

稳定且维护细节的液体与烟雾模拟算法

董泽建², 杨 猛^{1,3}, 刘晓君²

(1. 中国科学院计算技术研究所智能信息处理重点实验室, 北京 100190;

2. 山东丝绸纺织职业学院信息工程系, 山东 淄博 255300; 3. 中国科学院研究生院, 北京 100190)

摘 要: 半拉格朗日方法在欧拉模型的液体与烟雾模拟中, 因均值运算过多而存在数值粘性的问题, 且表面提取算法会抹平液体表面的细节。为此, 提出一种稳定且维护细节的液体与烟雾模拟算法。采用精确的数值方法求解对流项, 在流体的边界及内部分布隐式的无质量粒子, 利用简单的插值策略以及粒子和网格的双向影响, 保证数值的稳定性, 减少数值粘性, 并结合基于显式曲面机制的表面追踪方法来实现丰富的视觉效果。实验结果证明该算法具有维护细节的能力。

关键词: 流体模拟; 对流项; 数值粘性; 半拉格朗日方法

Stable and Detail-preserving Liquid and Smoke Simulation Algorithm

DONG Ze-jian², YANG Meng^{1,3}, LIU Xiao-jun²

(1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;

2. Department of Information Engineering, Shandong Silk Textile Vocational College, Zibo 255300, China;

3. Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

【Abstract】 In liquid and smoke simulation using Euler model, the common Semi-Lagrangian Method(SLM) has numerical viscosity due to excessive averaging steps, and the surface extracting method will further smear out details of water surface. This paper proposes a stable and detail-preserving liquid and smoke simulation algorithm. It employs numerical method with higher numerical accuracy to solve advection term. The algorithm distributes massless implicit particles near the surface and in the fluid volume, using simple interpolating strategy and utilizing the bidirectional influence between particles and grid, achieving the target of guaranteeing numerical stability and reducing numerical viscosity. It combines with surface tracking method based on explicit surface to achieve rich visual effects. Experimental results demonstrate the algorithm has the capability of preserving details.

【Key words】 fluid simulation; advection term; numerical viscosity; Semi-Lagrangian Method(SLM)

DOI: 10.3969/j.issn.1000-3428.2011.15.003

1 概述

基于物理的流体模拟能体现更符合真实物理状况的流体形变及流体与其他介质的作用, 因此, 它在 CG 动画、视频游戏、自然灾害模拟中有重要的应用价值, 一直是图形学领域的研究热点。在图形学领域基于物理的流体模拟的大部分知识与经验都来源于计算流体动力学(Computational Fluid Dynamics, CFD)。其目标是精确地计算各个时刻每个采样点处流场变量的数值, 而在图形学领域中, 追求的是在高效率下保持好的视觉效果, 但求解可能不精确符合真实的物理状况。流体控制方程 Navier-Stokes(N-S)方程提供了一个原理性的求解模型, 该领域的多数文献都是针对于如何快速、稳定地求解 N-S 方程的。

流体模拟领域有两大主流方法^[1]: (1)欧拉法, 即将空间离散为网格, 在每个网格的若干固定位置上采样各流场变量, 通过求解 N-S 方程得到在这些固定的网格位置上的每一时刻的变量数值, 然后对这些采样点的数值进行插值计算来决定任意位置的流场属性。(2)拉格朗日法, 也称粒子法, 通常采用 SPH(Smoothed Particle Hydrodynamics)模型。该方法用有质量的粒子为流体建模, 追踪每个粒子的受力, 再用牛顿定律来搬运粒子。

