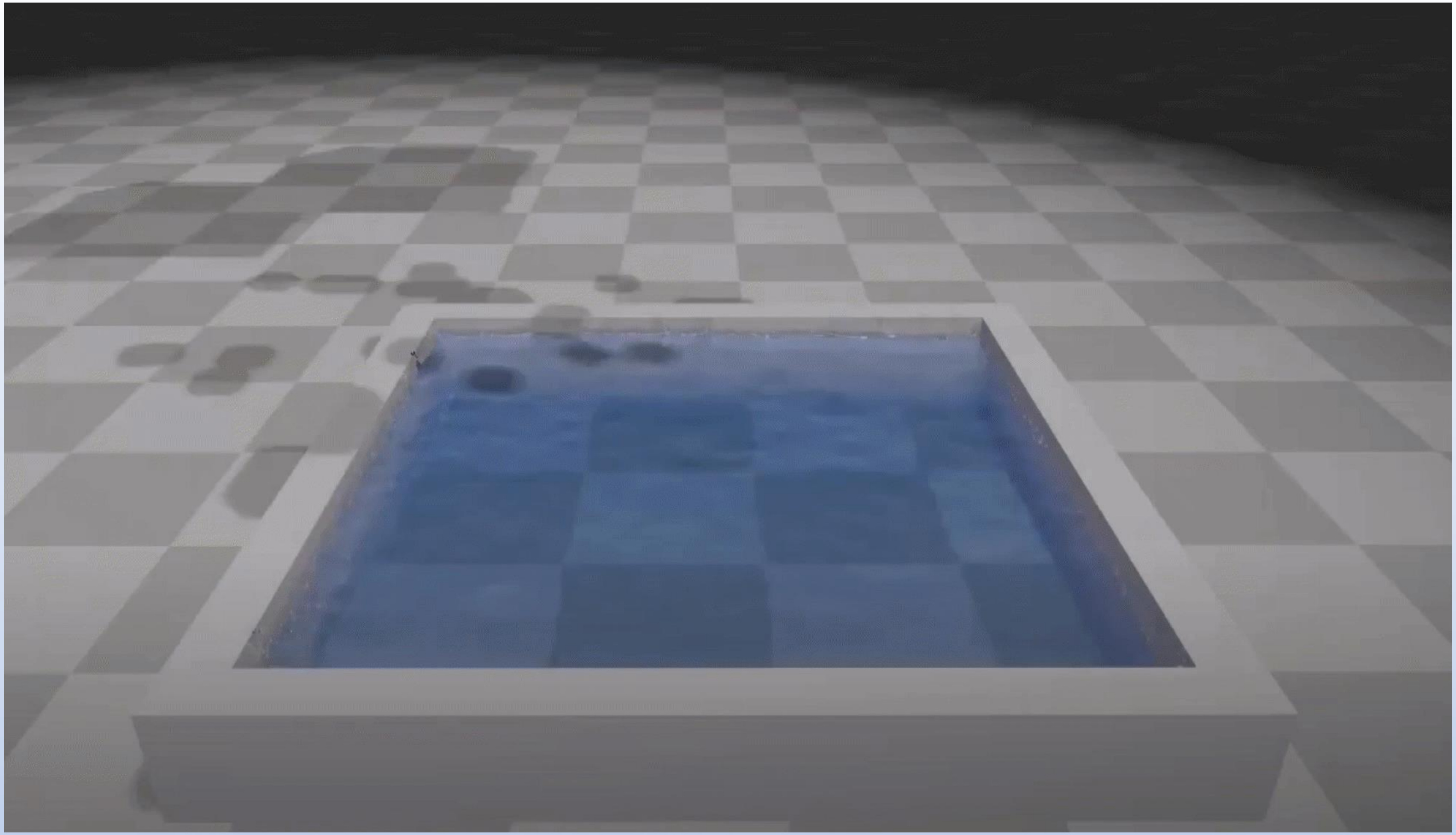
基于物理的流体模拟入门

公共技术组

Flex Demo



目录

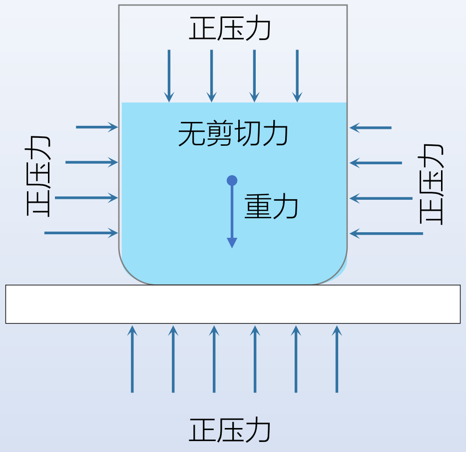
- 流体基础
- 基于力的流体模拟
- 基于约束的流体模拟

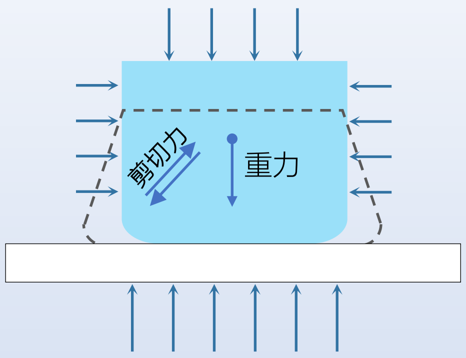
目录

- 流体基础
- 基于力的流体模拟
- 基于约束的流体模拟

认识流体

- 液体和气体等易于流动的物质的统称
  - 不断变形的运动
  - 受力时的运动状态





- 严格定义
  - 在任意小的剪切力作用下都会发生连续不断的角变形的物质

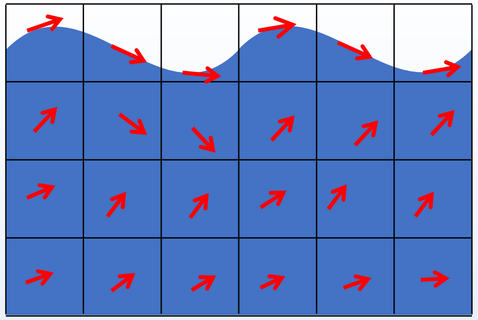
观察流体——两种不同的视角

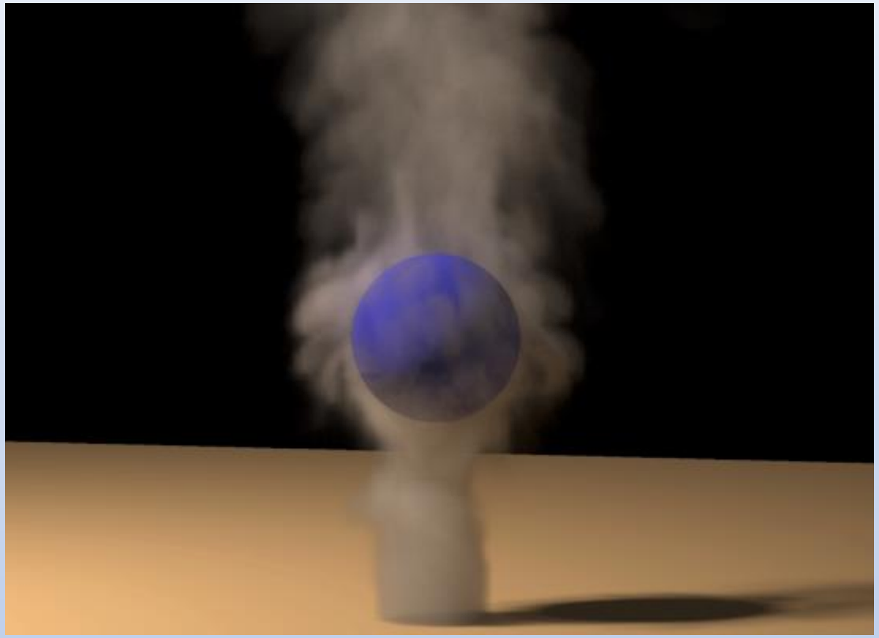
➢欧拉视角

固定位置

流体流过时测量

“岿然不动”



