基于物理的流体模拟入门

前言

FleX Demo

这是一个用英伟达 FleX 库实现的 Demo。FleX 是一款完全基于GPU的物理引擎，其中所有动态物体都是由粒子构成，可以很容易实现不同物体（刚体，软体，流体，布料等）之间的交互效果

目录

我们的内容要主要有三个部分。第一部分是流体基础，第二部分介绍基于力的流体模拟，第三部分基于约束的流体模拟

流体基础

接下来我们先介绍第一部分——流体基础。

观察流体——两种不同的视角

我们现在开始研究流体，我们先从观察流体入手。观察流体有两种不同视角

1、欧拉视角和拉格朗日视角

2、欧拉视角是在固定位置测量流过对应位置的物理量

3、拉格朗日视角是把流体视为运动的粒子，粒子携带了物理量

4、对于欧拉视角，你可以自己是岿然不动坐在那里，每时每刻问自己流过我这个位置的物质的物理量是多少。

5、对于拉格朗日视角，你可以把自己看成随波逐流的小船，每时每刻问自己，我在哪里，我的速度是多少，也就是自己是随模拟物质的一起移动。

6、左图是基于欧拉视角的烟雾模拟，右图是开场那个FleX Demo截图，是基于拉格朗日模拟。

总的来说，欧拉视角和拉格朗日视角只是从不同角度出发，研究同一个运动。

流体力学的白月光——NS方程

流体力学的白月光那就是NS方程。也就是纳维-斯托克斯方程。

NS方程经过欧拉、纳维、柯西和斯托克斯四位大神的不断完善，最后得出了适用于可压缩变黏度的黏性的最普适的流体运动方程，也奠定了NS方程在流体力学中犹如牛顿三大运动定律在经典力学中的地位。

目前方程的精确解求法数学家们暂时还没有找出来，所以一般都是弱解，也就是近似解。

向量微分基础

在开始详细介绍NS方程之前，先回顾一些向量微分基础

Nabla 算子、梯度、散度、拉普拉斯算子

Nabla 算子

先回顾一下nabla算子

Nabla 算子是一种向量微分算子

它直接作用于函数F，表示梯度

它与非标量函数F做点乘，表示散度

它与梯度做点乘是拉普拉斯算子

梯度

这个是梯度的数学定义和表示。我们再看一下什么是梯度物理意义？

1、梯度物理意义是沿梯度方向的方向导数最大，函数值增加最快

2、梯度有一个重要性质：梯度向量和等值面垂直。这个性质非常重要，我们后面会用到。

散度

接下来介绍一下散度

1、我们看散度的数学表达是怎样的，对于向量场A=P，Q,R，散度等于P、Q、R分别对xyz求偏导的和，可以记作 del 点乘 A。

2、而散度的物理意义是：向量场A在点M处的通量密度

3、什么是通量密度？M为中心，r为半径，做一个球体，球面记作 sigma r。

4、什么是通量?通量就是通过的量，所以向量场A通过曲面sigma r的通量就是，A在曲面上做面积分。当r趋向于0，单位体积的通量极限就是向量场A在点M处的通量密度

5、散度大于0，表示有流出；散度小于0表示有流入；散度为0，表示流出的等于流入。

6、如果向量场A处处散度为0，则成A为无源场

拉普拉斯算子

最后回顾下拉普拉斯算子

1、我们先看其数学定义，对于函数u=fxyz，那么拉普拉斯算子表示成u分别对xyz求二阶偏导数的和，记作 del 点乘 del u，也可以记作 del 平方 u

2、从定义可以看出，拉普拉斯算子是二阶微分算子。

3、拉普拉斯算子的物理意义是什么？它的物理意义是相当于对梯度场求散度

4、拉普拉斯算子实际描述的是函数在某一点周围的平均值与该点的函数值的差。

物质导数

接下来看一下什么是物质导数

1、物质导数针对的是流体微团，而不是空间固定的点。那什么是流体微团，无穷小的粒子，但是要足够大，大到能存放许多分子，使其被看成是连续介质。标量函数Q(x,y,z,t)，速度场V(u,v,w)

2、物质导数DQ/Dt表示为Q对时间t做偏导数，以及速度与函数Q梯度的点乘。

DQ/Dt的的和偏Q/偏t的差别，DQ/Dt流体微团在空间运动时的，物理量Q变化的瞬时时间变化率，而偏Q/偏t是在流体微团在固定点1时，物理量Q变化的瞬时时间变化率。