SPH 和欧拉模型各有优缺点, 如欧拉法易于形成平滑的表面而 SPH 易于体现飞溅等小的细节。本文的研究着眼于欧

拉模型, 通过对自由表面追踪策略和对流项求解方法的改进减少速度场的数值粘性, 并体现更丰富的表面视觉效果。

2 流体控制方程及数值方法

2.1 N-S 方程

无论液体还是烟雾, 它们的控制方程都是不同形式的 N-S 方程:

$$\frac{\partial \mathbf{V}}{\partial t} = -\mathbf{V} \cdot \nabla \mathbf{V} - \frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \mathbf{F} \quad (1)$$

$$\nabla \cdot \mathbf{V} = 0 \quad (2)$$

其中, 向量场 $\mathbf{V}$ 表示流体速度场, 它的 3 个分量 V_x 、 V_y 、 V_z 分别表示沿 3 个坐标轴正方向的标量值; ρ 为流体密度; p 为流体内部压力; μ 表示动态粘性系数; 向量 $\mathbf{F}$ 表示体积力(比如液体的重力、烟雾的浮力)产生的加速度; $\boldsymbol{\tau}$ 表示粘性应力张量; ∇ 表示梯度算子即 $(\partial/\partial x, \partial/\partial y, \partial/\partial z)$; $\nabla \cdot$ 表示散度算子。式(1)为动量方程, 描述了流体的动量守恒; 式(2)规定了流体的不可压缩性, 即速度场在任意位置散度为 0。

本文的模拟对象针对无粘流, 旨在稳定且有效地降低流体的数值粘性, 使流体尽可能呈现出理想的无粘流所表现的

作者简介: 董泽建(1981—), 男, 硕士研究生, 主研方向: 计算机辅助设计, 计算机图形学; 杨 猛, 博士研究生; 刘晓君, 硕士研究生

收稿日期: 2011-01-14 **E-mail:** dongzejian_2001@126.com

飞溅(液体)、漩涡(烟雾)等细节效果。因此, 本文忽略动量方程中的粘性项。

2.2 数值方法总体分析

文献[2]使用算子分割的数值方法来求解动量方程, 并引入了半拉格朗日方法(Semi-Lagrangian Method, SLM)来求解对流项。以文献[2-3]中的经典框架算法为例, 假设速度场 V^n 为第 n 时刻的求解结果, 若要求解第 $n+1$ 个时刻的速度场 V^{n+1} , 则算法依次求解对流项 $-V \cdot \nabla V$ 、体积力项 F 来更新当前速度场, 最后求解压力项 $-\nabla p / \rho$, 利用压力场 p^{n+1} 将当前的速度场 V^* 投影为零散度的速度场, 即下一时刻的速度场 V^{n+1} , 该过程如下:

- (1) 求解对流项: $V^* = \text{advect}(V^n)$;
- (2) 添加体积力加速度: $V^* += \Delta t F$;
- (3) 求解 $n+1$ 时刻的压力值: $p^{n+1} = \text{Project}(V^*)$;
- (4) 求解 $n+1$ 时刻的速度场: $V^{n+1} = V^* - \frac{\Delta t}{\rho} \nabla p^{n+1}$ 。

本文用标准的中央差分来离散控制方程。然而, 对流项 $-V \cdot \nabla V$ 的显式离散式的雅可比矩阵的特征值全为虚数, 所以直接推进求解的方法会导致数值膨胀。虽然可以将对流项离散为流量守恒形式的差分方程来保证数值的稳定性, 但这种方法受到严格的时间步长约束。因此, 文献[2]引入 SLM, 使对流项的求解允许较大的时间步长, 但它因均值运算过多, 极易导致数值耗散, 使得流体获得额外的人工粘性, 大幅抹平了高频细节, 这在烟雾模拟中最为明显, 使得烟雾中常见的漩涡现象消失。

文献[4]通过高阶插值的方法来锐化 SLM 达到减少数值粘性的目的。文献[5]通过添加额外的力场的方法来恢复一些被抹平的细节。但本文的算法采用了更加简单有效的策略, 使最终效果呈现出更低的粘性。本文的改进之处在于:

- (1) 使用显式的流体表面追踪方法;
- (2) 在流体域分布隐式的无质量粒子, 利用粒子携带的速度信息来求解对流项, 达到降低数值耗散的目的。

3 流场配置及液体的自由表面追踪

3.1 流场配置

本文使用的依然是使用规则的交错网格来离散流场, 即在每个单元格的中心采样压力, 在 3 个面心处采样速度分量 V_x 、 V_y 、 V_z , 如图 1(a)所示^[2]。借用图形学体渲染领域的术语, 每一个单元格可称为一个体素, 对每个体素使用三维下标来作为位置索引。初始化流场时还需要在每个单元格内生成粒子。本文在每个体素内生成 8 个粒子, 每个子单元内生成 1 个, 如图 1(b)所示。