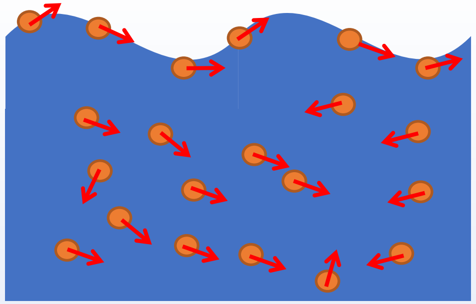


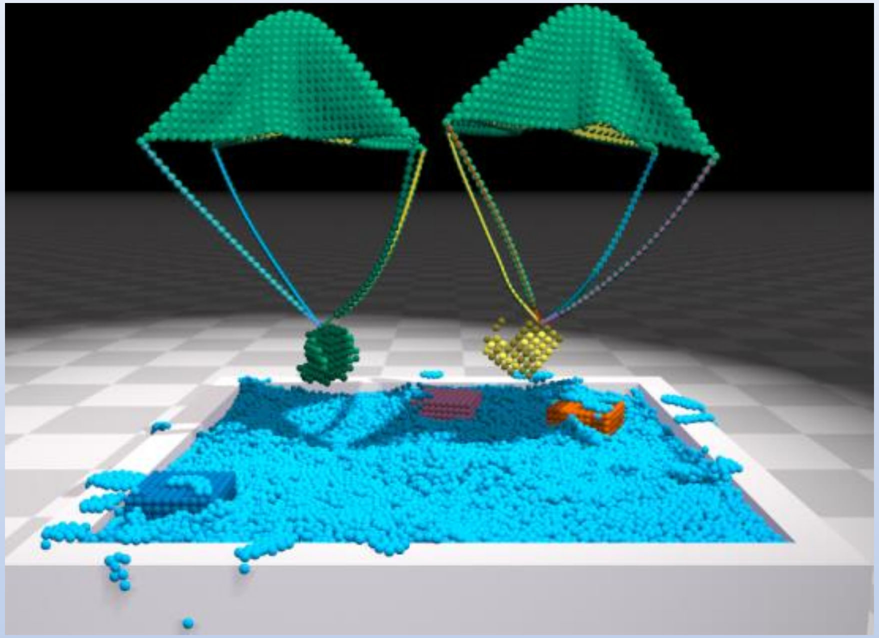
➢拉格朗日视角

运动的粒子

粒子携带物理量

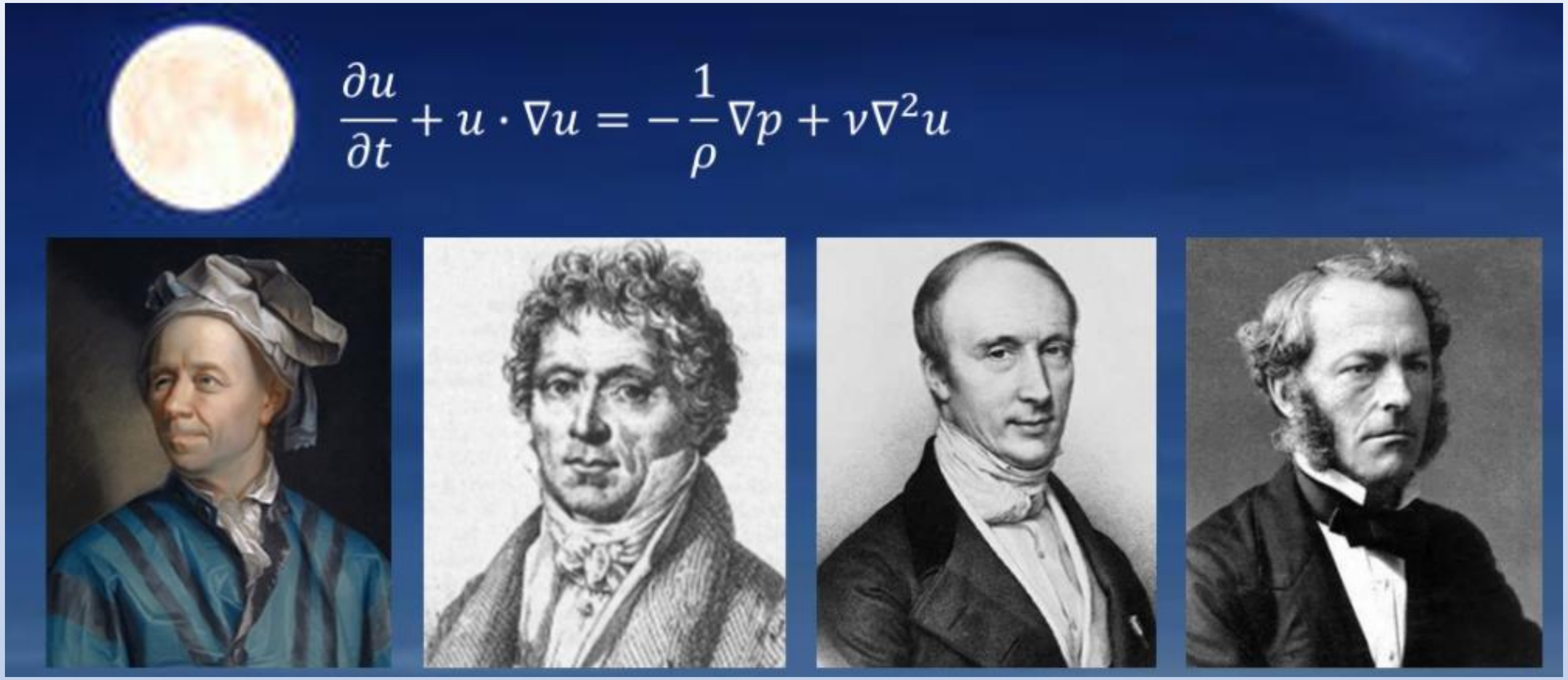
“随波逐流”





流体力学的”白月光”——NS方程

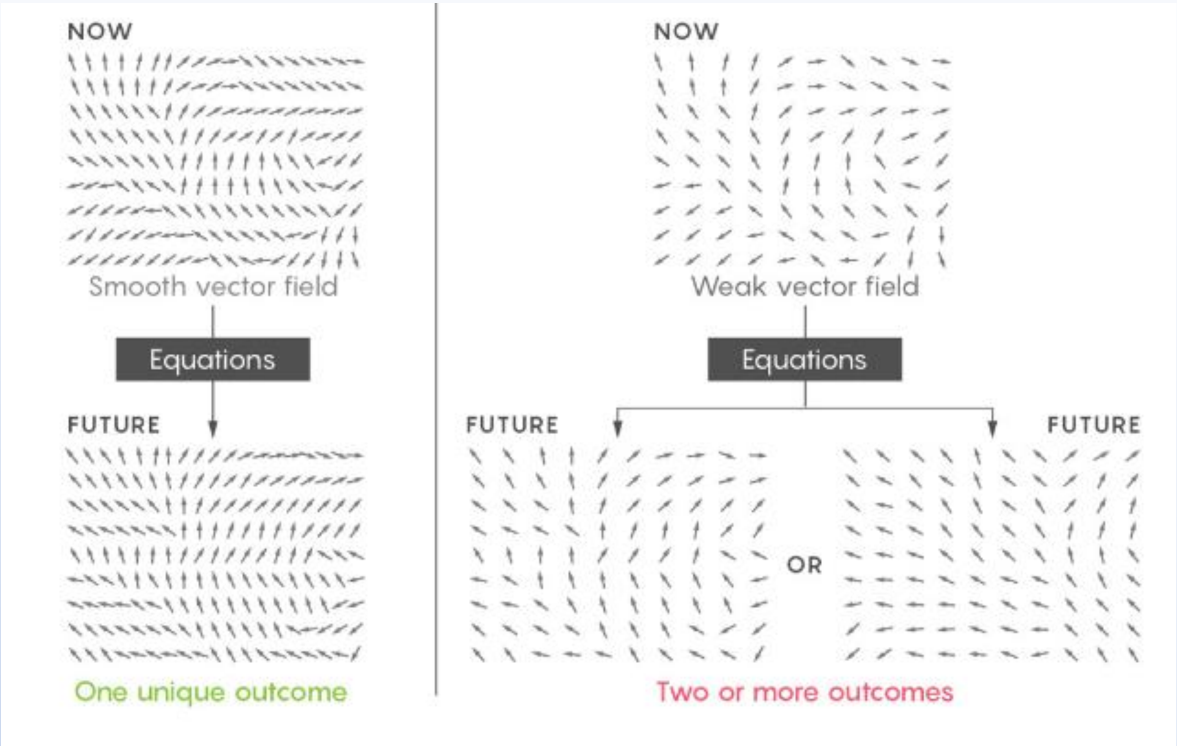
- ➢纳维-斯托克斯方程(Navier-Stokes equations)
- ➢适用于可压缩变粘度的粘性流体的运动
- ➢最普适的流体运动方程



流体力学的”白月光”——NS方程

- 千禧年七大数学难题之一
  - 光滑的速度场
  - 所有时刻的所有起点都有解
- 光滑解
  - 物理世界的完整写照
  - 最大化信息量
  - 要求在与流体相关的向量场内，每个点都存在一个向量
- “弱解”
  - 只需要能够计算某些点上的向量
  - 只需对向量的计算进行估算

- 光滑解 VS “弱”解



向量微分基础

- Nabla算子
- 梯度（Gradient）
- 散度（Divergence）
- 旋度（Curl）
- 拉普拉斯算子（ Lapacian ）

Nabla算子

- 向量微分算子

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$$

- 直接作用于函数F（标量或非标量）

$$\nabla F(\text{梯度})$$

- 与非标量函数F作点乘

$$\nabla \cdot F(\text{散度})$$

- 与非标量函数F作叉乘

$$\nabla \times F(\text{旋度})$$

梯度 (Gradient)

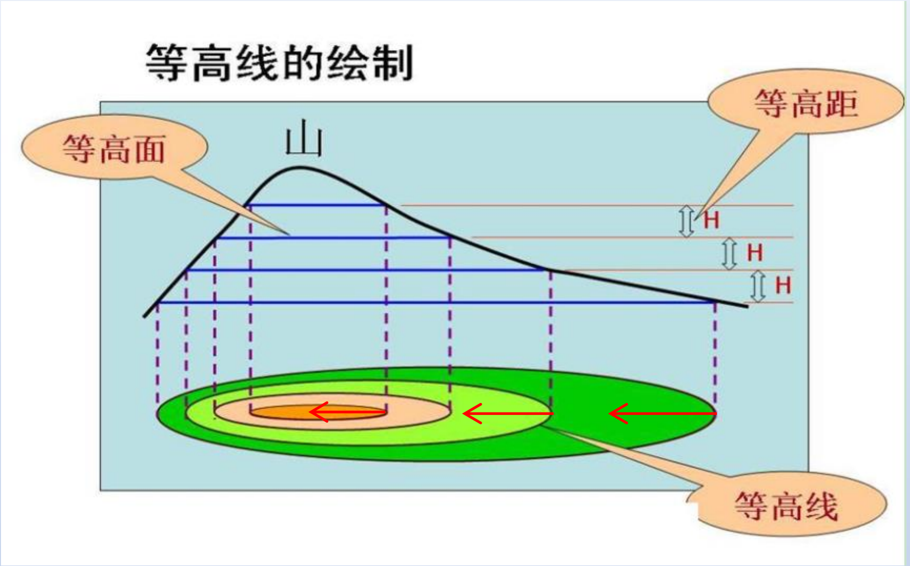
- 函数$u = f(x, y, z)$的梯度定义

$$grad\ f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) = \nabla f$$

- 沿梯度方向的方向导数最大

  （函数值增加最快）

- 梯度向量和等值曲面$f(x, y, z) = C$ 垂直

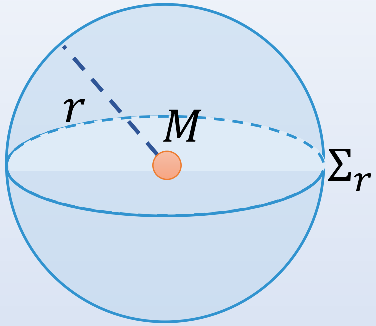


散度 (Divergence)

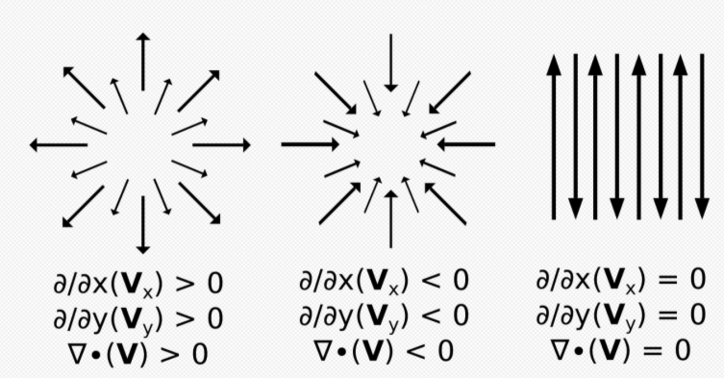
- 对于向量场$\boldsymbol{A}=(P(x,y,z),Q(x,y,z),R(x,y,z))$，称

$$div\ \boldsymbol{A}=\frac{\partial P}{\partial x}+\frac{\partial Q}{\partial y}+\frac{\partial R}{\partial z}=\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)\cdot(P,Q,R)=\nabla\cdot A$$

- 向量场A在点M处的通量密度
  - $\lim\limits_{r\to 0}\frac{\iint_{\Sigma_r}A\cdot dS}{V}=div\ A(M)$ （单位体积的通量）



- 如果向量场A处处有$div\ A=0$，则称A为无源场
- 高斯散度定理
  - $\oiint_{\partial V}\boldsymbol{A}d\boldsymbol{S}=\iiint_V \mathrm{div}\ \boldsymbol{A}\ \mathrm{d}V$

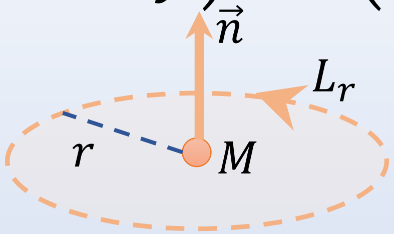


旋度 (Curl)

