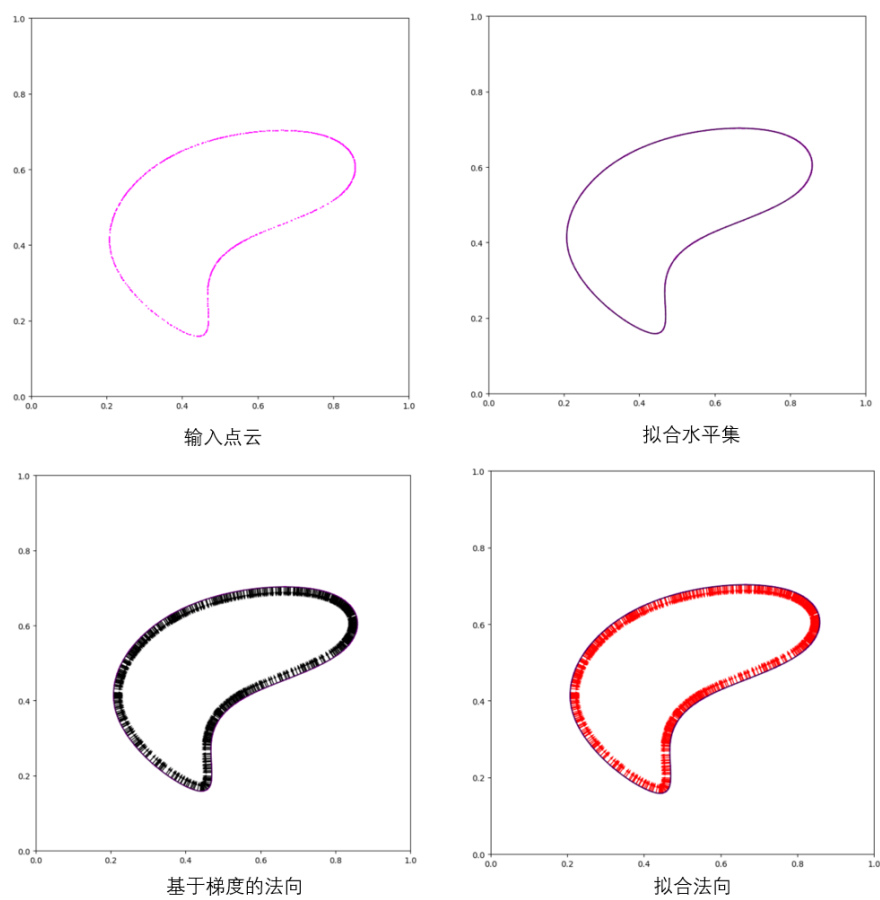


图 3.9 左上角为输入点云，右上角为拟合水平集，左下角为使用梯度直接计算法向，右下角为使用移动最小二乘拟合得来的法向

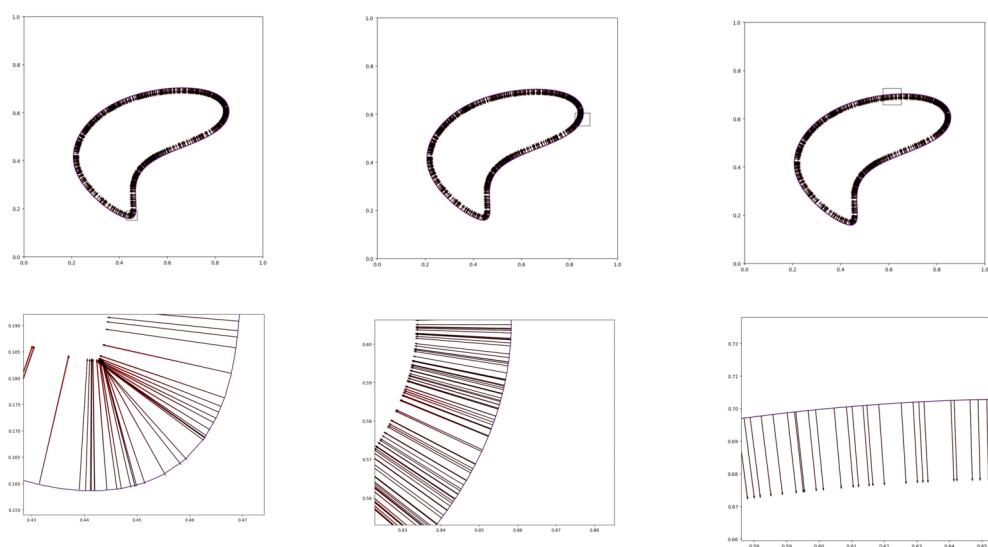


图 3.10 下方图表为上方图标选取的局部放大，红色箭头为拟合法向，黑色箭头为梯度计算的法向

4 数值方法

4.1 引言

在前文给出连续的控制方程，以及表面重建方法后，本章将离散化连续控制方程，以及针对流体表面张力给出一个物质的离散化方式，同时给出一个计算流水线将表面重建方法整合到物质点法中来完成我们的模拟。

4.2 离散控制方程

我们将使用伽辽金方法离散化控制方程，首先我们先将第一章中的动量强形式方程转换成弱形式，然后根据弱形式选取基函数以及对应的离散方法用于数值计算。此处我们回顾动量守恒方程的欧拉形式和拉格朗日形式，

$$\rho \frac{Dv}{Dt} = \text{div}(\sigma) + \rho g \quad \text{欧拉视角}$$

$$R(X, 0) \frac{\partial}{\partial t} V(X, t) = \text{DIV}(P) + Rg \quad \text{拉格朗日视角}$$

其中 $v(x, t) = V(X, t)$, $x = \phi(X, t)$ 。如果从拉格朗日视角来观察方程的左侧(时间变化项)，从欧拉视角来观察方程的右侧(空间变化项)，那么我们有如下混合欧拉拉格朗日形式的方程

$$R(X, 0) \frac{\partial}{\partial t} V(X, t) = \text{div}(\sigma) + \rho g \quad \text{混合欧拉拉格朗日视角}$$

