



哈尔滨工程大学
HARBIN ENGINEERING UNIVERSITY

基于`SPH`的舰船波浪砰击力学仿真

直接数值模拟和粒子搜索

汇报人：左志华 (zuo.zhihua@qq.com)

指导老师：廖康平

哈尔滨工程大学 船舶工程学院

目录

CONTENTS

01

直接数值模拟

02

粒子搜索

03

粒子搜索实例：四叉树

04

不完全回答，两个问题

05

总结



哈尔滨工程大学
HARBIN ENGINEERING UNIVERSITY

1 直接数值模拟

网格方法：

1. 直接数值模拟 (DNS)：要求追踪小涡旋，网格数大，计算量、储存量大。
2. 湍流模型：先验的湍流模式，减轻求解NS方程的负担。

SPH方法（粒子方法）：

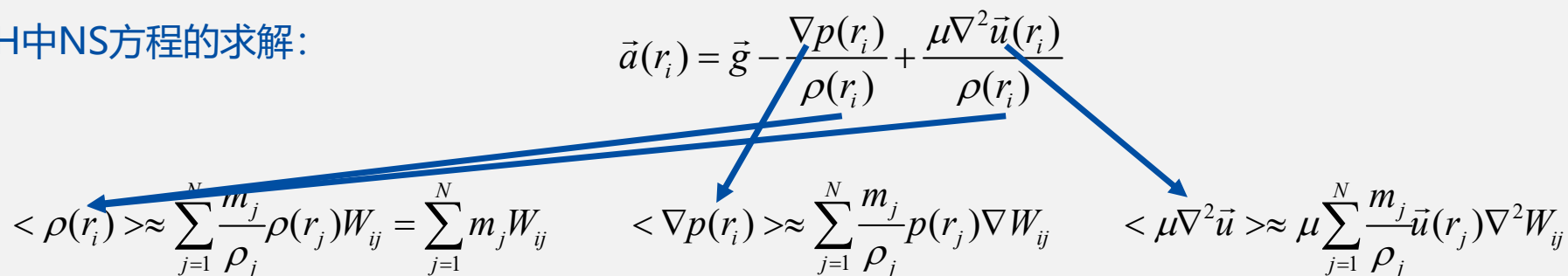
1. 直接数值模拟 (DNS)：被广泛描述，也对粒子数需求大，与网格方法（假设与描述）又有不同。
2. 湍流模型：先验的湍流模式，减轻求解NS方程的负担。（仅少量文献^[1]进行讨论）

SPH中NS方程的求解：

$$\vec{a}(r_i) = \vec{g} - \frac{\nabla p(r_i)}{\rho(r_i)} + \frac{\mu \nabla^2 \vec{u}(r_i)}{\rho(r_i)}$$

$$\langle \rho(r_i) \rangle \approx \sum_{j=1}^N \frac{m_j}{\rho_j} \rho(r_j) W_{ij} = \sum_{j=1}^N m_j W_{ij}$$

$$\langle \nabla p(r_i) \rangle \approx \sum_{j=1}^N \frac{m_j}{\rho_j} p(r_j) \nabla W_{ij}$$

$$\langle \mu \nabla^2 \vec{u} \rangle \approx \mu \sum_{j=1}^N \frac{m_j}{\rho_j} \vec{u}(r_j) \nabla^2 W_{ij}$$


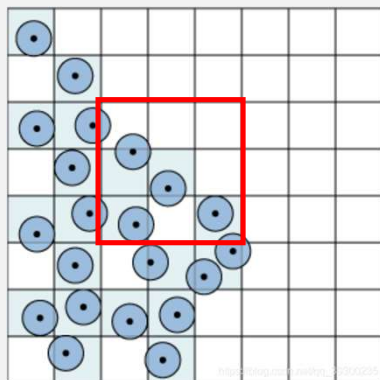
事物发展规律角度：较为成熟、健壮的DNS实现（探索性） → 可能可行的先验模型方案加速计算。