- 对于向量场$\boldsymbol{A}=(P(x,y,z),Q(x,y,z),R(x,y,z))$，其旋度定义为

$$\operatorname{curl}\boldsymbol{A}=\left(\frac{\partial R}{\partial y}-\frac{\partial Q}{\partial z},\frac{\partial P}{\partial z}-\frac{\partial R}{\partial x},\frac{\partial Q}{\partial x}-\frac{\partial P}{\partial y}\right)=\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)\times(P,Q,R)$$

$$=\nabla\times\boldsymbol{A}$$



- 绕单位向量$\vec{n}$的环流量密度
  - $\lim\limits_{r\to 0}\frac{\oint_{L_r}A\cdot dl}{S}=(curl\ \boldsymbol{A}(M))\cdot\vec{n}$ （单位面积的环流量）
- 向量场绕旋度的环流量密度最大
- 若向量场$\boldsymbol{A}$处处有$\operatorname{curl}\boldsymbol{A}=0$，则称$\boldsymbol{A}$为无旋场

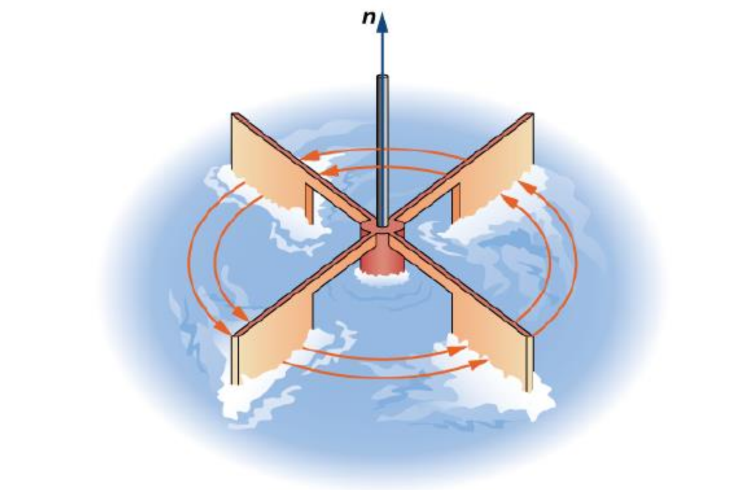


Figure 6.54 To visualize curl at a point, imagine placing a small paddlewheel into the vector field at a point.

拉普拉斯算子(Laplacian)

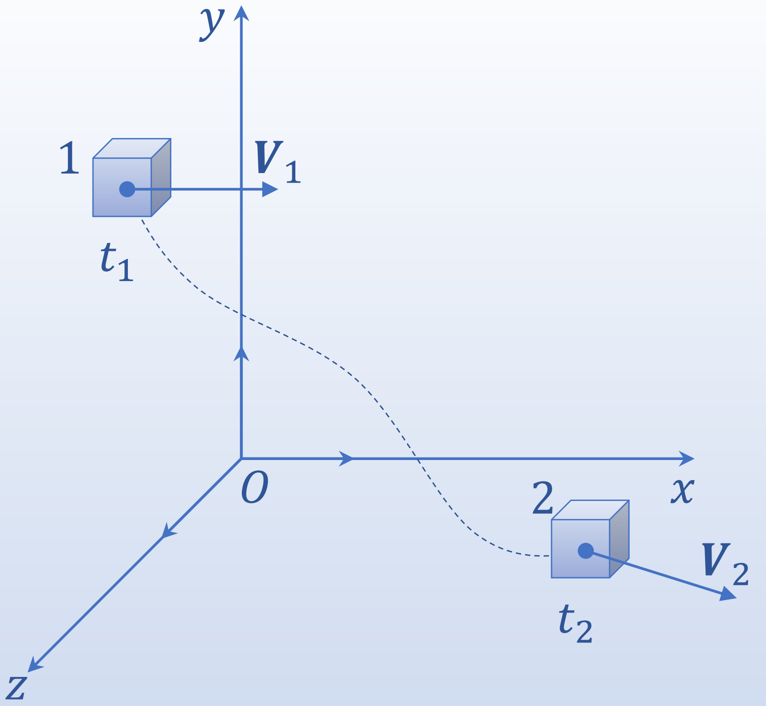
- 空间标量函数$u = f(x, y, z)$，称

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right) \cdot \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}\right) = \nabla \cdot \nabla u = \nabla^2 u$$

- 二阶微分算子
- 相当于对梯度场求散度
- 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(material derivative)

- 针对的是流体微团，而不是空间的固定点
- 标量函数$Q(x,y,z,t)$，速度场$\boldsymbol{V}(u,v,w)$
- $\frac{\mathrm{D}Q}{Dt}=\frac{\partial Q}{\partial t}+u\frac{\partial Q}{\partial x}+v\frac{\partial Q}{\partial y}+w\frac{\partial Q}{\partial z}=\frac{\partial Q}{\partial t}+\boldsymbol{V}\cdot\nabla Q$
- $\frac{\mathrm{D}}{Dt}\equiv\frac{\partial}{\partial t}$(当地导数) $+\boldsymbol{V}\cdot\nabla$(迁移导数)
- 对时间的全导数
  - $dQ=\frac{\partial Q}{\partial t}dt+\frac{\partial Q}{\partial x}dx+\frac{\partial Q}{\partial y}dy+\frac{\partial Q}{\partial z}dz$
  - $\frac{dQ}{dt}=\frac{\partial Q}{\partial t}+\frac{\partial Q}{\partial x}\frac{dx}{dt}+\frac{\partial Q}{\partial y}\frac{dy}{dt}+\frac{\partial Q}{\partial z}\frac{dz}{dt}$
  - $\frac{dQ}{dt}=\frac{\partial Q}{\partial t}+\frac{\partial Q}{\partial x}u+\frac{\partial Q}{\partial y}v+\frac{\partial Q}{\partial z}w$



目录

- 流体基础
- 基于力的流体模拟
  - NS方程的推导
  - NS方程求解
- 基于约束的流体模拟

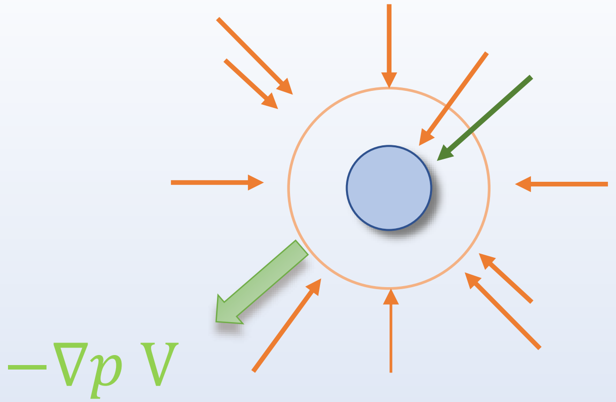
NS 方程

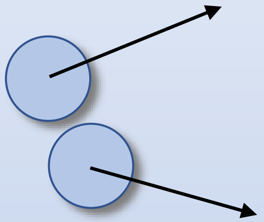
- 动量方程
  - $\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + v \Delta \vec{u} + \vec{g}$
- 质量守恒方程
  - $\nabla \cdot \vec{u} = 0$

- $\vec{u}$（速度）
- p（压力）
- ρ（密度）
- $\vec{g}$（重力）
- v（动力粘性系数）
- ∇（梯度算子）
- Δ（拉普拉斯算子）

NS 方程——动量方程

- 牛顿第二定律：$\vec{F} = m\vec{a} = m\frac{D\vec{u}}{Dt}$
- 受力分析
  - 重力：$m\vec{g}$
  - 压力：$-\nabla p\ \mathrm{V}$
  - 黏力：$\mathrm{V}\mu\nabla\cdot\nabla\vec{u} = \mathrm{V}\mu\nabla^2\vec{u}$
- $m\frac{D\vec{u}}{Dt} = m\vec{g} - \nabla p\ \mathrm{V} + \mathrm{V}\mu\nabla^2\vec{u}$
- $\frac{D\vec{u}}{Dt} = \vec{g} - \frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2\vec{u}$
- $\boxed{\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + v\nabla^2\vec{u} + \vec{g}}$ （动力粘性系数）





Diffusion of (relative) velocities

NS 方程——动量方程

- 拉格朗日视角

$$\boxed{\frac{D\vec{u}}{Dt}} = -\frac{1}{\rho}\nabla p + v\nabla^2\vec{u} + \vec{g}$$

$$\underbrace{\frac{Dq}{Dt}}_{\text{物质导数/随体导数}} = \underbrace{\frac{\partial q}{\partial t}}_{\text{当地导数}} + \underbrace{\vec{u}\cdot\nabla q}_{\text{迁移导数}}$$

- 欧拉视角

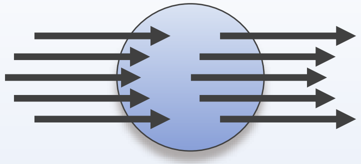
$$\boxed{\frac{\partial\vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}} = -\frac{1}{\rho}\nabla p + v\nabla^2\vec{u} + \vec{g}$$

$$\frac{\partial\vec{u}}{\partial t} = -\vec{u}\cdot\nabla\vec{u} - \frac{1}{\rho}\nabla p + v\nabla^2\vec{u} + \vec{g}$$

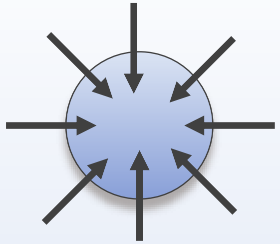
NS 方程——质量守恒方程

- 不可压缩性：体积和密度均为常数
- 体积V，边界曲面为∂V，其体积变化率为

$$\oiint_{\partial V} \vec{u} \cdot \vec{n}\, \mathrm{d}S = \iiint_V \nabla \cdot \vec{u}\, \mathrm{d}V$$

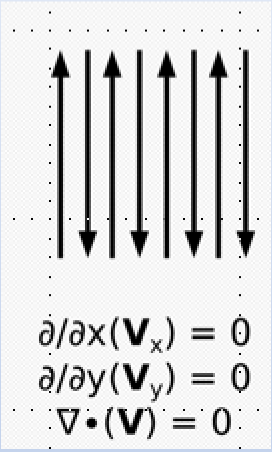


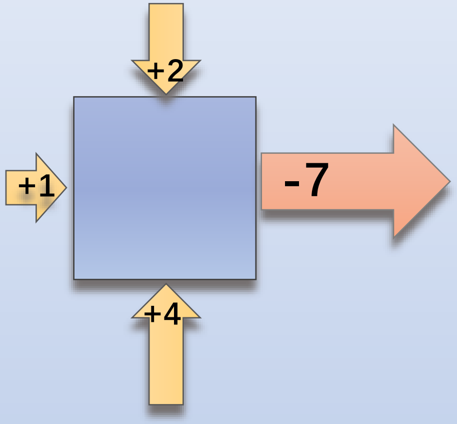
不可压缩



可压缩

- $\iiint_V \nabla \cdot \vec{u}\, \mathrm{d}V = 0$
- 体积$V \neq 0$ → $\nabla \cdot \vec{u} = 0$
- 速度场无散度（有进必有出）