4.2.1 时间离散化

在混合欧拉拉格朗日视角中，左端项为 $R(X, 0) \frac{\partial}{\partial t} V(X, t)$ ，相对于欧拉视角下 $\rho \frac{Dv}{Dt}$ 中的材料时间导数 $\frac{D}{Dt} = \frac{\partial}{\partial t} + v \cdot \nabla$ ，显然有更容易的离散方式，这也是时间变化项选择拉格朗日描述的原因。此处我们使用有限差分来离散速度有关时间的变化。

$$\frac{\partial}{\partial t} V(X, t) \approx \frac{V(X, t^{n+1}) - V(X, t^n)}{\Delta t}$$

同样的，在拉格朗日视角下的离散化可以转换为欧拉视角下的离散化，这里记 $x^n := \phi_{t^n}(X)$, $\hat{x}^{n+1} := \phi_{t^{n+1}, t^n}(x^n)$ ，同时 $v^n(x^n) := v(x^n, t^n)$, $\hat{v}^{n+1}(x^n) := v(\phi_{t^{n+1}, t^n}(x^n))$ ，因此以上差分格式可以写成

$$\frac{\partial}{\partial t} V(X, t) \approx \frac{\hat{v}^{n+1}(x^n) - v^n(x^n)}{\Delta t}$$

值得注意的是， $\hat{v}^{n+1}(x^n)$ 与 $v^n(x^n)$ 都是在 $\Omega^{t^n} = \phi_{t^n}(\Omega^0)$ 上定义的，因此该离散形式既可在欧拉视角下表示，同时也可以拉格朗日视角下表示。

4.2.2 弱形式

将左端形式带入欧拉视角中，我们有如下

$$\rho(x^n, t^n) \frac{\hat{v}^{n+1}(x^n) - v^n(x^n)}{\Delta t} = \operatorname{div}(\sigma) + \rho g$$

为了将表面张力融入方程中，我们补充边界条件如下

$$\begin{aligned} \rho(x^n, t^n) \frac{\hat{v}^{n+1}(x^n) - v^n(x^n)}{\Delta t} &= \operatorname{div}(\sigma) + \rho g \\ \sigma n &= t, x^n \in \partial\Omega^{t^n} \end{aligned}$$

上述方程以弱形式表示我们有，这里 w 选取为空间上具有紧支集的测试函数

$$\begin{aligned} \int_{\Omega^{t^n}} \rho(x, t^n) \frac{\hat{v}^{n+1}(x) - v^n(x)}{\Delta t} \cdot w dx &= \int_{\Omega^{t^n}} \operatorname{div}(\sigma) \cdot w dx + \int_{\Omega^{t^n}} \rho g \cdot w dx \\ \sigma n &= t, x \in \partial\Omega^{t^n} \end{aligned}$$

现在我们处理右端的散度项，由格林公式^[2]得

$$\begin{aligned} \int_{\Omega^{t^n}} \operatorname{div}(\sigma) \cdot w dx &= \int_{\partial\Omega^{t^n}} w \cdot (\sigma n) ds - \int_{\Omega^{t^n}} \sigma : \nabla w dx \\ &= \int_{\partial\Omega^{t^n}} w \cdot t ds - \int_{\Omega^{t^n}} \sigma : \nabla w dx \end{aligned}$$

我们先计算 $\int_{\partial\Omega^{t^n}} \sigma : \nabla w dx$ 部分。回顾流体弱可压缩模型获得的柯西应力

$$\sigma = \lambda(\det(F) - 1)I$$

带入可得 $\int_{\partial\Omega^{t^n}} \lambda(\det(F) - 1)I : \nabla w dx$ 。将上式整合我们得到

$$\int_{\Omega^{t^n}} \rho(x, t^n) \frac{\hat{v}^{n+1}(x) - v^n(x)}{\Delta t} \cdot w dx = \int_{\partial\Omega^{t^n}} w \cdot t ds - \int_{\Omega^{t^n}} \sigma : \nabla w dx$$

为了计算表面张力项，回顾我们在第一章得到的表面张力计算公式

$$S(\phi_t) = k \int_{\partial\Omega_{t^n}} \det(F(x; t, t^n)) \|F^{-T}(x; t, t^n) \tilde{n}\| d\tilde{s}$$

我们对其变分将获得流体表面上的表面张力场。为了方便，我们先记 $\tilde{F} = F(x; t, t^n)$ ，以及 $\Psi(\tilde{F}) = \det(\tilde{F}) \|\tilde{F}^{-T} \tilde{n}\|$ ，则

$$\begin{aligned} \delta\Psi(\tilde{F}) &= \delta(\det(\tilde{F}) \|\tilde{F}^{-T} \tilde{n}\|) \\ &= \delta[\det(\tilde{F})] \|\tilde{F}^{-T} \tilde{n}\| + \det(\tilde{F}) \delta\|\tilde{F}^{-T} \tilde{n}\| \\ &= \det(\tilde{F}) \tilde{F}^{-T} : \delta\tilde{F} \cdot \|\tilde{F}^{-T} \tilde{n}\| + \det(\tilde{F}) \delta\|\tilde{F}^{-T} \tilde{n}\| \end{aligned} \tag{4.1}$$

这里第三行我们使用了第二章提到的行列式求导结果 $\delta \det(F) = \det(F)F^{-T} : \delta F$, 接下来我们处理上述等式的右端项 $\delta \|\tilde{F}^{-T}\tilde{n}\|$. 这里注意到 $\|V\|^2 = V^T V$, 因此我们有 $2\|V\|\delta\|V\| = 2V^T \delta V$. 同时 $A^{-1}A = I$, 从这里我们由 $\delta A^{-1}A + A^{-1}\delta A = 0$, 整理得到 $\delta(A^{-1}) = -A^{-1}\delta A A^{-1}$. 那么我们从

$$\begin{aligned}\delta \|\tilde{F}^{-T}\tilde{n}\| &= \frac{\tilde{n}^T \delta(\tilde{F}^{-1})\tilde{F}^{-T}\tilde{n}}{\|\tilde{F}^{-T}\tilde{n}\|} \\ &= \frac{-\tilde{n}\tilde{F}^{-1}(\delta\tilde{F})\tilde{F}^{-1}\tilde{F}^{-T}\tilde{n}}{\|\tilde{F}^{-T}\tilde{n}\|}\end{aligned}$$