2 粒子搜索

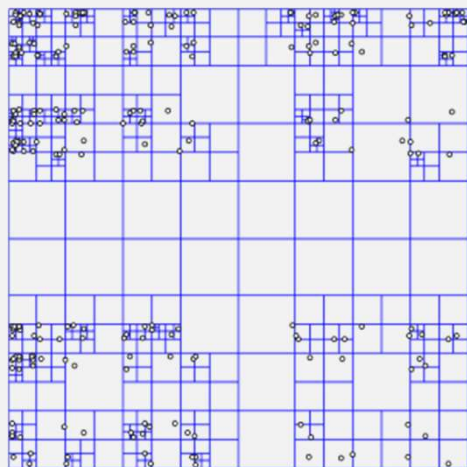
最近相邻粒子搜索法(NNPS)^[2]:

1. 全配对搜索法；2. 链表搜索法；3. 树形搜索法。

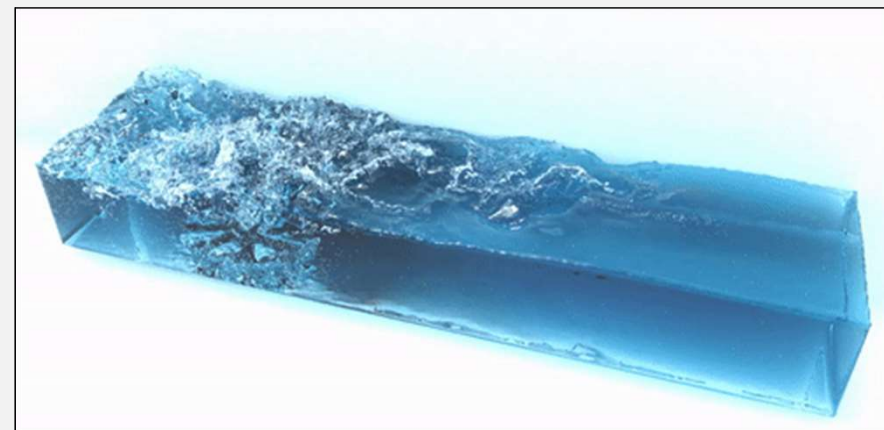
时间复杂度分别为 $O(N^2)$ 、 $O(N)$ 、 $O(N \lg N)$ 。



(链表搜索法) 插图来自互联网^[6]



(树形搜索法) 插图来自互联网^[7]



(SPlisHSPlasH, 采用紧凑哈希) 插图来自互联网^[4]

此外，还有一些新方法：

离散哈希、紧凑哈希^[3]。(SPlisHSPlasH^[4])

2 粒子搜索

算法的有趣例子^[5]：时间复杂度、空间复杂度

“老师让我把全班60本作业本按封面上的学号排好。

于是我灵活运用了快速排序的知识，从本堆中随便抽出一本，把学号比它小的本子放在左边，学号比它大的本子放在右边，再从左边这一堆挑出一本……

如此一来我的排本子的时间复杂度就从普通人用的插入排序的 $O(N^2)$ 变成了 $O(N \lg N)$ 。周围的同学投来好奇的目光，我洋洋自得，心想学过算法的我就是不一样。

快速排序效率果然很高，不一会儿，

我的桌子就放不下了_(:3」 ∠)_

(评论：忽略系数和空间复杂度的后果。)

四叉树(quadtrees)、八叉树(octree):