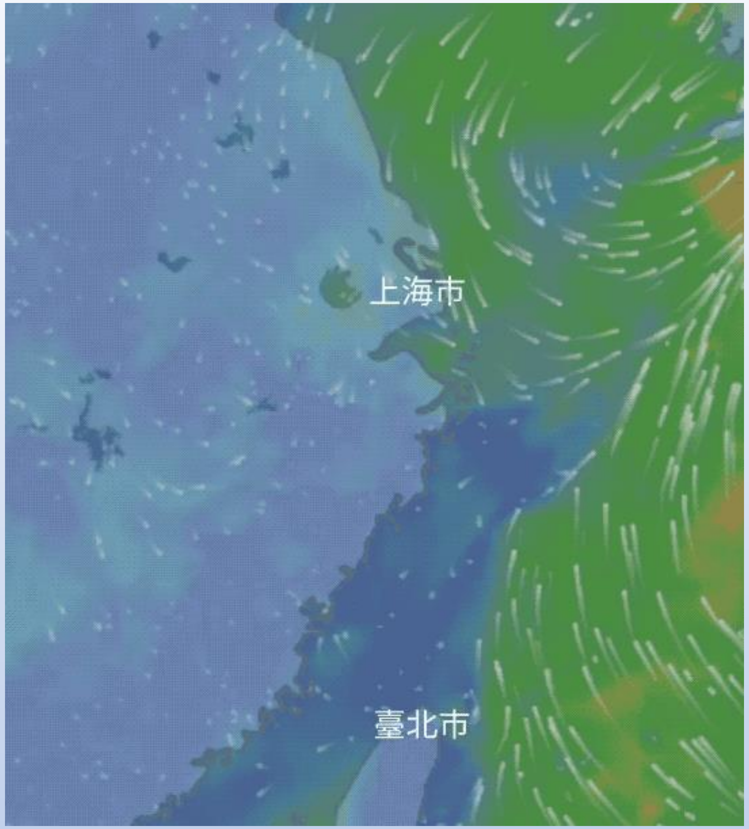


目录

- 流体基础
- 基于力的流体模拟
  - NS方程的推导
  - NS方程求解
- 基于约束的流体模拟

NS方程的求解

- $\frac{\partial \vec{u}}{\partial t} = -\vec{u} \cdot \nabla \vec{u} - \frac{1}{\rho} \nabla p + \cancel{v \nabla^2 \vec{u}} + \vec{g}$ (欧拉方程)
- $\nabla \cdot \vec{u} = 0$
- 目标：求流体的速度场$\vec{u}$



目录

- 流体基础
- 基于力的流体模拟
  - NS方程的推导
  - NS方程求解
    - 基于欧拉视角求解NS方程
    - 基于拉格朗日视角求解NS方程
    - 欧拉网格和拉格朗日粒子法的比较
- 基于约束的流体模拟

NS方程的求解——离散化

- MAC(marker and cell)
- 交叉排列的网格
- 不同类型的物理量存储于网格的不同位置
- 平均法

$$\vec{u}_{i,j}=\left(\frac{u_{i-1/2,j}+u_{i+1/2,j}}{2},\frac{v_{i,j-1/2}+v_{i,j+1/2}}{2}\right)$$

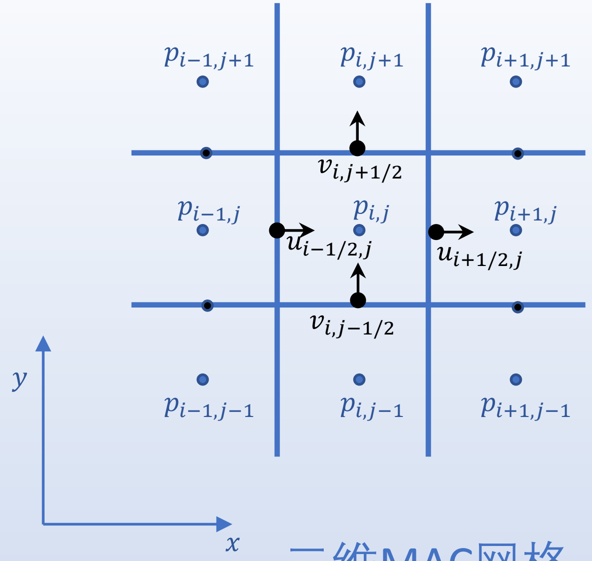
$$\vec{u}_{i+1/2,j}=\left(u_{i+1/2,j},\frac{v_{i,j-1/2}+v_{i,j+1/2}+v_{i+1,j-1/2}+v_{i+1,j+1/2}}{4}\right)$$

$$\vec{u}_{i,j+1/2}=\left(\frac{u_{i-1/2,j}+u_{i+1/2,j}+u_{i-1/2,j+1}+u_{i+1/2,j+1}}{4},v_{i,j+1/2}\right)$$

- 中心差分法

$$\left(\frac{\partial q}{\partial x}\right)_i\approx\frac{q_{i+1}-q_{i-1}}{2\Delta x}$$

$$\left(\frac{\partial q}{\partial x}\right)_i\approx\frac{q_{i+1/2}-q_{i-1/2}}{\Delta x}$$



二维MAC网格

NS方程的求解——分步(split)求解思想

$$\frac{\partial \vec{u}}{\partial t} = -\vec{u}\cdot\nabla\vec{u} - \frac{1}{\rho}\nabla p + v\nabla^2\vec{u} + \vec{g}$$

- $\dfrac{dq}{dt} = f(q) + g(q)$
- 直接用前向欧拉法:
  - $q^{n+1} = q^n + \Delta t(f(q^n) + g(q^n))$
- 分步求法:
  - $q^{*} = q^n + \Delta t\, f(q^n)$
  - $q^{n+1} = q^{*} + \Delta t\, g(q^{*})$

$$\begin{aligned}
q^{n+1} &= \left(q^n + \Delta t f(q^n)\right) + \Delta t\, g\left(q^n + \Delta t\, f(q^n)\right) \\
&= q^n + \Delta t f(q^n) + \Delta t\left(g(q^n) + O(\Delta t)\right) \\
&= q^n + \Delta t\left(f(q^n) + g(q^n)\right) + O(\Delta t^2) \\
&= q^n + \frac{dq}{dt}\Delta t + O(\Delta t^2)
\end{aligned}$$

NS方程的求解——分步求解

动量方程：$\frac{\partial \vec{u}}{\partial t} = -\vec{u} \cdot \nabla \vec{u} - \frac{1}{\rho} \nabla p + \cancel{\nu \nabla^2 \vec{u}} + \vec{g}$

质量守恒方程：$\nabla \cdot \vec{u} = 0$

split

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q = 0$$ (对流)

$$\frac{\partial \vec{u}}{\partial t} = \vec{g}$$ (体积力)

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho} \nabla p = 0$$ (压力/不可压缩)

$$s.t.\ \nabla \cdot \vec{u} = 0$$

初始化**无散度**速度场$\vec{u}_n$

对于每个时间步$n = 0,1,2, \ldots$

- 决定一个合理的时间步长$\Delta t = t_{n+1} - t_n$
- 计算对流项$\vec{u}_A = advect(\vec{u}_n, \Delta t, \vec{q})$
- 计算体积力项$\vec{u}_B = \vec{u}_A + \Delta t \vec{g}$
- 无散度投影$\vec{u}_{n+1} = project(\Delta t, \vec{u}_B)$

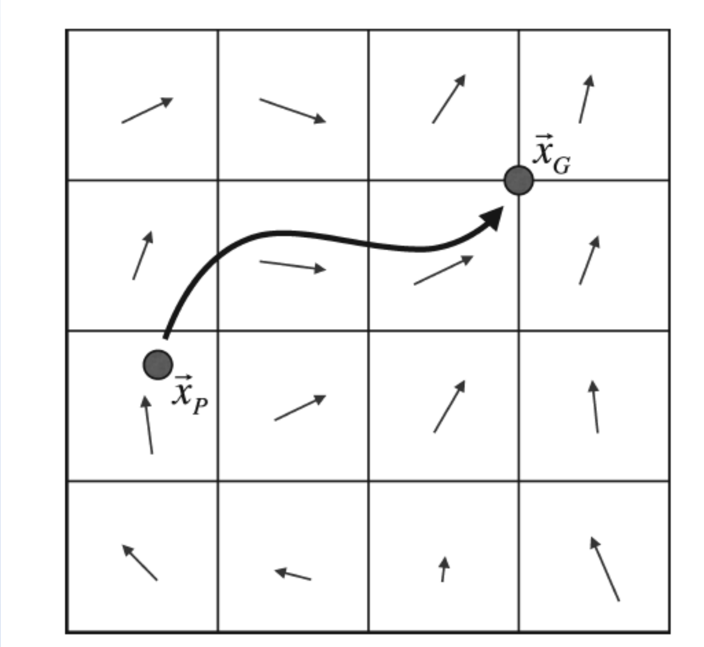
对流 Advection → 体积力 Body Force → 压力 Pressure

NS方程的分步求解——对流

- 基于粒子的方法不需要做对流
- 半拉格朗日对流法
- 目标：求G点在第n+1个时间步长的物理量q_G^{n+1}
- 相当于P点在第n个时间步长的物理量: q_P^n
- $\frac{d\vec{x}}{dt} = \vec{u}(\vec{x})$
- $\frac{\vec{x}_G - \vec{x}_P}{\Delta t} = \vec{u}(\vec{x}_G)$
- $\vec{x}_P = \vec{x}_G - \Delta t \vec{u}(\vec{x}_G)$

$advect(\vec{u}_n, \Delta t, \vec{q})$

$\frac{Dq}{Dt}=0$



NS方程的分步求解——压力/不可压缩投影

$\vec{u}_{n+1} = project(\Delta t, \vec{u}_B)$

- $\dfrac{\partial \vec{u}}{\partial t} + \dfrac{1}{\rho}\nabla p = 0$

  $s.t.\ \nabla \cdot \vec{u} = 0$
- 目标：最终算出的速度$\vec{u}^{n+1}$是无散度的
  - 用压力梯度∇p更新下一个时间步长的速度$\vec{u}^{n+1}$
  - 将$\vec{u}^{n+1}$代入无散度公式$\nabla \cdot \vec{u} = 0$
  - 求出满足速度场无散度的压力场p
  - 用上一步求出的压力场p再一次更新速度$\vec{u}^{n+1}$

NS方程的分步求解——压力/不可压缩投影

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0$$

$$s.t.\ \nabla \cdot \vec{u} = 0$$

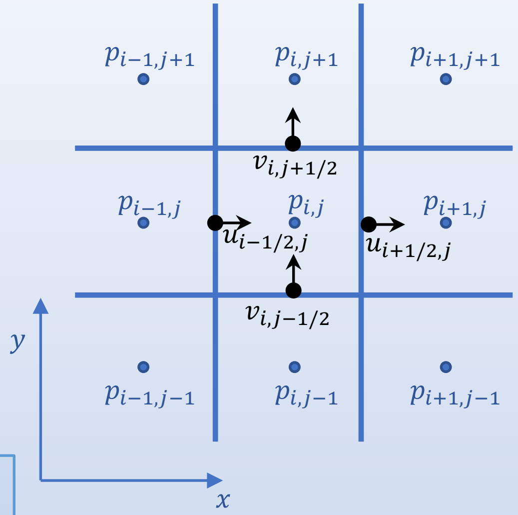
- 压力梯度离散化(The Discrete Pressure Gradient)：

$$\nabla p = \left(\frac{p_{i+1,j} - p_{i,j}}{\Delta x}, \frac{p_{i,j+1} - p_{i,j}}{\Delta x}\right)$$

- 速度散度离散化(The Discrete Divergence)

$$\nabla \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \approx \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta x} = 0$$

$$\boxed{\frac{u^{n+1}_{i+1/2,j} - u^{n+1}_{i-1/2,j}}{\Delta x} + \frac{v^{n+1}_{i,j+1/2} - v^{n+1}_{i,j-1/2}}{\Delta x} = 0}$$