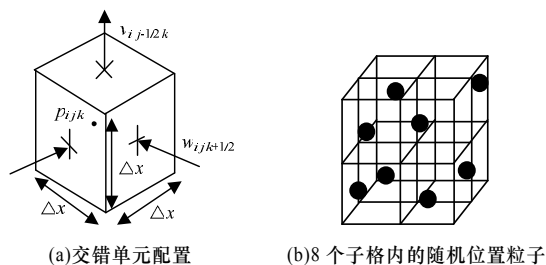


图1 网格配置示意图

单个粒子的数据结构如下:

```
typedef struct particle
{
    //Vec3f 为之前定义的三维向量结构
```

```
Vec3f Pos; //粒子位置
Vec3f velInfo; //粒子携带的速度信息 Vp
... //粒子携带的其他信息, 如纹理、温度、密度等
float radius; //粒子半径
} particle;
```

在液体模拟中, 粒子的半径是可变的, 其值为 Pos 到自由界面的距离^[6-7], 且最大值为 $0.5\Delta x$ (设 Δx 为单元格的边长), 最小值为 $0.1\Delta x$ 。

3.2 基于显式曲面机制的自由界面追踪

为了可视化流场, 烟雾需要密度场数据, 而液体则需要抽取用三角形带表示的自由界面^[8-9]。在大多数的流体模拟的文献里, 液体的自由界面都是通过隐式曲面来表示的, 即通过在网格上定义有向距离场来估算自由界面的位置, 然后通过求解 LevelSet 方程来更新自由曲面^[10]。而可视化时, 一般用 MarchingCube 算法根据当前网格的距离场数据来抽取出显式的三角形带, 然后渲染三角形带即可^[11]。但限于计算网格的分辨率, 求解 LevelSet 方程时会产生相应规模的数值误差, 虽然这些误差可以通过 Particle Level Set 算法^[6]来进一步修正, 但之后的 MarchingCube 过程仍然会进一步造成细节的丢失。因为 MarchingCube 算法在某一条单元格的边上最多计算一个交点, 而且是通过线性插值来估算, 实际上自由界面完全可以和一条边有多个交点。

本文使用的自由界面追踪方法是基于显式曲面的, 即直接用显式定义的三角形带来表示自由界面, 在对流过程中, 每个三角形的顶点都根据当前的速度场移动。因此, 三角形带的拓扑结构是不断变化的, 与 MarchingCube 算法抽取的三角形带相比, 显式曲面能更好地为飞溅这种小规模细节建模, 并且比隐式曲面更易于体现更丰富的几何特征尤其是尖锐的波浪。然而, 三角带的重构过程比之求解 LevelSet 方程要更加困难复杂, 但有一个非常实用的工具——El ToPo, 可以高效地处理三角形带的平流及重构^[12]。因此, 本文使用 El ToPo 来追踪自由界面。

4 改进的对流项求解

本文用粒子来求解对流项, 由于采用了显式曲面来追踪表面, 粒子在本算法的作用仅仅是减少数值粘性, 并不参与自由表面的重构过程。因此, 本算法中的粒子可看作是一种隐式粒子。

根据当前的速度场来移动粒子, 粒子的移动速度可通过周围相应的网格采样点处的速度进行三线性插值来计算, 三线性插值示意图如图 2 所示, 其中, 控制点为周围的 8 个网格顶点。

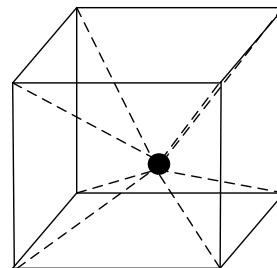


图2 三线性插值示意图

当移动粒子时, 本文使用二阶 Runge Kutta 时间积分法^[2]。对于每个粒子, 根据其平流后的位置, 将粒子携带的各个信息按距离权重来将其贡献累加到附近的网格点上。以任一粒子 P 携带的速度信息 V_p 为例, 对周围任意一网格点 X