四叉树、八叉树在图像处理、物体碰撞、粒子法中有一些应用。

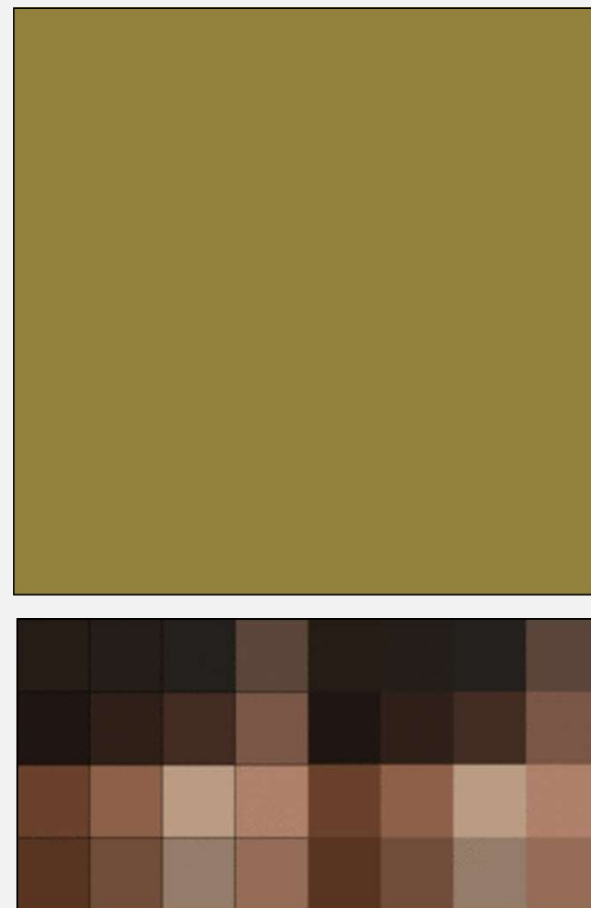
四叉树是在二维图片中定位像素的唯一适合的算法。^[7]

树形搜索法非常适合求解具有可变光滑长度的问题，这种算法通过粒子的位置来构造有序树（二叉树、四叉树、八叉树）。树形搜索算法的时间复杂度是 $O(N \lg N)$ 。

在实际编程实现中，往往可以采用面向对象方法、递归函数等编程手段实现树形搜索法，经有关资料证明，**结合树形搜索法的SPH实现是非常高效和健壮的，尤其是用于求解具有可变光滑长度和粒子数量庞大的问题时。**^[2]

插图来自互联网^[8]

3 粒子搜索实例：四叉树



四叉树(quadtree)编程实现：

3 粒子搜索实例：四叉树

仓库：zozuha/quad-tree(<https://github.com/zozuha/quad-tree>) (MIT开源许可证).

类（面向对象父类）：

点

矩形

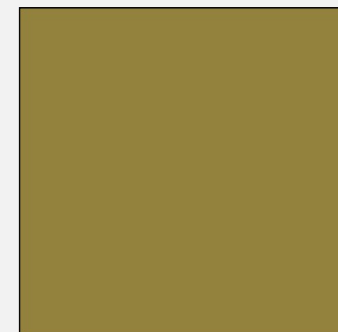
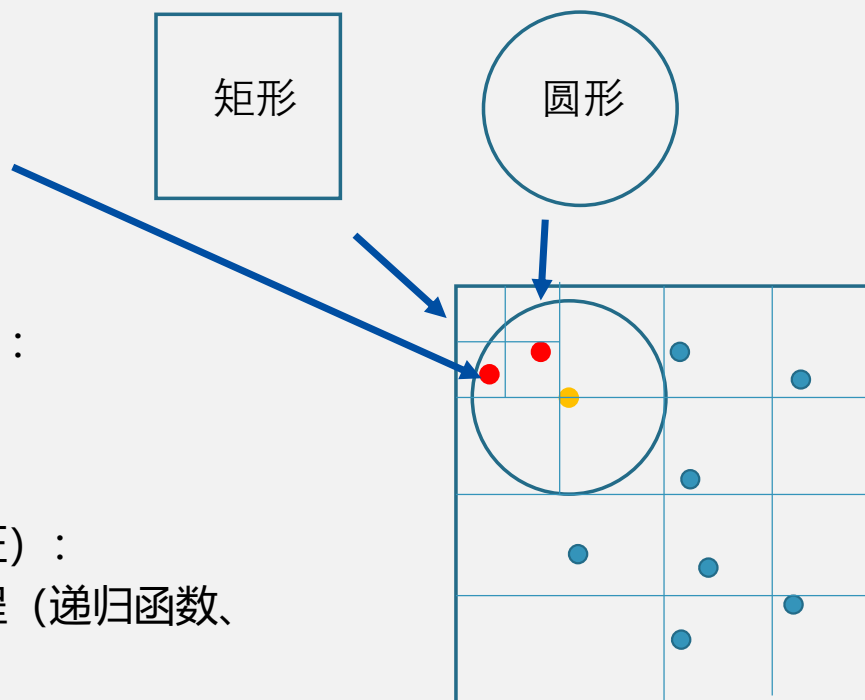
圆形