3、我们把这个定义为当地导数

4、而迁移导数在物理上表示流体微团从一点运动到另一点，因为流场空间不均匀性引起的时间变化率。也就是物质导数可以分成当地导数和迁移导数之和。

物质导数的物理意义是什么呢？假如我们去爬山，我们观察物理量温度的变化，中午温度比早上高，山顶温度比山脚低。也就是说你从一个点爬到另一个点，温度变化不仅跟时间变化有关，跟空间位置变化有关，这就是物质导数的物理意义。

5、物质导数鱼时间全导数的关系是什么呢？

6、物质导数实际上是对时间全导数，只是DQ/Dt的表示突显其物理意义，而dQ/dt数学上更正式一点

基于力的流体模拟

前面介绍完流体相关数学物理知识，接下来，介绍基于力的流体模拟。先介绍NS方程的推导

NS方程——两种视角

刚才观察流体提到有两种视角，所以NS方程也有两种视角的版本。

虽然欧拉网格法求出来的方程解更加精确，但是由于欧拉视角的NS方程太复杂，这里时间关系，我们先介绍相对简单的拉格朗日视角方程推导及其求解，下次再介绍欧拉视角相关的推导和求解。

NS方程——动量方程

现在我们开始从拉格朗日视角来推导NS方程。我们把流体看成是粒子构成的。

1、根据牛顿第二定律 F=ma，我们把a写成速度对时间的微分形式。

2、那么合力F是什么，我们进行受力分析。

3、首先是体积力，什么是体积力？就是穿越空间作用在流体粒子上的非接触力，这里就是重力

4、我们直接用mg可以得到

4、另外其他粒子还会对当前粒子产生作用力，也就是内力。

第一种内力是**压力**，

5、压力是由高压区指向低压区的，所以我们可以通过负压力梯度来测量粒子处的压力不平衡，

6、另外，还需要在体积上做积分，为了简单化，我们直接乘以V来表示，所以粒子受到的压力就是负的压力梯度乘上V

7、第二种内力是流体粘度引起的**黏力**。

8、粘性流体试图抵抗变形，这种内力试图使这个粒子以周围粒子的平均速度运动，也就是使邻近粒子之间的速度差异最小化。

9、而拉普拉斯算子衡量一个量与周围平均数的差，所以这里黏力用拉普拉斯算子表示。而黏力跟压力类似，为了方便计算，直接乘以V代替积分，并且加上粘性系数mu。

10、最后，把所有合力加起来，得到这样子的

11、两边同时除以质量m，进一步化简并且加入粘性系数，得到拉格朗日视角的NS方程

NS方程求解

刚才推导了NS方程拉格朗日视角版本，现在我们看如何求近似解

一种近似求解NS方程的方法 —— SPH

1、接下来我们看一下基于拉格朗日视角解流体力学的一个方法——smooth particle hydrodynamics

2、SPH是一种核密度估计（kernel density estimation，简称KDE）。把空间中的物理量用它周围一个范围内的相同物理量通过逼近 delta 函数的核函数来进行插值。

3、每个粒子代表一定的流体体积

4、属性存储在粒子上

5、属性是由其领域粒子的属性值加权决定；

6、其采用平滑核函数W来对权重进行插值。

核函数W实际上是一个中间大，两边小的函数。所以它可以让接近这个点x的粒子贡献更大，远离这个点的粒子贡献更小。当粒子离得特别远的时候，超过核函数半径h后，贡献就是0了。

SPH有个最大的好处，就是直观容易理解，而粒子天然支持并行，所以计算机实时应用更倾向于这种算法。

SPH

接下来我们看下我们怎么靠这个核函数来计算各种物理量的

1、空间中的任意位置xi的物理量A

2、那么物理量 Ai 的值就通过前面所说的邻域里的物理量 Aj 通过加权计算得到。这里的下标 j 表示邻域粒子，质量除以密度表示体积，这里我们为了计算简便，物理量对应的体积分都直接乘上 V 来表示。xi-xj 表示两个粒子距离，h是核函数半径