将上式带入 (4.1) 式得

$$\delta\Psi(\tilde{F}) = \det(\tilde{F})\tilde{F}^{-T} : \delta\tilde{F} \cdot \|\tilde{F}^{-T}\tilde{n}\| - \frac{\tilde{n}^T \tilde{F}^{-1}(\delta\tilde{F})\tilde{F}^{-1}\tilde{F}^{-T}\tilde{n}}{\|\tilde{F}^{-T}\tilde{n}\|}$$

则

$$\begin{aligned}\delta\Psi(F(x_0; t_0, t_0)) &= \delta\Psi(I) \\ &= I : \delta\tilde{F} - \tilde{n}^T \delta\tilde{F}\tilde{n} \\ &= I : \delta\tilde{F} - \tilde{n}\tilde{n}^T : \delta\tilde{F} \\ &= (I - \tilde{n}\tilde{n}^T) : \delta\tilde{F}\end{aligned}$$

而表面能量变化等于表面张力沿位移做的功, 这里 $\delta\phi_{t^n}(X) = w \circ \phi_{t^n}(X)$, 即

$$\delta S(\phi_{t^n}) = \int_{\partial\Omega^{t^n}} w \cdot t ds$$

因此我们有

$$\begin{aligned}\int_{\partial\Omega^{t^n}} w \cdot t ds &= \delta S(\phi_{t^n}) \\ &= k \int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) : \delta\tilde{F} ds \\ &= k \int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) : \nabla w ds\end{aligned}$$

因此我们可以得到

$$\begin{aligned}\int_{\Omega^{t^n}} \rho(x, t^n) \frac{\hat{v}^{n+1}(x) - v^n(x)}{\Delta t} \cdot w dx &= k \int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) : \nabla w ds \\ &\quad - \int_{\Omega^{t^n}} \lambda(\det(F \circ \phi_t^{-1}) - 1) I : \nabla w dx \quad (4.2)\end{aligned}$$

4.2.3 空间离散化

在得到 (4.2) 式后我们将积分离散化, 首先我们假设 Ω^{t^n} 被离散为点云 \mathcal{V} , 如图??所示。

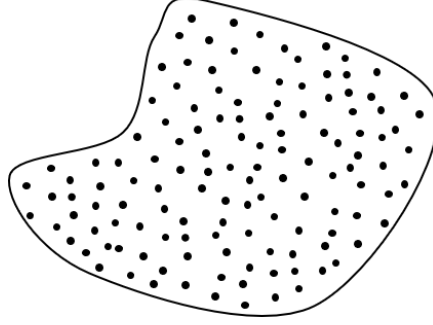


图 4.1 连续体离散化成点云

其次假设有限元函数空间为

$$\{f : f(x) = \sum_{\mathbf{i}} f_{\mathbf{i}} B_{\mathbf{i}}(x), \mathbf{i} = ijk, i, j, k \in 1, 2, 3\}$$

因此 $\hat{v}^{n+1}(x) = \sum_{\mathbf{i}} \hat{v}_{\mathbf{i}}^{n+1} B_{\mathbf{i}}(x)$, $v^n(x) = \sum_{\mathbf{i}} v_{\mathbf{i}}^n B_{\mathbf{i}}(x)$, 测试函数 $w(x) = e_{\alpha} B_{\mathbf{i}}(x)$, 这里 e_{α} 为三维空间的第 $\alpha \in \{1, 2, 3\}$ 个标准基。带入 (4.2) 式左边有

$$\int_{\Omega^{t^n}} \rho(x, t^n) \frac{\hat{v}^{n+1}(x) - v^n(x)}{\Delta t} \cdot w dx = \int_{\Omega^{t^n}} \rho(x, t^n) \sum_{\mathbf{i}} \frac{\hat{v}_{\mathbf{i}}^{n+1} - v_{\mathbf{i}}^n}{\Delta t} B_{\mathbf{i}}(x) \cdot e_{\alpha} B_{\mathbf{j}}(x) dx$$

右边为

$$\int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) : e_{\alpha} \nabla B_{\mathbf{j}}(x) ds - \int_{\Omega^{t^n}} \lambda(\det(F \circ \phi_t^{-1}) - 1) I : e_{\alpha} \nabla B_{\mathbf{j}}(x) dx \quad (4.3)$$

如果我们同时考虑 $e_{\alpha}, \alpha = 1, 2, 3$, 这样上式即在三个维度统一表示为

$$\begin{aligned} \int_{\Omega^{t^n}} \rho(x, t^n) \sum_{\mathbf{i}} \frac{\hat{v}_{\mathbf{i}}^{n+1} - v_{\mathbf{i}}^n}{\Delta t} B_{\mathbf{i}}(x) B_{\mathbf{j}}(x) dx &= \int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) \nabla B_{\mathbf{i}}(x)^T ds \\ &\quad - \int_{\Omega^{t^n}} \lambda(\det(F \circ \phi_t^{-1}) - 1) \nabla B_{\mathbf{i}}(x)^T dx \end{aligned}$$

对积分离散化我们有

$$\begin{aligned} \sum_{\mathbf{i}} \int_{\Omega^{t^n}} \rho(x, t^n) \frac{\hat{v}_{\mathbf{i}}^{n+1} - v_{\mathbf{i}}^n}{\Delta t} B_{\mathbf{i}}(x) B_{\mathbf{j}}(x) dx &\approx \sum_{\mathbf{i}, p} B_{\mathbf{i}}(x_p^n) B_{\mathbf{j}}(x_p^n) \frac{\hat{v}_{\mathbf{i}}^{n+1} - v_{\mathbf{i}}^n}{\Delta t} \int_{\Omega_p^{t^n}} \rho(x, t) dx \\ &= \sum_{\mathbf{i}, p} B_{\mathbf{i}}(x_p^n) B_{\mathbf{j}}(x_p^n) \frac{\hat{v}_{\mathbf{i}}^{n+1} - v_{\mathbf{i}}^n}{\Delta t} m_p \end{aligned} \quad (4.4)$$