而言,要计算粒子 P 对 X 的速度值的贡献,则 P 的权重与 P 到 X 的距离 d 相关。因此, X 点的速度值计算方式如下:

$$V(X) = \sum_{i=1}^n (R-d_i) V_{P_i} / \sum_{i=1}^n (R-d_i)$$

其中, n 代表以 X 为中心; 半径为 R 的包围球内的粒子的数量。因此, 对粒子 P 的权值的计算可以看作是一种线性衰减函数, 即距离越远, P 对 X 的影响越小。这里本文将半径 R 的值设为 $1.2\Delta x$ 以提高计算效率。

注意: 粒子携带的速度信息 V_P 与粒子本身的速度并不相等, 前者用于更新各个网格点的速度, 后者决定粒子自身的运动。

在平流粒子之前, 需要先平流液体的自由界面, 而对于烟雾而言, 需要平流烟雾的温度 T 与浓度 S , 因为 T 与 S 决定了烟雾所受的浮力, 有:

$$\frac{\partial T}{\partial t} + V \cdot \nabla T = 0$$

$$\frac{\partial S}{\partial t} + V \cdot \nabla S = 0$$

$$f_{\text{buoy}} = -\alpha S + \beta(T - T_{\text{amb}})$$

其中, f_{buoy} 为浮力; α 与 β 为大于 0 的自定义常量; T_{amb} 为环境温度。求解完对流项后, 将当前的速度场保存为 V_{save} 。

5 压力的求解

求解压力的主要目的有 2 个:

- (1) 保证流体的不可压缩性, 即质量守恒;
- (2) 满足自由界面处及流体-固体界面处的边界条件。

因此, 对于任何一个单元的压力采样点(即单元中心), 需要满足方程:

$$\frac{\Delta t}{\rho} \nabla \cdot (\nabla P^{n+1}) = \nabla \cdot V^*$$

所有的方程将联立为一个稀疏的线性系统, 采用矩阵的表示方法, 如果用 G 来表示用中央差分来离散的梯度算子矩阵, 用 D 来表示相应的散度算子矩阵, 则线性系统可表示为:

$$DG\hat{p} = D\hat{V}^*$$

其中, $\hat{p}$ 表示所有的压力点组成的向量; $\hat{V}^*$ 表示相应的所有的存储于各个面心处的中间速度组成的向量。如果不考虑边界条件, 即将所有单元看作流体单元, 则从中央差分的形式可以看出 D 其实是 G 的负转置, 即 $D = -G^T$ 。系数矩阵 $DG = -G^T G$, 因此, 该系数矩阵是一个标准的二次型。

如果考虑边界条件。首先看流体-固体处的边界条件, 该条件要求流体不能流入固体, 即边界处流场速度与固体在法线(可由表示固体的三角形带计算)方向上保持相同。本文使用同文献[2]相同的方法, 即在与固体边界相交的单元的面心处直接用固体法向速度的各分量来设定 V^{n+1} 。

对于自由界面处的边界条件, 可认为自由界面上的压力值为 P^{air} , 而空气域内属性值可看作流体域的连续延伸, 且为线性的延伸^[13]。例如, 2 个压力点 $P_{i,j,k}$ 和 $P_{i+1,j,k}$, 若 $P_{i,j,k}$ 处于流体域而 $P_{i+1,j,k}$ 处于空气域, 则 $P_{i,j,k}$ 与 $P_{i+1,j,k}$ 满足线性关系:

$$\theta P_{i+1,j,k} + (1-\theta) P_{i,j,k} = P^{\text{air}}$$

$$\theta = \frac{d(P_{i,j,k})}{d(P_{i,j,k}) + d(P_{i+1,j,k})}$$

其中, 函数 $d(X)$ 表示点 X 到自由界面的距离。由于是显式曲面, 可以通过求到最近的三角形的距离来计算 d , 因此离散位于 $P_{i,j,k}$ 与 $P_{i+1,j,k}$ 中点处的梯度 ∇p 时, 将上式代入就可以将