5、因为粒子间距离和核半径在公式里不影响计算结果，所以可以简化写法为

6、有些情况，我们需要用到一阶的，因此这里用核函数的梯度来近似

7、同样的，二阶求导，我们用核函数的二阶拉普拉斯算子来近似

SPH

前面提到NS方程是用 F=ma 得来的

1、我们把NS方程两边同时乘以密度 rho，可以得到形如这样的方程

2、这样的形式的方程，左右两边其实都是F/V，所以后面我们用小写 f 表示 F/V。也就是 f=F/V

3、按照前面说的SPH算法，我们算法先进行邻域搜索，找出一定范围内的粒子。

4、计算方程左右各项。

5、先计算左边的密度值，然后不管体积力、黏力还是压力的计算都需要用到密度，最后一次全部计算负压力梯度、黏力项、体积力项（重力项）。

SPH——密度

接下来我们看密度怎么计算

1、根据KDE算法，粒子i的物理量用邻域内粒子j物理量加权求得

2、这里物理量是密度，把密度rho代入

3、分子分母都有密度，可以约掉，化简可得粒子i密度计算公式

SPH——压力

计算完左边的密度后，我们看右边的压力项

1、这里借鉴理想气体状态方程，k是刚度系数，rho_0是静止密度，rho_i是粒子i的密度。把它引入到求解流体压力。这里可以看出压力和密度成正比。

2、如右图，当粒子i的密度大于静止密度的时候，压力就会变成排斥力，让其分开；当密度小于静止密度时，压力就会变成凝聚力，也就产生负压。然而负压会引入一些模糊不准确的问题，

3、所以这里限制负压出现

4、这里求的是压力梯度，所以这里用刚才提到的核函数梯度来求解压力梯度。

4、把压力pi直接代入KDE公式，得到压力梯度求解公式

5、那么这样子是否就正确呢？实际上，这里存在一个问题。我们来看一下两个相邻粒子的情况

6、p1和p2，代入压力梯度公式，

7、由于F=fV，代入求出压力，

8、那么可以看出来F1和F2只跟p1和p2有关，因此，如果p1和p2不相等，

9、那么计算出来的F1和F2就违反牛顿第三定律。

10、因此，不能使用这种形式的压力梯度公式，我们要用两个压力求平均的形式，这样的修改就不是真正梯度了，但是保持了力的相对性。

SPH——黏力

然后我们看怎么求黏力

1、黏力用二阶拉普拉斯微分算子表示。使用KDE二阶公式近似求解。

2、同样的黏力项也要满足力的对称性，也就是作用力与反作用力的大小相等。

3、根据黏力是流体相对运动产生的，所以黏力只依赖相对速度，不依赖绝对速度，也就是依赖速度差，

4、所以黏力项的KDE公式可以表示成邻居粒子j与粒子i的相对速度形式。

SPH——体积力

体积力，也就是重力，这个最简单了，直接密度乘以重力加速度。这里没什么可以分析的了

SPH——速度和位移

我们开篇也提到拉格朗日视角就是每时每刻问自己位置在哪里，速度是多少。

因此这里整个流程就是，

1、先合力的计算 F/V : f_i = 压力 + 黏力 + 体积力

2、根据牛顿第二定律 a = f/rho 求出加速度

3、然后根据速度和加速度关系，求出下一个时间步的速度

4、最后根据速度、时间、位置关系，求出下一个时间步的位置。

SPH——算法

整个算法的伪代码是这样的：

对整个流体的每一个粒子 i 搜索出其邻域粒子

对每一个粒子 i ，根据KDE公式求解密度

对每一个粒子 i ，根据KDE公式一次性求解压力梯度、黏力、体积力

最后对每一个粒子 i ，根据牛顿第二定律计算其加速度，根据加速度计算出速度，然后根据速度、时间、位移关系求出其最新位置

这就是SPH算法

SPH——核函数选取

SPH 算法有个需要讨论的问题，那就是核函数W的选取

1、核函数的选取对整个模拟的稳定性、准确性以及运行速度都会有影响。

2、一般选用比较多的是 poly6

3、和 spiky 核函数

但是poly6核函数并不完美，它本身有缺陷。那这个缺陷是什么呢?