二维MAC网格

NS方程的分步求解——压力/不可压缩投影

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0$$

$s.t.\ \nabla \cdot \vec{u} = 0$

$$\frac{\vec{u}^{n+1} - \vec{u}^{n}}{\Delta t} + \frac{1}{\rho}\nabla p = 0$$

$$u_{i+1/2,j}^{n+1} = u_{i+1/2,j} - \Delta t \frac{1}{\rho}\frac{p_{i+1,j} - p_{i,j}}{\Delta x}$$

$$u_{i-1/2,j}^{n+1} = u_{i-1/2,j} - \Delta t \frac{1}{\rho}\frac{p_{i-1,j} - p_{i,j}}{\Delta x}$$

$$v_{i,j+1/2}^{n+1} = v_{i,j+1/2} - \Delta t \frac{1}{\rho}\frac{p_{i,j+1} - p_{i,j}}{\Delta x}$$

$$v_{i,j-1/2}^{n+1} = v_{i,j-1/2} - \Delta t \frac{1}{\rho}\frac{p_{i,j-1} - p_{i,j}}{\Delta x}$$

- 压力梯度：$\nabla p = \left(\frac{p_{i+1,j} - p_{i,j}}{\Delta x}, \frac{p_{i,j+1} - p_{i,j}}{\Delta x}\right)$
- 速度散度：$\frac{u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1}}{\Delta x} + \frac{v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1}}{\Delta x} = 0$
- 压力方程(The Pressure Equations)

$$\frac{\Delta t}{\rho}\left(\frac{4p_{i,j} - p_{i+1,j} - p_{i,j+1} - p_{i-1,j} - p_{i,j-1}}{\Delta x^2}\right) = -\left(\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta x}\right)$$

$-\frac{\Delta t}{\rho}\nabla \cdot \nabla p = -\nabla \cdot \vec{u}$（泊松方程）

NS方程的分步求解——压力/不可压缩投影

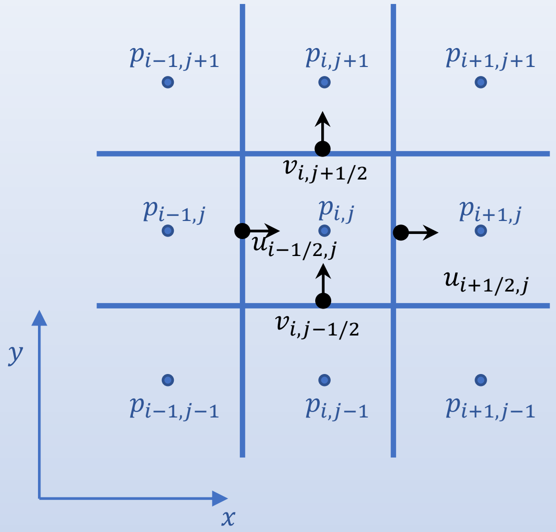
$$\frac{\Delta t}{\rho}\left(\frac{4p_{i,j}-p_{i+1,j}-p_{i,j+1}-p_{i-1,j}-p_{i,j-1}}{\Delta x^2}\right)=-\left(\frac{u_{i+1/2,j}-u_{i-1/2,j}}{\Delta x}+\frac{v_{i,j+1/2}-v_{i,j-1/2}}{\Delta x}\right)$$

$$-\frac{\Delta t}{\rho}\nabla\cdot\nabla p=-\nabla\cdot\vec{u}$$ （泊松方程）

$$\boldsymbol{Ap}=\boldsymbol{b}$$

$$\underbrace{\begin{bmatrix} a_{0,0} & \cdots & a_{0,m*n} \\ \vdots & \ddots & \vdots \\ a_{m*n,0} & \cdots & a_{m*n,m*n} \end{bmatrix}}_{O(m^2n^2)} \underbrace{\begin{bmatrix} p_{0,0} \\ \vdots \\ p_{m,n} \end{bmatrix}}_{O(m\times n)} = \underbrace{\begin{bmatrix} \nabla\cdot\vec{u}_{0,0} \\ \vdots \\ \nabla\cdot\vec{u}_{m,n} \end{bmatrix}}_{O(m\times n)}$$

$$O(m^2n^2)$$



二维MAC网格

NS方程的求解

$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q = 0$(对流)

$\frac{\partial \vec{u}}{\partial t} = \vec{g}$ (体积力)

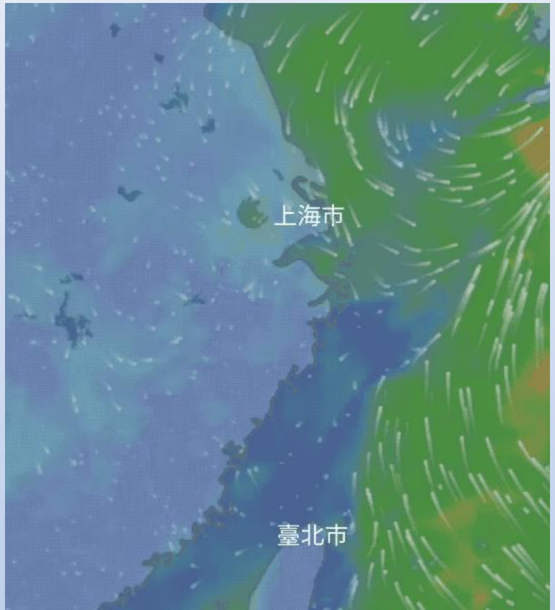
$\begin{cases} \frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho} \nabla p = 0 \\ \nabla \cdot \vec{u} = 0 \end{cases}$(压力/不可压缩)

初始化**无散度**速度场$\vec{u}_n$

对于每个时间步$n = 0,1,2,\ldots$

- 决定一个合理的时间步长$\Delta t = t_{n+1} - t_n$
- 计算对流项$\vec{u}_A = advect(\vec{u}_n, \Delta t, \vec{q})$
- 计算体积力项$\vec{u}_B = \vec{u}_A + \Delta t \vec{g}$
- 无散度投影$\vec{u}_{n+1} = project(\Delta t, \vec{u}_B)$

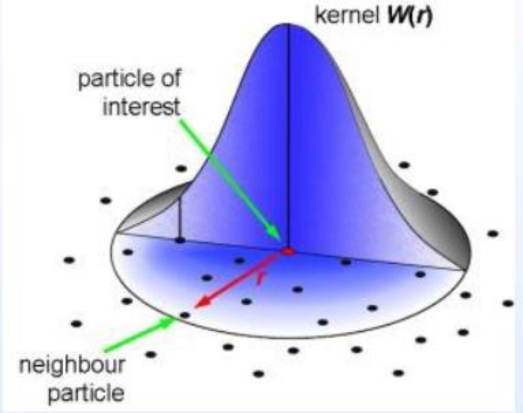
对流 Advection → 体积力 Body Force → 压力 Pressure

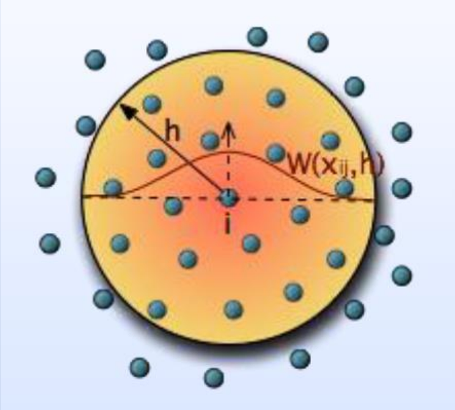


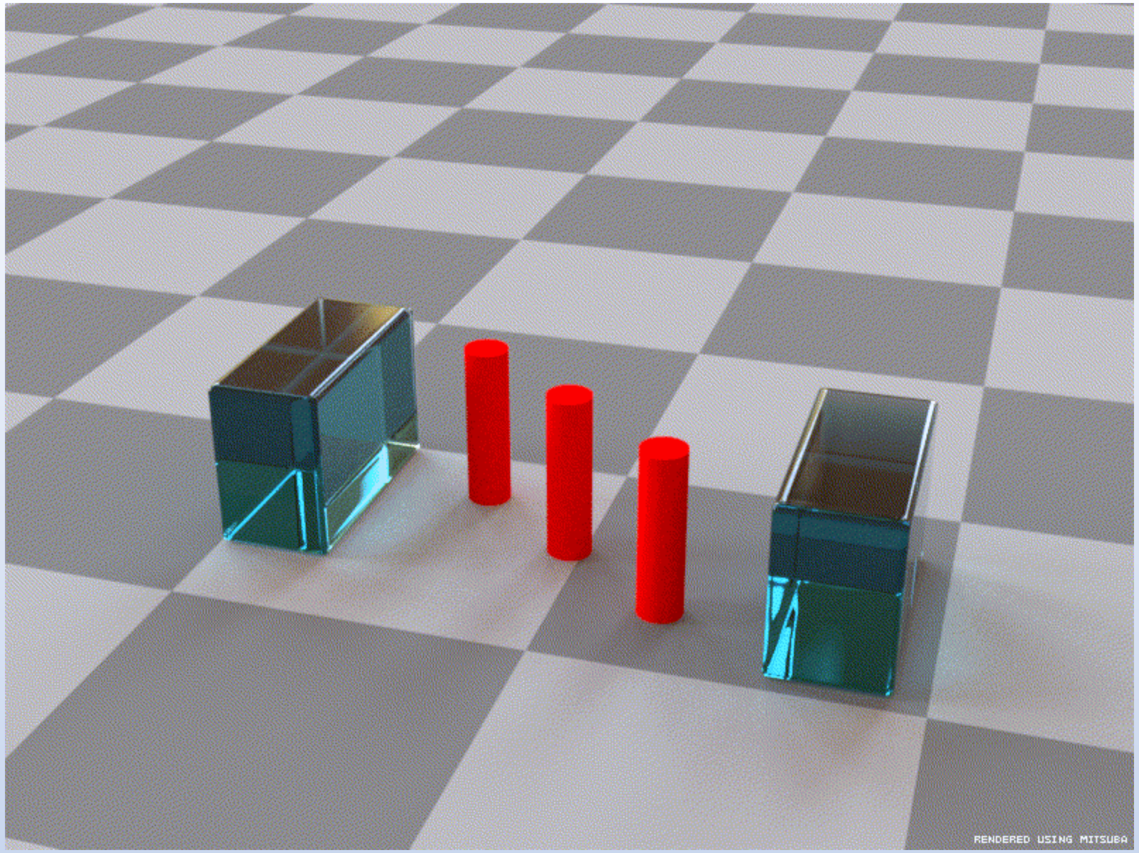
目录

一种近似求解NS方程的方法 —— SPH

- Smooth Particle Hydrodynamics
- 核密度估计
  - 每个粒子代表一定的流体体积$V_i = \frac{m_i}{\rho_i}$
  - 属性存储在粒子上
    - 由其邻域粒子的属性值的加权平均决定
    - 采用平滑核函数W来对权重进行插值