此处 $p \in \mathcal{V}$ 为离散点云上的点, $x_p^n \in \mathbb{R}^3$ 为 t^n 时刻点 p 的位置, $m_p = \int_{\Omega_p^{t^n}} \rho(x, t) dx$, 这里 m_p 为粒子 p 周围的质量, 根据质量守恒, m_p 与时间无关. 再记 $\hat{M}_{\mathbf{j}\mathbf{i}} = \sum_p B_{\mathbf{j}}(x_p^n) B_{\mathbf{i}}(x_p^n) m_p$. 通常我们会牺牲精度, 使用矩阵行求和的方式^[21]来简化矩阵 \hat{M} . 即 $M_{\mathbf{j}\mathbf{i}} = \sum_{\mathbf{i}} \hat{M}_{\mathbf{j}\mathbf{i}}$, M 的非对角项为 0. 同时注意到, \mathbf{B} -样条的单位分解性, 我们有

$$M_{\mathbf{j}\mathbf{i}} = \sum_p m_p B_{\mathbf{j}}(x_p^n) \sum_{\mathbf{i}} B_{\mathbf{i}}(x_p^n) = \sum_p m_p B_{\mathbf{j}}(x_p^n)$$

同理我们可以得到

$$\begin{aligned} \int_{\Omega^{t^n}} \lambda(\det(F \circ \phi_t^{-1}) - 1) \nabla B_{\mathbf{j}}(x)^T dx &\approx \sum_p \lambda(\det(F_p^n) - 1) \nabla B_{\mathbf{j}}(x_p^n)^T \int_{\Omega_p^{t^n}} dx \\ &= \sum_p \lambda(\det(F_p^n) - 1) \nabla B_{\mathbf{j}}(x_p^n)^T V_p^n \\ &\approx \sum_p \lambda(\det(F_p^n) - 1) \nabla B_{\mathbf{j}}(x_p^n)^T V_p^0 \end{aligned}$$

这里 F_p^n 为粒子 p 在 t^n 时刻的形变梯度, V_p^n 为时刻 t^n 时 p 邻域小块 $\Omega_p^{t^n}$ 的体积, 由于我们将使用较大的体积惩罚来减小体积变化, 因此我们可以近似 V_p^n 为初始时刻的体积 V_p^0 .

最后我们来处理表面积分项 $\int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) \nabla B_{\mathbf{i}}(x)^T ds$, 假定我们已经获取了点云在 t^n 时刻的表面, 总表面积为 $Area$, 且已在表面上均匀的采点, 每个点分配面积为 $A_s = \frac{Area}{\#\mathcal{S}}$, 表面点集为 \mathcal{S} .

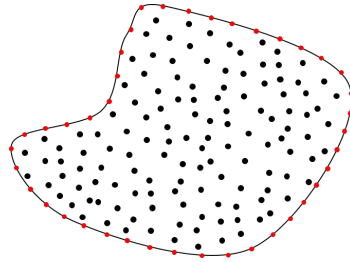


图 4.2 红色点 s 为采样获得的表面点集 \mathcal{S}

那么我们有

$$\int_{\partial\Omega^{t^n}} (I - \tilde{n}\tilde{n}^T) \nabla B_{\mathbf{j}}(x)^T ds \approx \sum_{s \in \mathcal{S}} (I - \tilde{n}_s \tilde{n}_s^T) \nabla B_{\mathbf{j}}(x_s^n)^T A_s \quad (4.5)$$

将上述离散化结果整理我们得到

$$\begin{aligned} \sum_{p \in \mathcal{V}} \frac{\hat{v}_{\mathbf{j}}^{n+1} - v_{\mathbf{j}}^n}{\Delta t} m_p B_{\mathbf{j}}(x_p^n) &= k \sum_{s \in \mathcal{S}} (I - \tilde{n}_s \tilde{n}_s^T) \nabla B_{\mathbf{j}}(x_s^n)^T A_s \\ &\quad - \lambda \sum_{p \in \mathcal{V}} (\det(F_p^n) - 1) \nabla B_{\mathbf{j}}(x_p^n)^T V_p^0 \end{aligned} \quad (4.6)$$

如果记 $m_j^n = \sum_p m_p B_j(x_p^n)$, 则上述问题变为

$$m_j^n \frac{\hat{v}_j^{n+1} - v_j^n}{\Delta t} = k \sum_{s \in \mathcal{S}} (I - \tilde{n}_s \tilde{n}_s^T) \nabla B_j(x_s^n)^T A_s - \lambda \sum_{p \in \mathcal{V}} (\det(F_p^n) - 1) \nabla B_j(x_p^n)^T \quad (4.7)$$

自然我们给出 p 的位置 x_p^n 更新公式

$$x_p^{n+1} = x_p^n + \Delta t \tilde{v}_p^{n+1} \quad (4.8)$$

同时回顾第二章的 $\dot{F} = \nabla v \cdot F$, 我们得到

$$F_p^{n+1} = (I + \Delta t \cdot \nabla v_p^{n+1}) F_p^n \quad (4.9)$$

4.2.4 数据表示与储存

从式 (4.7), (4.8), (4.9) 可以看到, 我们需要的变量有 $x_p^n \in \mathbb{R}^3$, $v_p^n \in \mathbb{R}^3$, $F_p^n \in \mathbb{R}^{3 \times 3}$, 这些变量我们都储存在粒子上, 由于粒子上储存质量信息不会随着时间改变, 因此该数据表示方式自然的满足了质量守恒方程。如图??所示

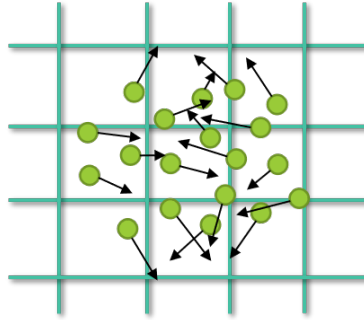


图 4.3 粒子上的信息

同时注意到这里在 t^n 时刻需要参与计算的信息还有 $v_j^n \in \mathbb{R}^3$ 和 $m_j^n \in \mathbb{R}$, 这里的信息我们储存在格点上。如图??所示, 这里我们注意到 $m_j^n = \sum_p m_p B_j(x_p^n)$, 此处实际上我们把粒子上的数据分散到了网格上, 同时注意到 $\sum_j m_j^n = \sum_j \sum_p m_p B_j(x_p^n)$, 这里再使用 B-样条的单位分解性, 我们有 $\sum_j m_j = \sum_p m_p$, 因此该分配方式依然是质量守恒的过程。