右边上图是poly6，实线是原函数、短虚线是梯度，点虚线是拉普拉斯，从图上可以看出，原函数是符合单调性，而一阶和二阶函数都不符合单调性，也就是核函数算出来的权重不是从远到近，越来越大。而spily函数在一阶梯度和二阶拉普拉斯都具有很好的单调性。因此，我们一般估算时候原函数选用poly6，而一阶和二阶函数选用spiky。

当然，还有很多其他核函数可以选择，这个取决于其实现效果。

SPH——邻域搜索

1、另外还有一个问题最影响影响计算性能，那就是邻域搜索

邻域搜索最简单粗暴的方法就是遍历，当然，这样子这个时间复杂度只有 O n 平方，性能非常差。

2、为了提升性能，我们可以将空间划分成大小为 h 的单元

3、那么只需要搜索附近的27个单元就可以

4、大概步骤是:

创建网格

插入粒子

计算邻域，也就是搜索邻域粒子

当然，这里只是一种相对简单的方法，还有更多的提升性能的搜索算法，例如常用的空间哈希法等，这里算法选择得好，能让性能大大的提升。这里就不展开讨论了。

基于约束的流体模拟

刚才介绍了基于力的流体模拟，接下来介绍基于约束的流体模拟

基于力的动力学

我们再回顾刚才介绍过的基于力的动力学

1、我们一般都要受力分析，

2、计算内力，如流体的黏力、压力等

3、计算外力，如重力、碰撞力、风力等

4、把内力和外力加起来得出合力

5、根据牛顿第二定律，用合力求加速度

6、根据速度、加速度关系更新速度

7、最后根据速度、时间、位移关系，更新位置

基于力的动力学的缺陷

事实上基于力的动力学在计算机中有其缺陷，那缺陷是什么呢？

1、刚才也说了，一般的过程就是求力、求加速度、求速度、求位置这样子的顺序

2、这里最大的问题就是求力了。重力、摩擦力等都还算比较好求

3、碰撞力，这个就比较麻烦了。

4、从右图可以看到到，当两个物体穿透引发碰撞，

5、这时候根据碰撞力求速度

6、然后根据速度来求位置

7、而碰撞力是个瞬时力，作用时间极短，步长不好拿捏，一般都需要很小的步长。

8、其次，其值可能非常巨大

9、然后其随时间迅速变化，其规律非常复杂

10、因步长不能太大，所以需要数值积分

这些在计算机上模拟会提高了门槛，因此有人提出了另外的方法

Position Based Dynamics

为了减少某些力不好求解的情况，有人提出了一套基于位置的动力学，简称PBD，下面介绍下PBD。

基于位置的动力学（PBD）

1、PBD用约束投影代替力和数值积分

2、还是刚才的碰撞情况，PBD的过程是，如右图，我们只检测穿透发生引起的碰撞

4、根据碰撞约束计算物体修正位置，让物体分离

5、根据修正位置求解速度

开场动画使用的就是这套方案。这里没有求力，而是用约束来代替。

PBD算法

我们来看一下PBD算法具体步骤

对于一个N个顶点，M个约束表示动力学物体

1、初始化位置xi、速度vi、wi为质量的倒数，这里用倒数是因为一来可以通过无穷大质量也就是wi=0来表示静止物体，另外也能减少除法计算。

2、每一个时间步长进行迭代

3、先根据牛顿第二定律，通过外力求受外力影响的速度 vi

4、然后根据速度、时间、位置关系，预测当前时间步的位置 x*

5、接着检测是否产生碰撞约束，注意这里仅仅是检测，并不是碰撞约束投影求解

6、然后根据所有约束，做约束投影，计算出修正位置 delta p

7、根据修正位置，修正预测位置 x*

8、根据最终位置和当前位置，计算出当前时间步速度 vi

9、再用修正后的预测位置更新当前位置 xi

10、最后是根据摩擦力、恢复系数等非约束力，再对速度修正和位置，得出新速度和位置

PBD算法中位置修正

举个例子说明这个位置修正

一个圆上的粒子，位置在 xi

1、粒子 i 由于受外力影响，根据牛顿第二定律，计算出预测位置 x*i

2、然而这个位置并不是最终位置，根据约束投影求出修正位移 delta pi，使得粒子回到圆上。

3、最后根据修正后的位置计算出新速度

这个就是位置修正过程

PBD算法中速度修正

然后我们来看速度修正的例子

一个粒子 i 的位置为 xi，一个正方形物体静止在那里

1、粒子 i 受外力作用，根据牛顿第二定律，移动到 x*i这个位置

2、然后这个位置因为已经穿透到正方形物体里，所以产生了碰撞约束

3、根据碰撞约束投影，得到修正位移 delta pi，使粒子不穿进物体内部

5、然后这样子还不够，因为物体之间可能会有其他非约束力的产生，例如摩擦力

6、这里根据恢复系数和摩擦系数，再修正位置到 xi' 这里

7、最后根据这个位置更新速度

这里也就是最后还会计算一些非约束的力，进一步修正速度和位置。当然，这里是根据实际模拟物体来决定是否需要进一步修正速度。例如流体就没有恢复系数

约束

刚才提到了约束，那什么是约束呢?