问题域和查找域（类的实例）：

构建四叉树和四叉树查询。

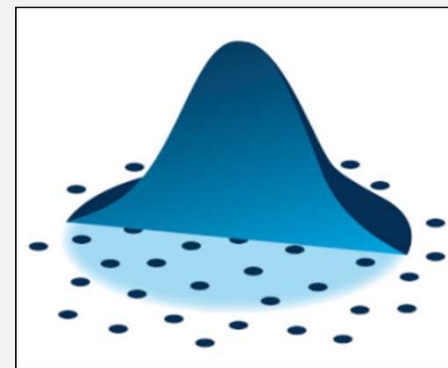
编程方法（健壮和高效的保证）：

面向对象式编程，函数式编程（递归函数、链表存储）。



误差来源：模型误差^[2]：
2处近似

$$\begin{aligned} f(r) &\approx \langle f(r) \rangle = \int_{\Omega} f(r') W(r-r', h) dr' \\ &\approx \sum_{j=1}^N f(r_j) W(r-r_j, h) \Delta V_j \\ &= \sum_{j=1}^N \frac{m_j}{\rho_j} f(r_j) W(r-r_j, h) \end{aligned}$$



插图来自互联网

湍流模型：

还在发展，在高雷诺数场景还存在适应性问题。（光滑长度、粒子数量）

4 不完全回答，两个问题

5 总结

汇报小结： 粒子方法与网格方法在实现上特性不同。相比于成熟的网格方法和实现，SPH方法还存在一些理论和实践的不足之处：理论完善、误差分析、数值加速算法。（已经有很多理论尝试在提高SPH的场景适应性和健壮性）SPH现阶段更适用于一些探索性、小型应用场景。



QQ群：光滑粒子流体动力学（SPH）学习群（667316027）。
现有群友24+6人。



开源代码托管仓库：[zozuha/SPH \(github.com\)](https://github.com/zozuha/SPH)（已支持四种构建系统：make/cmake/visual studio/fpm，已进行代码重构（Modernized Refactor））



静态演示网页：1. [交流研讨 | SPH \(zozuha.github.io\)](https://zozuha.github.io)：添加SPH Tutorial；
2. [课程：高等水动力学 | SPH-homework \(zozuha.github.io\)](https://zozuha.github.io)：添加第一次作业内容。



欢迎加入与参与！

参考文献

- [1] J.J. Monaghan. A turbulence model for Smoothed Particle Hydrodynamics[J]. 2011.
- [2] G. R. Liu, M. B. Liu. 光滑粒子流体动力学--一种无网格粒子法[M]. 湖南大学出版社, 湖南, 2005.
- [3] Dan Koschier, et al. SPH Techniques for the Physics Based Simulation of Fluids and Solids[J]. 2019.
- [4] SPLisHSPlasH. <https://github.com/InteractiveComputerGraphics/SPLisHSPlasH>.
- [5] 你在生活中用过最高级的算法知识是什么？ [EB/OL]. (2018-03-12)
<https://www.zhihu.com/question/67860343/answer/336070565>.
- [6] SPH（光滑粒子流体动力学）流体模拟实现二：SPH算法（4）-算法实现1[EB/OL]. (2019-10-21)
https://blog.csdn.net/qq_39300235/article/details/102657777.
- [7] 百度百科. 四叉树[EB/OL]. <https://baike.baidu.com/item/%E5%9B%9B%E5%8F%89%E6%A0%91/8557650>.
- [8] 维基百科. 四叉树[EB/OL]. <https://en.wikipedia.org/wiki/Quadtree>.



哈尔滨工程大学
HARBIN ENGINEERING UNIVERSITY

请老师和各位同学指正

汇报人：左志华 (zuo.zhihua@qq.com)

哈尔滨工程大学 船舶工程学院