4.3 数值修正

在上一节得到连续方程的离散化之后, 我们接下来给出该离散方程的计算策略以及一些工程实践上的修正。

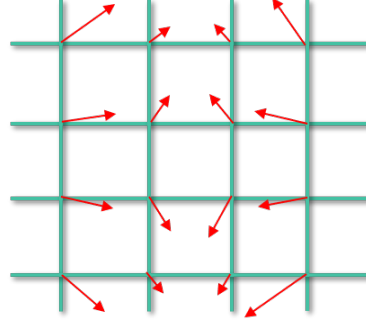
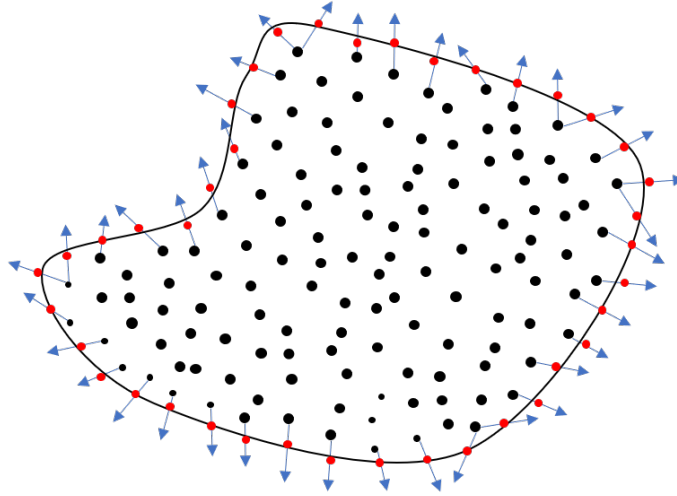


图 4.4 格点上的信息

4.3.1 表面重建与粒子属性调整

在给定的粒子集合 \mathcal{V} 以及 t^n 时刻每个粒子 $p \in \mathcal{V}$ 的位置 x_p 时，我们可以使用第??章提到的方法重建表面网格，然后使用泊松圆盘采样获取表面的粒子 \mathcal{S} 。在构造隐式曲面时，我们需要额外的偏移量，此处我们对每一个 $s \in \mathcal{S}$ ，在 \mathcal{V} 中寻找距离 s 最近的粒子 $p_s \in \mathcal{V}$ ，偏移方向为 $n_s = \frac{x_s - x_{p_s}}{\|x_s - x_{p_s}\|}$ ，如图??所示。我们将 \mathcal{S} 和 \mathcal{N} 作为输入可以得到逼近的隐式曲面，然后估计 \mathcal{S} 中表面粒子的法向 \tilde{n}_s 。

图 4.5 箭头为偏移方向，红色点为表面粒子 $s \in \mathcal{S}$ ，箭头起始端为 $p_s \in \mathcal{V}$

在实践中，表面张力项 $f^{surface} = k \sum_{s \in \mathcal{S}} (I - \tilde{n}_s \tilde{n}_s^T) \nabla B_j(x_s^n)^T A_s$ 需要传导到内部粒子，与液体压力共同影响形变结果，如果表面粒子离内部粒子最短距离大于 $\sqrt{3}h$ (即在 s 所占的一个格子之外)，那么该表面粒子 $s \in \mathcal{S}$ 将难以通过网格把表面张力传导到内部粒子。为了解决这一问题，我们对每一个表面粒子 $s \in \mathcal{S}$ 寻找其在内部的最近的点 $p_s \in \mathcal{V}$ ，然后生成一个嵌入点 o_s ，嵌入点 o_s 的位置我们选取为 $x_{o_s} = 2x_{p_s} - x_s$ ，嵌入点集合我们记为 \mathcal{O} 。显然， \mathcal{O} 与 \mathcal{S} 有一一对应关系，而

内部点 $p \in \mathcal{V}$ 与表面点集可能是多对一，因此我们记录每一个内部点 v 所相关的表面点个数 C_v ，同时我们将内部点的质量均匀的分布给表面粒子和嵌入粒子，即 $m_s = m_{o_s} = \frac{m_{p_s}}{2C_{p_s}+1}$ ，其中内部粒子 p_s 的质量 m_{p_s} 重新赋值为 m_s 。同样的，我们赋予当前时刻表面粒子速度 $v_s^n = v_{p_s}^n$ 以及嵌入粒子 o_s 的速度 $v_{o_s}^n = v_{p_s}^n$ 。

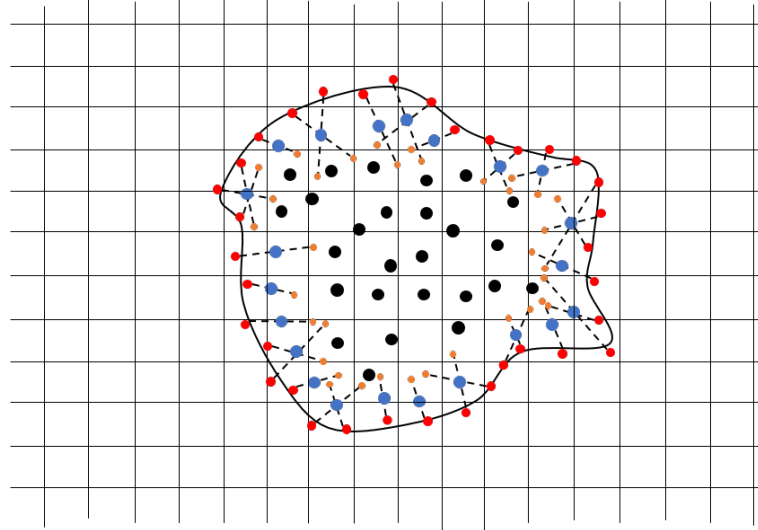


图 4.6 红色点为表面粒子 $s \in \mathcal{S}$ ，红色点通过虚线所连的蓝色点为离 s 最近的内部点 $p_s \in \mathcal{V}$ ，橙色点 $o_s \in \mathcal{O}$ 为新嵌入内部点