1、约束是一个优化问题的解需要符合的条件

2、约束分为等式约束和不等式约束

3、约束有多种类型

4、模拟布料的距离约束，如右图，这是一种等式约束

5、模拟刚体、塑料的形状约束

6、模拟流体的密度约束

7、模拟气体的体积约束

8、无穿透的接触约束，如右图，这是一种不等式约束

这里只举一些常用的约束，还有很多种类型的约束，这里就不一一列出了。

PBD的物理意义

PBD的物理意义是什么?

1、PBD方法研究的是一个带约束的运动问题

2、举个例子，一个绿色小球沿着铁环在运动，在这个过程中，它除了受重力作用外，还会受到铁环的作用力，还有空气阻力、摩擦力等一系列的力。不管它什么速度，它运动轨迹肯定是在铁环上，像重力这种外力对小球的加速度贡献，我们可以很容易求出来，但是其他的力呢？铁环对它的力怎么估计呢？这就有点难办了，所以我们这里通过约束来求其近似值，把小球约束在圆形轨道上运动，而不去求解力。

3、这个思想其实就是高斯最小二乘约束原理

4、什么是高斯最小二乘约束？那就是受约束物体，它的运动轨迹是约束对加速度改变的总和的最小值

5、高斯最小二乘约束原理数学表达

6、红色部分就是约束对加速度的改变有多大。p头上两点表示位移对时间的二阶导数，也就是加速度，这个加速度就是最后真实的加速度，而F/m就是外力产生的加速度。

7、因此，PBD其实就是求满足约束的最小的位置改变，这就是PBD的P

高斯小二乘约束原理应用

按照这样的思路，我们把高斯最小二乘约束原理改一下形式:

这表示求满足函数最小值的参数pi。接下来根据这个思路，我们推导下简化的方程。

1、定义pit和vit分别为质点在时间步t时候的位置和速度，delta t 一个时间步长

4、质点i位置：等于 上一个时间步的位置 + 当前受外力影响改变的位移 + 修正位置 delta pi

5、质点i速度：等于两个时间步位置差除以时间，把公式（1）代入，求得公式（2）

6、质点加速度：等于两个时间步的速度除以时间，把公式（2）代入，求得只含修正位移和外力相关的公式（3）

7、把公式（3）代入最小二乘约束原理里，化简可得

8、因为求的是满足函数最小值的参数，所以可以直接去掉 delta t 平方，也为了方便后面求导运算，加上个1/2

9、把求最小值参数的函数写成矩阵形式

10、最后加个约束，位置要满足约束，C(p)=0，加上修正位移delta p 后也要满足约束，也就是C(p+delta p)=0

单个约束优化求解

接下来我们看单个约束优化求解

1、我们把公式写成更专业的形式，其中s.t.表示subject to，也就是约束于。现在就是一个约束优化问题。即满足约束 C(p+delta p)=0条件下，求满足函数最小值时的参数delta p

2、那么怎么求解呢？这里我们引入拉格朗日乘子法。

3、我们把约束优化写成这种形式，其中 fx 是一个标量场，gx 是约束函数。

4、如图所示，蓝色线为 fx 等值线，当没有约束的时候，极值应该在最小的蓝色线上，这里没画出来，应该是无限趋近中心，其实是一个点。

5、但是加了 gx 函数约束后，那么 fx 最小值只能在 黑色线上找。

6、有了函数 gx 约束后，极值应该在蓝色线和黑色线的共同切线上

7、蓝色箭头表示梯度的反方向。因为相切，两条线的梯度方向相同或者相反，也就是两函数梯度是平行的。

7、因此满足方程 fx 梯度 + lambda 乘 gx梯度 = 0。所以对于等式约束优化，当函数梯度等于等式约束的梯度的线性组合时，可以找到最优解。

8、拉格朗日乘子法就是定义一个新函数拉格朗日函数，对该函数求导并令其为0，就得到上面方程。

9、现在令f(p)=1/2 delta p 的转置乘质量矩阵 M 乘 delta p ,g(p)=C(p)