消去 $P_{i+1,j,k}$:

$$\nabla p = \frac{P_{i+1,j,k} - P_{i,j,k}}{\Delta x} = \frac{P^{\text{air}} - P_{i,j,k}}{\theta \Delta x}$$

将上式代入到 G 中对应的矩阵行中, 得到的矩阵依然是二次型的形式, 具体而言就是常系数泊松方程。求解泊松方程, 最高效的方法是带预处理的共轭梯度算法(Preconditioned conjugate gradient, PCG), 通常可以在 10 个迭代步内求得较为精确的数值解。求解 $n+1$ 时刻的压力后, 计算 $n+1$ 时刻的速度场:

$$V^{n+1} = V^* - \frac{\Delta t}{\rho} \nabla p^{n+1}$$

然后, 更新粒子的信息。其中, 对于烟雾的温度、浓度这些属性, 可以直接用周围网格点的值插值计算。而对于粒子携带的速度信息 V_P , 算法不对其直接更新, 而是更新 V_P 的增量, 及先计算周围网格点的速度的增量: $\Delta V = V^{n+1} - V_{\text{save}}$, 然后将 ΔV 插值到粒子的位置, 作为 V_P 的增量。

SLM 与本算法在物理原理上是相同的, 即流场中某一空间位置处的物理信息来自其上游。SLM 方法相当于假设上游处恰好有一个粒子在 Δt 时间后经过当前待求解的网格点, 且该粒子的值由周围的网格点的值来决定, 插值模板为三线性模板; 类似的, 本文算法相当于若干上游的粒子在 Δt 时间后到达以待求网格点为球心 R 为半径的球形区域内, 这些粒子携带的信息一起决定了该网格点的值。本文算法的粒子线性插值模板如图 3 所示。

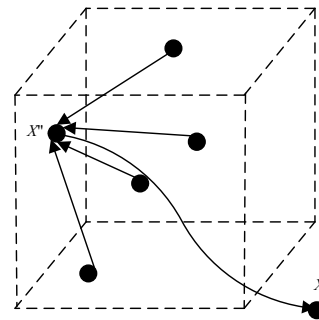


图 3 本文算法的粒子线性插值模板

相对于 SLM 方法, 本文算法大幅地减少了速度场数值粘性, 且具有同等的数值稳定性。原理在于:

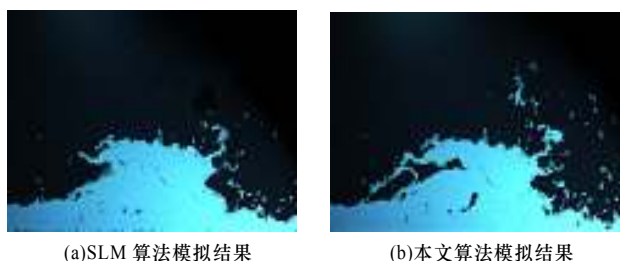
(1) 该方法基于实际的物理信息由粒子携带这个事实, 且网格与粒子速度的相互制约保证了数值的稳定。

(2) 数值粘性源于插值运算, 插值运算可看作一种平滑操作。在最后更新粒子的 V_P 时, 被平滑的不是 V_P 本身而是 V_P 的增量, 这就等于间接地提高了 V_P 的误差的阶, 因此数值粘性被抑制。

注意: 在真实的物理情况中, 烟雾的温度 T 、浓度 S 在平流时相当于一种耗散过程。因此, 更新粒子信息直接平滑 T 与 S 就相当于通过数值方法完成了这个耗散的物理过程。

6 实验结果与分析

本文的实验环境为 WindowsXP, 开发环境为 Microsoft Visual Studio2005, 平台配置为 Intel 2.83 GHz 4 Core CPU, 2 GB 内存, NVIDIA Geforce 9800GT 显卡。渲染工作通过输出的隐式平面数据传送给渲染器来完成, 渲染器为 sprites 和 pixie。图 4 展示了一个二维小规模波浪场景以观察不同算法对耗散的抑制, 帧率为 40 帧/s。可以看出, 本文算法的数值耗散更低, 对波形的尖锐度、飞溅等细节体现得更好。