4.3.2 粒子网格信息转换

实际上在上一节中，我们已经给出了把粒子上的质量信息传递到网格上的计算公式，即 $\tilde{m}_j^n = \sum_p \tilde{m}_p B_j(x_p^n)$ 。除了质量信息之外，我们仍然需要速度信息 v_j^n ，基于动量守恒的考虑^[7]，我们采用的传递方式为将粒子的动量信息传递到网格上，再通过动量和质量的关系获取网格上的速度，即

$$(mv)_j^n = \sum_p \tilde{m}_p v_p^n B_j(x_p^n) \quad (4.10)$$

$$v_j^n = \frac{(mv)_j^n}{m_j^n} \quad (4.11)$$

和质量守恒的证明方式类似，我们也可以证明该过程是动量守恒的，详细证明见^[7]。其中质量和动量的传递到网格过程我们称之为粒子转网格 (Particle to Grid, 简记为 P2G)。

为了将网格速度信息转换到内部粒子 (Grid to Particle, 简记为 G2P) 上，最直观的办法是在粒子上采样空间网格上的速度场，即

$$v_p^n = \sum_i v_i^n B_i(x_p^n) \quad (4.12)$$

然而实际上,这样的采样办法只取得了速度的零阶信息,在实践中表现出来过粘的结果^[2],其中一种解决方案为保存空间速度场一阶速度信息,该方式被称为 APIC^[2]。即对每个粒子额外添加 $A_p^n \in \mathbb{R}^{3 \times 3}$, 一般而言 A_p^n 选取为速度场在 x_p^n 处的梯度, 在 APIC 方法中选择为梯度的近似量。

$$A_p^n = \sum_{\mathbf{i}} \frac{4}{h^2} B_{\mathbf{i}}(x_p^n) v_{\mathbf{i}}^n (x_{\mathbf{i}} - x_p^n)^T \quad (4.13)$$

对于接近表面的内部粒子 $\mathcal{V}_S = \{p_s \in \mathcal{V} : p_s = \arg \min_{p \in \mathcal{V}} \|p - s\|, s \in \mathcal{S}\}$, 每一个 $p \in \mathcal{V}_S$ 记 $\mathcal{N}_p = \{s \in \mathcal{S} : s = \arg \min_{s \in \mathcal{S}} \|p - s\|\} \cup \{o_s \in \mathcal{O} : s = \arg \min_{s \in \mathcal{S}} \|p - s\|\}$ 为其相关的表面粒子集合。这里 $p_s \in \mathcal{V}_S$ 的近似梯度计算方式修正为

$$A_{p_s}^n = \frac{1}{2C_{p_s} + 1} \sum_{q \in \mathcal{N}_{p_s} \cup \{p_s\}} \sum_{\mathbf{i}} B_{\mathbf{i}}(x_q^n) \frac{4}{h^2} v_{\mathbf{i}}^n (x_{\mathbf{i}} - x_{p_s}^n)^T \quad (4.14)$$

在网格转粒子的步骤中添加了梯度信息的传递之后, 我们修正 P2G 的过程。在每一个粒子上, 我们使用仿射函数来近似粒子所携带的速度 $v_p^n(x) = v_p^n + A_p^n(x - x_p^n)$, 这里动量传递据此修改为

$$(mv)_{\mathbf{j}}^n = \sum_p m_p v_p^n(x_{\mathbf{j}}) B_{\mathbf{j}}(x_p^n) \quad (4.15)$$

同样的, 表面粒子 $s \in \mathcal{S}$ 与对应的嵌入粒子 $o_s \in \mathcal{O}$ 也赋予对应的梯度信息, $A_s = A_{o_s} = A_{p_s}$, 这里 p_s 为 \mathcal{V} 中距离 s 最近的内部粒子, 同样使用 (4.15) 式来传递到网格上。

在计算过程中, 我们使用了 A_p^n 在近似替代 t^n 时刻速度场在位置 x_p^n 处的梯度, 因此我们在更新形变梯度 F_p^n 时, 为了节省计算量, 我们也直接将 ∇v_p^n 替代为 A_p^n , 即

$$F_p^{n+1} = (I + A_p^{n+1}) F_p^n \quad (4.16)$$

4.3.3 网格上力的计算

网格上由弱可压缩模型带来的力如下

$$f_{\mathbf{j}}^{liquid} = -\lambda \sum_p (det(F_p^n) - 1) \nabla B_{\mathbf{j}}^T(x_p^n) \quad (4.17)$$

由表面张力带来的效果为

$$f_{\mathbf{j}}^{surface} = k \sum_{s \in \mathcal{S}} (I - \tilde{n}_s \tilde{n}_s^T) \nabla B_{\mathbf{j}}(x_s^n)^T A_s \quad (4.18)$$

因此网格上的速度场更新为

$$\hat{v}_j^{n+1} = \frac{1}{m_j^n} (m_j^n v_j^n + \Delta t f_j^{liquid} + \Delta t f_j^{surface} + \Delta t m_j^n g) \quad (4.19)$$

此处 g 为重力加速度。

4.4 计算流程

在前文给出了每一个步骤的计算细节之后，我们在本节总结算法的计算流程。首先我们的输入为粒子集合 \mathcal{V} 及每个粒子 $p \in \mathcal{V}$ 的预置位置 x_p^0 以及质量 m_p 。默认初始化速度 $v_p^0 = 0$ ，形变梯度与速度梯度为单位阵 I 。之后根据粒子的位置 x_p^n 激活相邻的空间网格，并在激活网格上初始化 m_i 与 v_i 的内存，同时生成 **Marching cube** 所需的距离场。接下来，在激活的稀疏网格使用 **Marching Cube** 算法提取距离场的水平集网格，并使用泊松圆盘采样在网格面上均匀采取表面粒子 \mathcal{S} 。然后我们估计表面粒子 $s \in \mathcal{S}$ 的偏移量以及生成嵌入粒子 \mathcal{O} ，该过程用到的临近粒子搜索依然可以利用稀疏网格 \mathcal{G} 做空间哈希。在有了偏移量之后，将偏移量和粒子位置作为输入，利用 **LSPIA** 拟合隐式曲面，并获得连续的法向。在这之后进行 **P2G**, **GOP**, **G2P** 三个步骤并更新粒子状态获得 $x_p^{n+1}, v_p^{n+1}, F_p^{n+1}, A_p^{n+1}$ ，并作为下一帧的输入。同时我们输出当前时刻的 x_p^{n+1} 用作渲染。流程图见??。