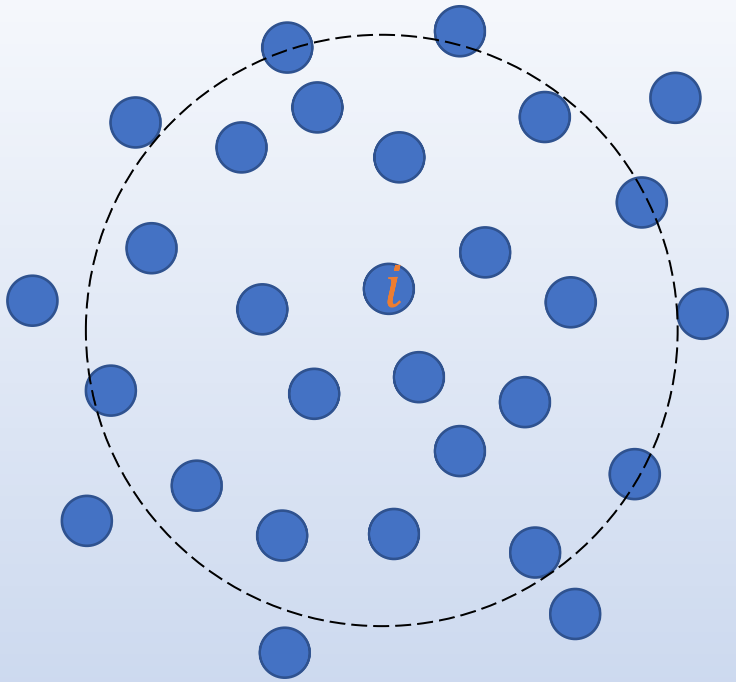






SPH

- 空间中任意位置$\vec{x}_i$的物理量A
- $A_i = \sum_j \frac{m_j}{\rho_j} A_j W(\vec{x}_i - \vec{x}_j, h)$
- $A_i = \sum_j \frac{m_j}{\rho_j} A_j W_{ij}$(简化写法)
- $\nabla A_i = \sum_j \frac{m_j}{\rho_j} A_j \nabla W_{ij}$
- $\nabla^2 A_i = \sum_j \frac{m_j}{\rho_j} A_j \nabla^2 W_{ij}$



SPH

- $\frac{\partial \vec{u}}{\partial t} + \cancel{\vec{u} \cdot \nabla \vec{u}} = -\frac{1}{\rho}\nabla p + v\nabla^2 \vec{u} + \vec{g}$

$\vec{a} = \frac{\vec{F}}{m}$

- $\cancel{\nabla \cdot \vec{u} = 0}$

两边同时乘以ρ

- $\rho \frac{\partial \vec{u}}{\partial t} = -\nabla p + \mu \nabla^2 \vec{u} + \rho \vec{g}$

$\vec{f} = \vec{F}/\mathrm{V}$

邻域搜索
Neighbor Search

一次全部计算

密度
Density

体积力 / V
Body Force

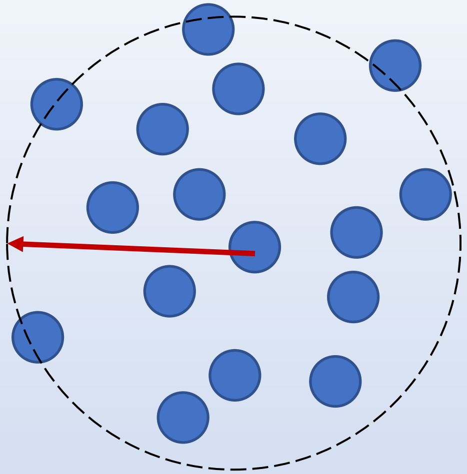
黏力 / V
Viscosity

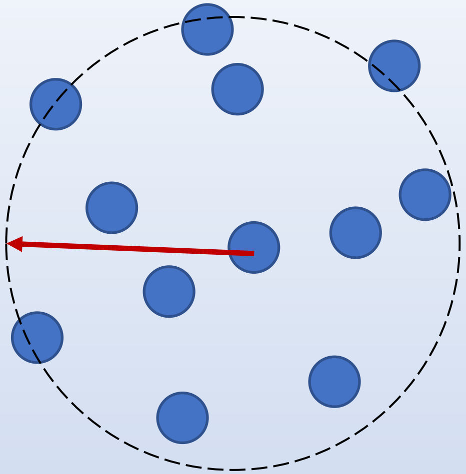
压力 / V
Pressure

SPH——密度

- $A_i = \sum_j \frac{m_j}{\rho_j} A_j W_{ij}$
- $\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W_{ij}$
- $\rho_i = \sum_j m_j W_{ij}$

$$\rho \frac{\partial \vec{u}}{\partial t} = -\nabla p + \mu \nabla^2 \vec{u} + \rho \vec{g}$$



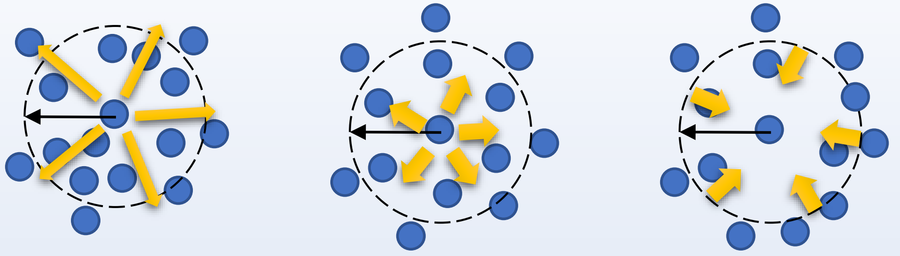


SPH——压力

$$\rho\frac{\partial\vec{u}}{\partial t} = -\nabla p + \mu\nabla^2\vec{u} + \rho\vec{g}$$

- $p_i = k(\rho_i - \rho_0)$
- $p_i = \max(k(\rho_i - \rho_0), 0)$
- $\nabla A_i = \sum_j \frac{m_j}{\rho_j} A_j \nabla W_{ij}$
- $\vec{f}_i^{pressure} = -\sum_j \cancel{\frac{m_j}{\rho_j} p_j \nabla W_{ij}}$

$\rho_0 = 1000$



$\rho_i = 1200$, $p_i = 200\text{k}$

$\rho_i = 1100$, $p_i = 100\text{k}$

$\rho_i = 800$, $p_i = -200\text{k}$

$$\boxed{\vec{f}_i^{pressure} = -\sum_j \frac{m_j}{\rho_j}\frac{p_i + p_j}{2}\nabla W_{ij}}$$

p_1 p_2

$$\vec{f}_1^{pressure} = -\frac{m_2}{\rho_2}p_2\nabla W_{12}$$
$$\vec{f}_2^{pressure} = -\frac{m_1}{\rho_1}p_1\nabla W_{21}$$

$$\vec{F} = \vec{f}V$$

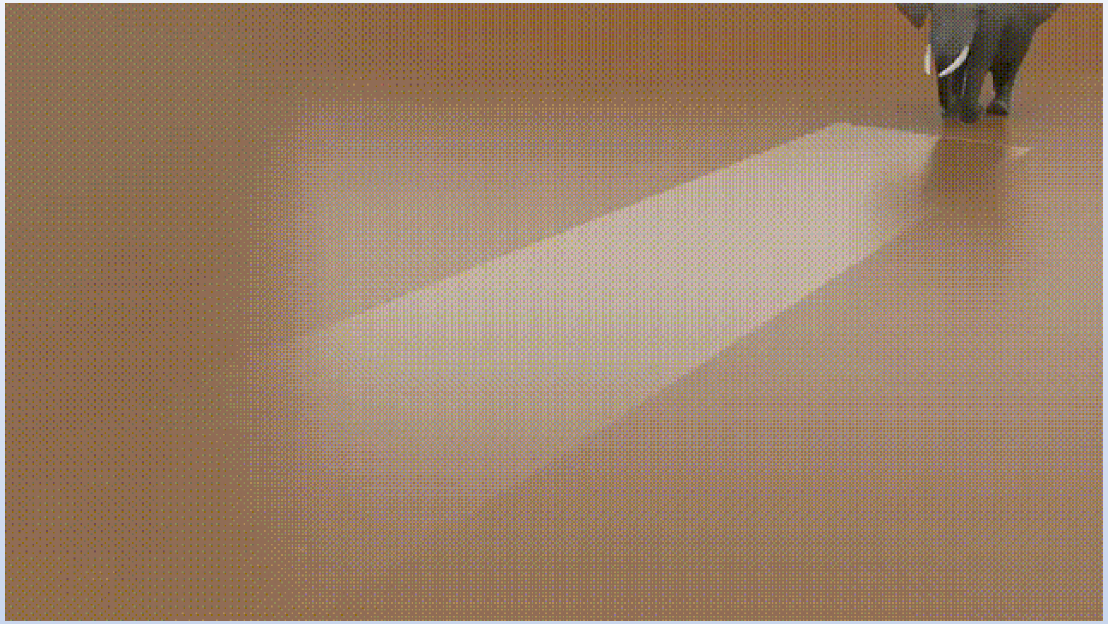
$$\vec{F}_1^{pressure} = -\frac{m_1}{\rho_1}\frac{m_2}{\rho_2}p_2\nabla W_{12}$$
$$\vec{F}_2^{pressure} = -\frac{m_2}{\rho_2}\frac{m_1}{\rho_1}p_1\nabla W_{21}$$

$$\vec{F}_1^{pressure} = -\vec{F}_2^{pressure}$$

SPH——黏力

$$\rho\frac{\partial\vec{u}}{\partial t} = -\nabla p + \mu\nabla^2\vec{u} + \rho\vec{g}$$

- $\vec{f}_i^{viscosity} = \mu\sum_j \cancel{\frac{m_j}{\rho_j}\vec{u}_j\nabla^2} W_{ij}$
- 作用力与反作用力的大小相等
- 只依赖于速度差，不依赖绝对速度
- $\vec{f}_i^{viscosity} = \mu\sum_j \frac{m_j}{\rho_j}(\vec{u}_j - \vec{u}_i)\nabla^2 W_{ij}$



SPH——体积力

- $\vec{f}_i^{gravity} = \rho_i \vec{g}$

$$\rho \frac{\partial \vec{u}}{\partial t} = -\nabla p + \mu \nabla^2 \vec{u} + \rho \vec{g}$$

SPH——速度和位移

- 合力/N： $\vec{f}_i = \vec{f}_i^{pressure} + \vec{f}_i^{viscosity} + \vec{f}_i^{gravity}$
- 加速度： $\vec{a} = \frac{\vec{f}_i}{\rho_i}$
- 速度： $\vec{u}_i(t+1) = \vec{u}_i(t) + \Delta t \frac{\vec{f}_i}{\rho_i}$
- 位置： $\vec{x}_i(t+1) = \vec{x}_i(t) + \Delta t \vec{u}_i(t+1)$

SPH——算法

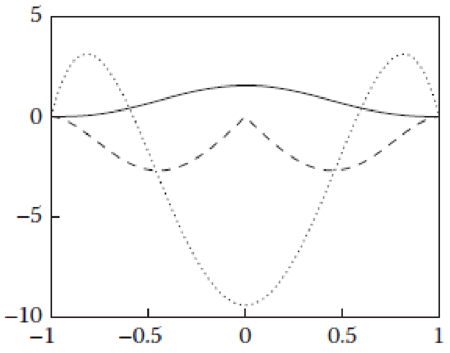
伪代码：

```
while animating do
    for all i do
        find neighborhoods N_i(t)
    for all i do
        compute density ρ_i(t)
        compute density p_i(t)
    for all i do
        compute forces F^{p,v,g,ext}(t)
    for all i do
        compute new velocity u_i(t)
        compute new position x_i(t)
```