10、引入拉格朗日乘子 lambda

11、可以得到一个方程

12、为了方便后面表示，写成矩阵形式。一个方程两个未知数，怎么解？

单个约束优化求解

1、通过拉格朗日乘子法得出一个方程两个未知数

2、别忘了，我们还有约束方程 C(p+delta p)=0。但是约束函数可能是线性的也可能是非线性。我们把其变成线性，更好求解。

3、我们通过泰勒展开，把这个非线性约束方程近似成线性方程

4、两个等式联立方程组，可得

5、两个方程两个未知数，可以分别求得拉格朗日乘子 lambda 和 修正位置 delta p

多个约束优化求解

1、前面讨论的是N个粒子受1个约束的情况

2、现在来看N个粒子受M个约束的情况，也就是多约束的情况

3、这里可以看到由M个约束构成了一个方程组。

4、由于物理模拟中约束的个数，梯度的维度都无法保证，彼此间的是否线性相关也无法保证，所以这个方程组可能有唯一解，也可能没有解，也可能有无限多个解。

5、如图所示，三个等式约束，要同时满足三个约束的点是不存在的，所以是无解的

6、那么是不是就无法求解呢？在PBD中直接无视是否有解。直接通过迭代法去求解。

7、PBD中常用的迭代法有高斯-赛德尔迭代。什么是高斯赛德尔迭代呢？那就是在方程组中，先求出 delta p1，然后代入到第二个方程中，求 delta p2，不断迭代，最后求出近似解。

7、PBD中还可以用雅可比迭代，也就是每个方程各自计算求解，不依赖前一个方程的解作为下一个方程的输入

约束求解器

1、如右图，高斯赛德尔先代入第一个方程，函数解在解空间 l2 上。第二步代入第二个方程，函数解在解空间 l1 里。也就是说用高斯赛德尔方法解会在解空间之间来回跳跃，然后慢慢的靠近共同的解空间。从这里可以看出高斯赛德尔迭代迭代速度不快，并且因为依赖上一个解作为下一个的输入，所以不可并行。

2、雅克比迭代法代入方程1，计算出到f1解空间的向量，代入f2，计算出另一个到解空间f2的向量，然后向着两个向量的合向量前进，寻找f1和f2的相交解空间。这样的话如果f1，和f2的解空间在同一个方向，雅克比迭代法使用合向量作为步长，经常会一步迈过，下一步再迈回起始点，导致不能收敛。从这里也可以看出，雅可比迭代收敛可能更慢，甚至不收敛，但是因为不依赖上一个结算结果，所以可以并行

3、为了解决雅可比不收敛问题，有人提出平均雅可比法。就是对影响粒子i的粒子数量求平均

4、又有人觉得这样子收敛还不够快，加入了超松弛因子，进一步加快收敛。这个也是个数学上方程组迭代求解的方法，这里就不展开讨论。

约束求解优先级

前面讨论约束的时候，提到约束有许多不同的约束类型

1、我们按照约束类型分组，构造不同优先级

2、优先级高的先处理，然后把 delta p i 累加到 p i 上，再处理低优先级的

4、如，先处理碰撞约束，再处理密度约束。

这样做能加快约束修正位置的收敛速度，能够更快接近真实解

Position Based Fluid

以上是基于位置动力学的基础，有了这个基础，我们看PBD在流体中的应用，也就是PBF

PBF——流体的密度约束

前面我们提到，流体我们使用的是密度约束。

1、在不可压缩流体模拟中，我们希望粒子i的密度尽量与静止密度rho_0相同，因此需要针对每一个流体粒子都施加一个密度约束，也就是 C_i() = rho_i / rho_0 - 1 = 0

2、从这里看到，我们已知静止密度 rho_0，所以我们需要求的只有粒子密度 rho_i

3、那怎么求 rho_i 呢?