4.5 实验结果

为了验证提出的表面张力计算方法，以下给出四个具体的数值实验。以下实验表面粒子采样皆为 40000，空间网格 $\frac{1}{256}$ ，时间步长为 $\Delta t = 1e^{-5}$ ，为了保证足够的抗压缩性弱可压缩系数 λ 选取为 60000。以下实验底层代码均为 C++，使用 python 调用，实验平台 CPU 为 i7-9700@3.00GHz，内存为 16GB。

4.5.1 水滴自由下落

该实验表面张力系数 $k = 8.0$ ，球体由 80000 粒子表示。该实验总耗用内存约为 14GB，每个时间步长平均为 0.7487s，每一帧由 50 个时间步长组成，平均一帧耗时为 37 秒。水滴由初速度 0.1，初始高度 0.02，自由下落，在触及光滑表面后形变并回弹，具体实验场景见图??。

4.5.2 拓扑变化

该实验表面张力系数 $k = 2.0$ ，球体由 60000 粒子表示。该实验耗用内存约为 12GB，每个时间步长平均为 0.8523s，每一帧由 150 个时间步长组成，平均一帧耗时为 122 秒。水滴初速度为 0.9，初始高度为 0.2，自由下落。可以发现在触及光滑表面后出现拓扑变化，然后再表面张力的影响下慢慢恢复，具体实验场景见??。