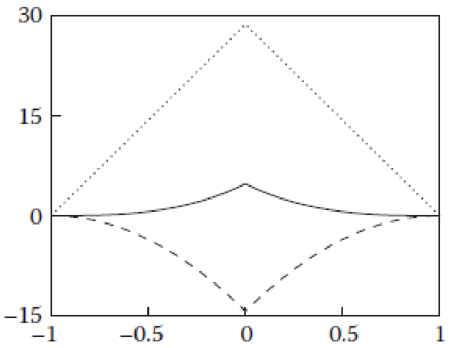
- $\rho_i = \sum_j m_j W_{ij}$
- $p_i = k(\rho_i - \rho_0)$
- $\vec{f}_i^{pressure} = -\sum_j \frac{m_j}{\rho_j} \frac{p_i + p_j}{2} \nabla W_{ij}$
- $\vec{f}_i^{viscosity} = \mu \sum_j \frac{m_j}{\rho_j} (\vec{u}_j - \vec{u}_i) \nabla^2 W_{ij}$
- $\vec{f}_i^{gravity} = \rho_i \vec{g}$
- $\vec{u}_i(t+1) = \vec{u}_i(t) + \Delta t \frac{\vec{f}_i}{\rho_i}$
- $\vec{x}_i(t+1) = \vec{x}_i(t) + \Delta t \vec{u}_i(t+1)$

SPH——核函数的选取

- 稳定性
- 准确性
- 速度
- $W_{poly6}(r,h)=\frac{315}{64\pi h^9}\begin{cases}(h^2-r^2)^3, 0\le r\le h\\ 0, \qquad otherwise\end{cases}$
- $W_{spiky}(r,h)=\frac{15}{\pi h^6}\begin{cases}(h-r)^3, 0\le r\le h\\ 0, \qquad otherwise\end{cases}$



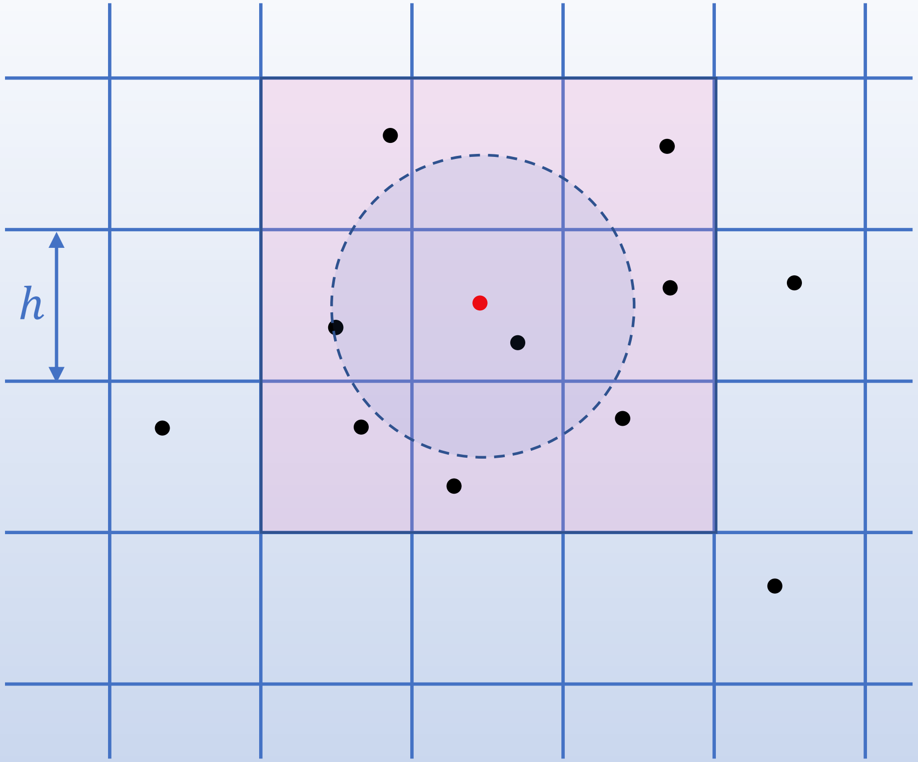
(a)



(b)

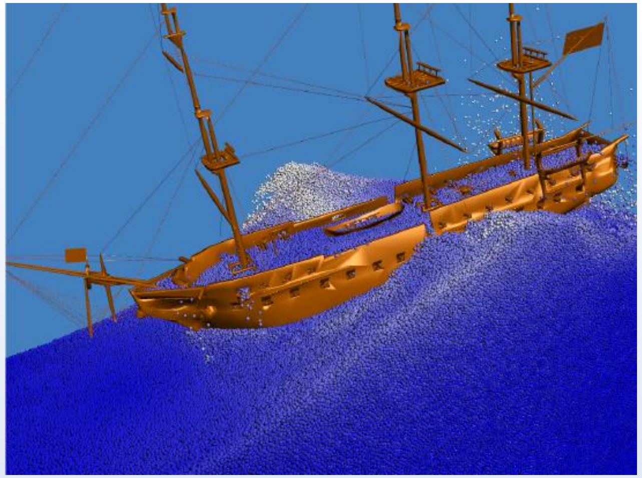
SPH——邻域搜索

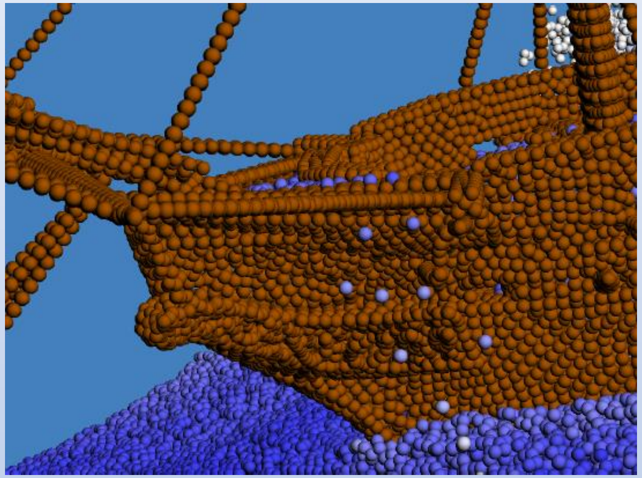
- 最耗性能的部分
- 将空间划分成大小为h的单元
- 只需搜索27个单元
- 步骤
  - 创建网格
  - 插入粒子
  - 计算邻域



SPH——修正密度计算

- 边界粒子b_i的体积：$V_{b_i} = \frac{m_{b_i}}{\rho_{b_i}} = \frac{m_{b_i}}{\sum_k m_{b_k} W_{ik}}$
- 流体粒子f_i的密度：$\rho_{f_i} = m_{f_i}\sum_j W_{ij} + m_{f_i}\sum_k W_{ik}$
- 边界粒子体积大小对流体密度的影响：$\Psi_{b_i}(\rho_0) = \rho_0 V_{b_i}$
- 修正后的流体密度：$\rho_{f_i} = m_{f_i}\sum_j W_{ij} + m_{f_i}\sum_k \Psi_{b_k}(\rho_0) W_{ik}$
- 两个流体粒子之间的压力：$\vec{F}^p_{i\leftarrow j} = -m_i m_j (\frac{p_j}{\rho_i \rho_j}) \nabla W_{ij} = -m_i m_j (\frac{p_x}{\rho_x^2}) \nabla W_{ij}$
- 边界粒子b_j对流体粒子f_i的压力：$\vec{F}^p_{f_i\leftarrow b_j} = -m_{f_i} \Psi_{b_j}(\rho_0) (\frac{p_{f_i}}{\rho_{f_i}^2}) \nabla W_{ij}$
- 液体粒子f_i对边界粒子b_j的压力：$\vec{F}^p_{b_j\leftarrow f_i} = -\vec{F}^p_{f_i\leftarrow b_j}$

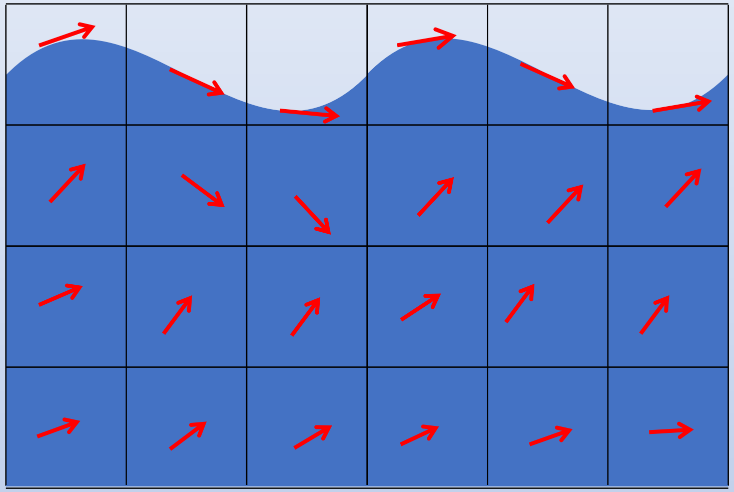


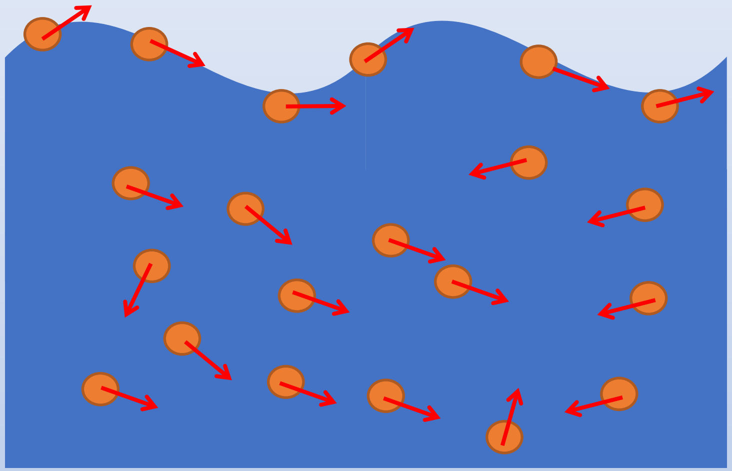


目录

欧拉网格和拉格朗日粒子法的比较

- 欧拉网格法
  - 投影 ☺
  - 邻域查找 ☺
  - 对流 ☹
  - 并行 ☹
- 拉格朗日粒子法
  - 对流 ☺
  - 并行 ☺
  - 动量、能量守恒 ☺
  - 邻域查找 ☹