4、前面 SPH 的 KDE 算法提到 物理量 Ai 是用附近邻域内的对应物理量的加权和估算得到。

5、这物理量是密度，因此把密度代入，可得

6、而刚才PBD中使用到了梯度，则根据KDE梯度公式，流体KDE中用到的梯度是。其中 k 是包括自身i和邻居j

PBF——流体的密度约束

1、大家注意了，之前SPH中提到的都是邻居 j，而这里用了 k，而k是包含自己的，所以梯度分成两种情况：

当 k=i，也就是k是自己时，通过所以邻居粒子的密度加权求和估算得到；当 k=。j 时，也就是k是邻居粒子时，直接用邻居粒子的梯度求得

2、怎么理解？我们可以这么理解：当 k=i 时，表示约束函数 C_i 关于 x_i 的梯度，方向为 x_j 出发指向 x_i。

3、当 k=j 时，表示约束函数 C_i 关于 x_j 的梯度，方向为 x_i 出发指向 x_j 。

4、我们在 SPH 那一节也提到，W选择 poly6，而梯度选择 spiky

5、根据PBD算法，还需要求拉格朗日乘子

6、拉格朗日乘子，对于一个约束c_i 中所有粒子而言都是一样。这里的 lmabda i 的下标 i 是指多约束中的第几个约束。

PBF——拉格朗日乘子中的除0问题

上面的拉格朗日乘子法中，还有个除0的问题。

如果一个约束条件不能被违反，则称为硬约束，反之，能够一定程度上被违反称为软约束。理想情况下，我们希望都是硬约束，然而由于计算机误差或者数值稳定性等原因，我们有时也需要约束呈现软性质。

1、两个粒子距离r=x_i-xj

2、当两个粒子距离r等于核半径h时，则核函数W=0，如果粒子之间都处于这种状态，由前面的PBF梯度公式可以知道，所有梯度求出来都是0.

3、从而前述的拉格朗日乘子的分母则为0

4、为了解决这个问题，PBF借鉴了 open dynamics engine 中的混合约束法，使密度约束变成软约束。具体做法就是加入松弛因子 epsilon。

PBF——位置修正

上面的拉格朗日乘子法中，还有个除0的问题。

如果一个约束条件不能被违反，则称为硬约束，反之，能够一定程度上被违反称为软约束。理想情况下，我们希望都是硬约束，然而由于计算机误差或者数值稳定性等原因，我们有时也需要约束呈现软性质。

1、两个粒子距离r=x_i-xj

2、当两个粒子距离r等于核半径h时，则核函数W=0，如果粒子之间都处于这种状态，由前面的PBF梯度公式可以知道，所有梯度求出来都是0.

3、从而前述的拉格朗日乘子的分母则为0

4、为了解决这个问题，PBF借鉴了 open dynamics engine 中的混合约束法，使密度约束变成软约束。具体做法就是加入松弛因子 epsilon。

PBF——Tensile Instability

什么鬼畜问题呢?

对于采用SPH估算密度的流体模拟方法，通常需要30-40个邻居粒子才能使密度求值结果趋向于静态密度。

1、在邻居粒子不足的情况下

2、会导致求出的流体密度低于静态密度，也就是前面提到的负压问题

3、负压会导致产生不符合真实情况的凝聚现象

4、如图所示，就会出现这种现象

5、那么如何避免这种现象出现呢？解决方法有两种

6、添加一种排斥力，避免粒子凝聚，前面也提到了一般求力都需要耗费不少力气，所以还有另外一种更快的方法

7、另外一种方法是这样的。前面提到约束有等式约束和不等式约束。等式约束总是会进行约束投影操作，而不等式约束只有违反不等式的时候，也就是C_i <= 0的时候，才进行约束投影。因此，只有rho_i/rho_0-1<=0时候才进行约束投影。

8、直观理解就是只有粒子靠得比较进的时候，才需要进行让粒子分开的操作，而当约束条件满足的时候，就不进行约束投影了，这就避免的凝聚问题。

后续

前面介绍了基于拉格朗日视角的NS方程及其求解，也介绍了工程实现上的PBD和PBF方法。而流体力学及其模拟方面，还有很多事情可以做的。例如

1、用于实时应用的统一的粒子物理系统，这也是 FleX 和 obi 的系统框架，可以很方便的实现各种不同物质类型的之间的交互模拟

2、还有前面只介绍了相对简单的拉格朗日视角的求解，还有可以进一步了解欧拉网格法

3、除此以外，还可以使用混合欧拉-拉格朗日法

4、还有上面提到的统一粒子系统，需要通过各种约束来实现的

5、在计算机上，为了提高性能，可以加入并行计算

6、还有流体中最复杂的涡流和湍流的模拟

7、还有这次只介绍了物理模拟部分，还可以去了解流体渲染相关的

8、还有很多很多的，例如MPM等方法也可以进行模拟

参考文献

这里是相关的一些参考文献，当然，这不是全部。