(a)SLM 算法模拟结果

(b)本文算法模拟结果

图4 SLM算法与本文算法在数值粘性上的对比

液体与烟雾的模拟截图如图5所示, 所模拟的场景的分辨率为 $100 \times 90 \times 80$, 速度为 170 s/帧。图5的渲染都应用了体绘制技术^[14], 从图5(b)可以看到, 用显式表面的自由界面追踪方法配合低数值粘性的对流项求解方法使得液体的表面呈现出复杂的拓扑变化, 且可以在自由界面附近成功地解析比较稀薄的水层。从图5(c)可以看到, 本文的算法可以较好地保持速度场的旋度, 并且本文的算法没有添加额外力场或求解旋度方程, 烟雾中的较大的漩涡得益于本文的对流项算法, 而小规模漩涡则得益于算法在大漩涡附近添加的简单噪声模型。



(a)第30帧液体模拟



(b)第50帧液体模拟



(c)第50帧烟雾模拟



(d)第70帧烟雾模拟

图5 液体与烟雾的模拟截图

7 结束语

本文利用粒子携带的速度信息求解对流项, 采用简单的插值策略达到了降低数值粘性的目的, 同时结合显式的自由表面机制得到了精良的视觉效果。但遍布流域的粒子会造成计算及存储的负担加大, 因此, 还可以考虑将本文算法与SLM方法耦合的策略, 利用一些高阶插值方法或通过改进网格配置进一步减少数值粘性, 这是下一步的研究方向。

参考文献

- [1] Foster N, Metaxas D. Realistic Animation of Liquids Graph[J]. Models Image Process, 1996, 58(5): 471-483.
- [2] Stam J. Stable Fluids[C]//Proc. of SIGGRAPH'99. Los Angeles, California, USA: [s. n.], 1999.
- [3] Foster N, Fedkiw R. Practical Animation of Liquids[C]//Proc. of SIGGRAPH'01. Los Angeles, California, USA: [s. n.], 2001.
- [4] Molemaker J, Cohen M J, Patel S, et al. Low Viscosity Flow Simulations for Animation[C]//Proc. of SIGGRAPH'08. Dublin, Ireland: [s. n.], 2008.
- [5] Fedkiw R, Stam J, Jensen H W. Visual Simulation of Smoke[C]//Proc. of SIGGRAPH'01. Los Angeles, California, USA: [s. n.], 2001.
- [6] Enright D, Marschner S, Fedkiw R. Animation and Rendering of Complex Water Surfaces[J]. ACM Trans. on Graph, 2002, 21(3): 736-744.
- [7] Enright D, Losasso F, Fedkiw R. A Fast and Accurate Semi-Lagrangian Particle Level Set Method[J]. Computers and Structures, 2005, 83(6): 479-490.
- [8] 刘 焕, 罗大庸. 基于粒子系统的污染气流扩散仿真[J]. 计算机工程, 2009, 35(12): 254-255.
- [9] 刘耀周, 张锡恩. 基于粒子系统的导弹飞行航迹及烟雾的特效生成[J]. 计算机工程, 2004, 30(1): 174-176.
- [10] Sethian J. A Fast Marching Level Set Method for Monotonically Advancing Fronts[J]. Proceedings of the National Academy of Sciences, 1996, 93(4): 1591-1595.
- [11] Hu Wei, Qin Kaihuai, Wang Huawei, et al. Reflection and Refraction on Implicit Surfaces[J]. Journal of Computer Science and Technology, 2006, 21(2): 166-172.
- [12] Brochu T, Bridson R. Robust Topological Operations for Dynamic Explicit Surfaces[J]. SIAM Journal on Scientific Computing, 2009, 31(4): 2472-2493.
- [13] Enright D, Nguyen D, Gibou F, et al. Using the Particle Level Set Method and a Second Order Accurate Pressure Boundary Condition for Free Surface Flows[C]//Proc. of the 4th ASME/JSME Joint Fluids Conference. Hamamatsu, Japan: [s. n.], 2003.
- [14] Pharr M, Humphreys G. Physically Based Rendering: From Theory to Implementation[M]. [S. l.]: Morgan Kaufmann, 2004.

编辑 顾姣健