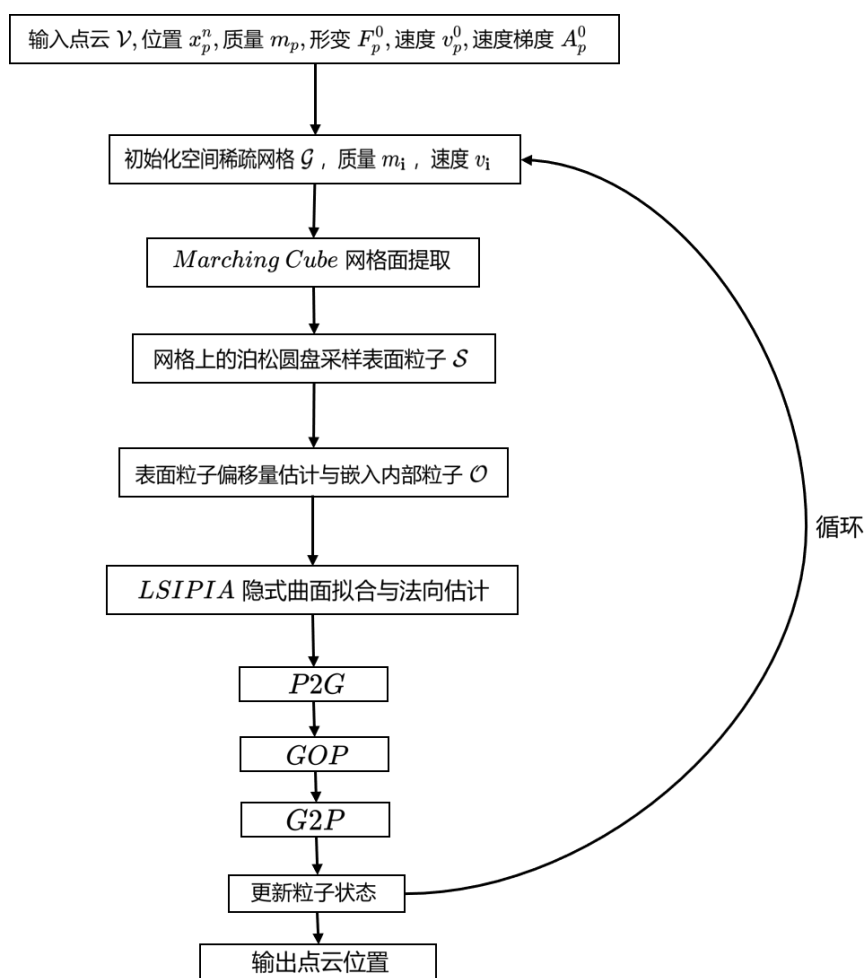


图 4.7 计算流水线

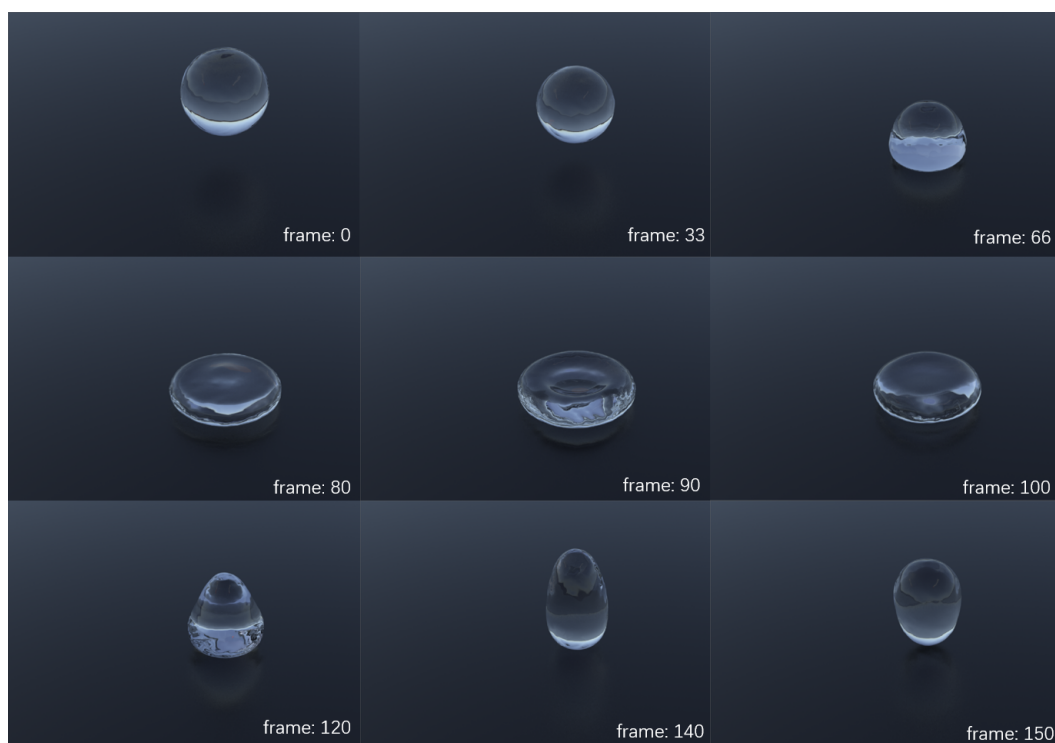


图 4.8 水滴下落实验

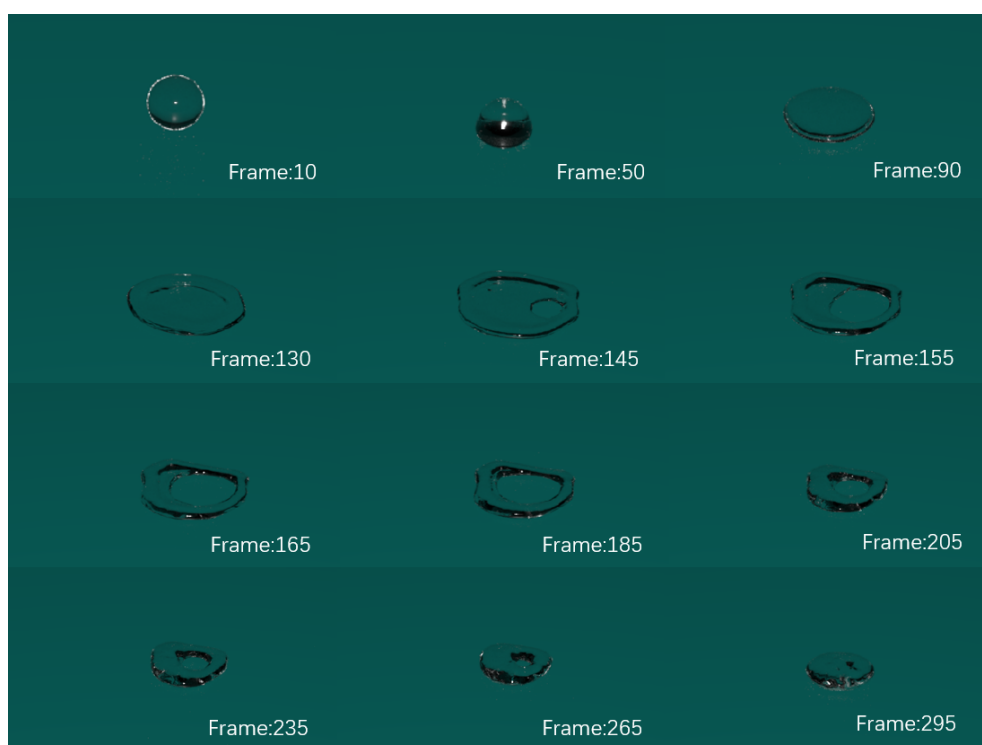


图 4.9 水滴拓扑变化

5 总结与展望

5.1 本文工作总结

本文基于连续介质力学对流体表面张力与流体的抗压缩性进行建模，使用动量守恒以及本构模型给出了对应的偏微分方程。并且针对性的对物质点法给出了表面张力的连续形式，并在伽辽金方法框架下将表面张力从边界条件转换到方程中，使其在物质点法的框架下易于离散。

同时本文利用了隐式曲面来拟合粒子表示的流体表面，相比于传统的网格方法可以更方便的实现拓扑变化。同时隐式曲面的拟合算法大量复用了物质点法本身所需的数据结构，并通过改进迭代方式使其易于并行，相比原始的构建隐式曲面方法达到四个数量级的加速。同样本文给出了一个配套的法向估计算法，在实践中与直接计算隐式曲面法向没有明显的差别，同时大大简化了计算流程，并让计算过程易于实现单指令多数据流并行。

最后，本文将隐式曲面与物质点法的计算流程结合，给出了相应的偏微分方程离散化格式以及新的计算流程，并使用嵌入内部粒子的方式，提高了辛积分格式在表面张力计算中的稳定性。为了验证本文给出的方法，我们给出了几个基本的数值实验，在几个测试集中本文方法成功的捕捉到了流体表面张力以及抗压缩的特征。

5.2 未来工作展望

本文将表面张力融入了物质点法中，并针对物质点法改进表面重建的方法，使得其在速度和内存消耗上都有着很大的提升，但是该方法还有着很大的改进空间以及进一步探索的空间。

1、进一步提高辛格式方法的稳定性，尽管本文能够求解一些例子，但是由于物质点法本身对时间步长要求较严格，导致某些算例所需要的时间步长特别小，最终影响整体计算性能。

2、为了方便实现，本文对流体抗压缩性建模使用的为弱可压缩能量，最终在模拟上表现出一些不真实性。如何高效的将不可压缩限制条件融入到求解流程，也是一个值得探索的方向。

3、物质点法本身在耦合多种物质求解取得了巨大的成功，在添加了表面张力之后，我们可以考虑添加流体更多的属性，以及探索如何模拟流体和不同表面接触的效果。同时将流体蒸发现象考虑其中也是一件有趣的事情。

作者简历

张某某，男，19XX 年，汉族，浙江杭州人。20XX 年考入浙江大学数学科学学院（信息与计算科学专业），20XX 年本科毕业，获得理学学士学位。20XX 年进入浙江大学数学科学学院计算数学专业研究生学习至今。

1. 工作经历

- 20XX-20XX 年，在 XX 公司 XX 部门 XX 岗位
- 20XX-20XX 年，在 XX 公司 XX 部门 XX 岗位

2. 参与的项目

- 20XX-20XX 年，参与 XXXX 项目
- 20XX-20XX 年，负责 XXXX 项目

3. 攻读学位期间发表的论文

- 猪八戒, 猪悟能, 天蓬元帅, 等. 论流体食物的持久保存 [D]. 硕士学位论文. 北京: 广寒宫大学, 2005