目录

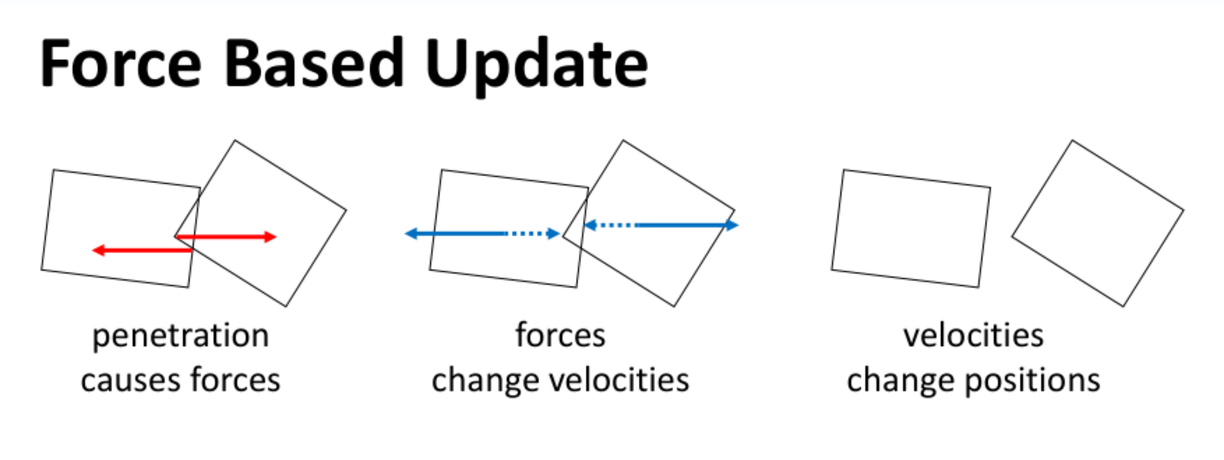
- 流体基础
- 基于力的流体模拟
- **基于约束的流体模拟**

基于力的动力学

- $\vec{F}_{internal}$—— 如流体内部的粘滞力(Viscosity)、压力(Pressure)等
- $\vec{F}_{external}$—— 如重力(Gravity)、碰撞力(Collision)、风力(Wind)等
- 合力 $\vec{F} = \vec{F}_{internal} + \vec{F}_{external}$，计算加速度 $\vec{a} = \frac{\vec{F}}{m}$
- $\vec{v} = \vec{a}t$
- $\vec{x}^* = \vec{x}_0 + \vec{v}t$

基于力的动力学的缺陷

- 力→加速度→速度→位置
  - 重力
  - 摩擦力
  - 碰撞力
    - 瞬时力：作用时间极短→步长
    - 非常巨大
    - 随时间迅速变化，其规律非常复杂
    - 塑性变形
    - 能量转换（发声、发光、发热）
    - 数值积分



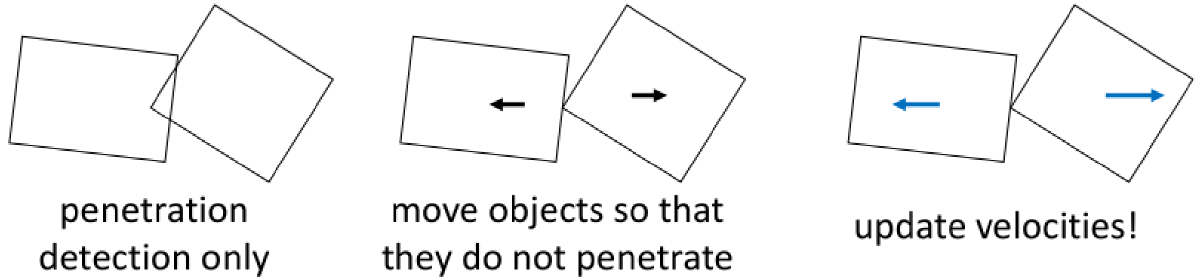
目录

- 流体基础
- 基于力的流体模拟
- **基于约束的流体模拟**
  - Position Based Dynamics
  - Position Based Fluid

基于位置的动力学（PBD）

- 用约束投影代替力和数值积分
  - 只检测发生穿透碰撞
  - 根据约束计算物体修正位置
  - 根据修正位置求解速度

Position Based Update



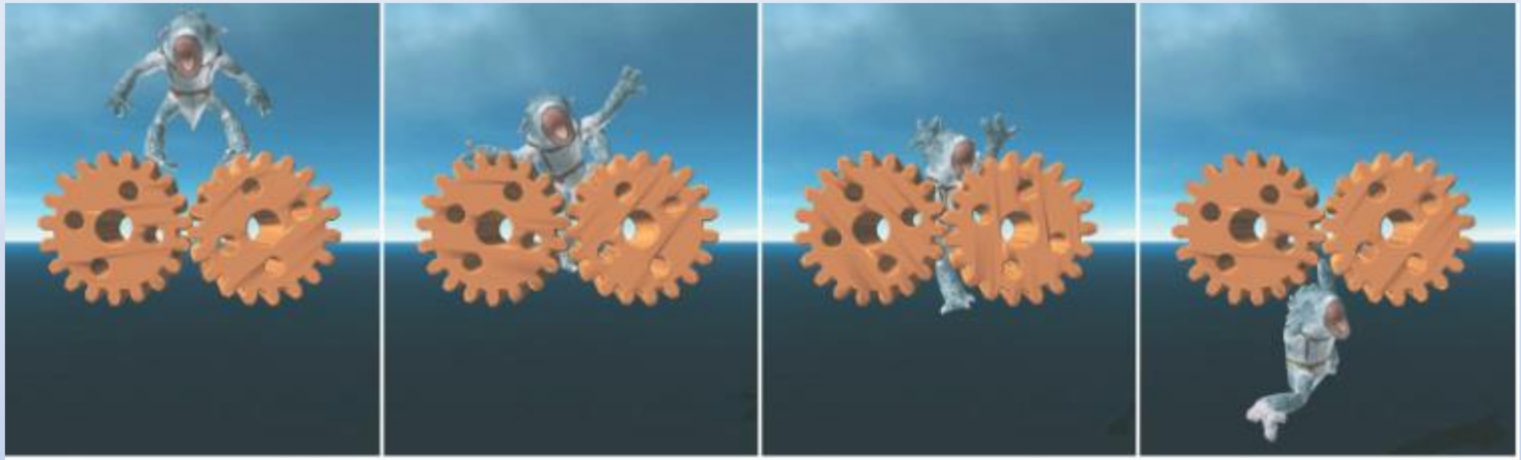


Figure 1: *A known deformation benchmark test, applied here to a cloth character under pressure.*

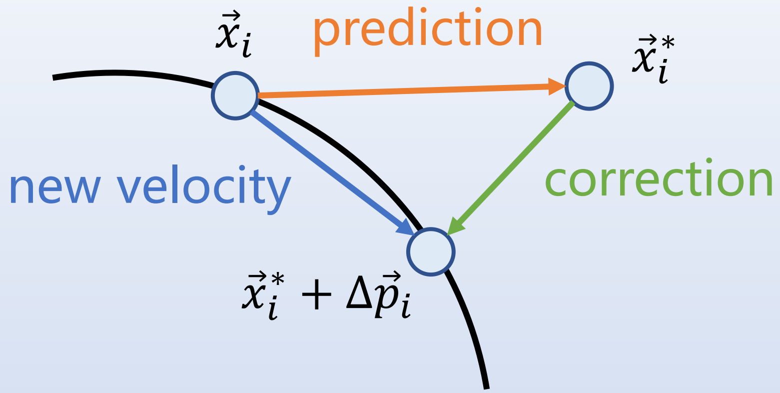
PBD算法

N顶点, M约束**表示动力学物体**

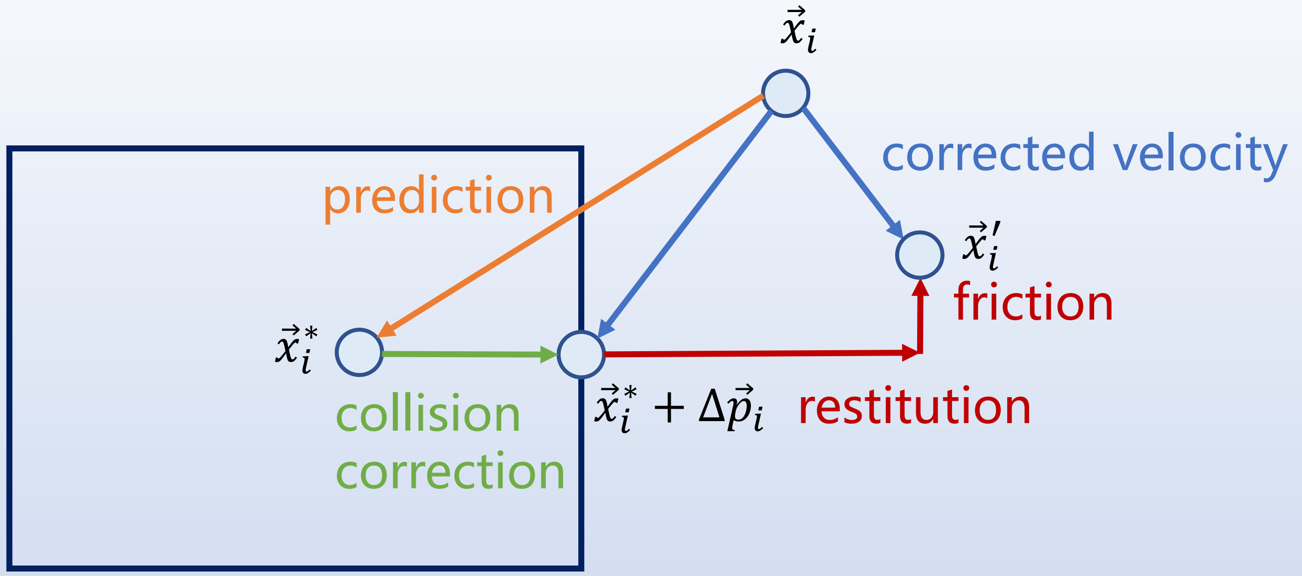
init $\vec{x}_i, \vec{v}_i, w_i \leftarrow {}^1/_{m_i}$	$\vec{x}_i, \vec{v}_i, \vec{x}_i^* \quad \in \mathbb{R}^{3N}$
loop	
$\quad \vec{v}_i \leftarrow \vec{v}_i + \Delta t\, w_i\, \vec{F}_{ext}$	
$\quad \vec{x}_i^* \leftarrow \vec{x}_i + \Delta t \vec{v}_i$	prediction
$\quad generateCollisionConstraints(\vec{x}_i \rightarrow \vec{x}_i^*)$	detect collision
$\quad \Delta\vec{p}_i \leftarrow projectConstraint(C_1, \dots, C_{M+M_{coll}}, \vec{x}_1, \dots, \vec{x}_n^*)$	constraint position
$\quad \vec{x}_i^* \leftarrow \vec{x}_i^* + \Delta\vec{p}_i$	position correction
$\quad \vec{v}_i \leftarrow (\vec{x}_i^* - \vec{x}_i)/\Delta t$	velocity update
$\quad \vec{x}_i \leftarrow \vec{x}_i^*$	position update
$\quad velocityUpdate(\vec{v}_1, \dots, \vec{v}_n)$	velocity correction
end loop	

PBD算法中位置修正

➢例子：圆上的粒子



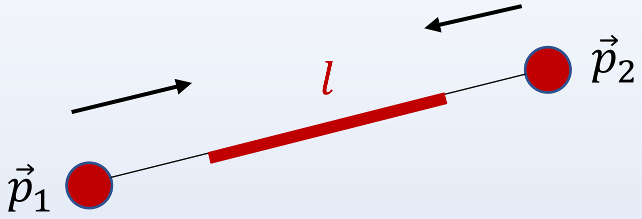
PBD算法中速度修正



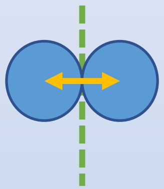
$\vec{x}_i$
corrected velocity
prediction
$\vec{x}_i'$
friction
$\vec{x}_i^*$
collision
correction
$\vec{x}_i^* + \Delta\vec{p}_i$
restitution

约束(Constraints)

- 约束是一个优化问题的解需要符合的条件
  - 等式约束
  - 不等式约束
- 约束类型
  - 距离约束（布料）
  - 形状约束（刚体，塑料）
  - 密度约束（流体）
  - 体积约束（气体）
  - 接触约束（无穿透）



$$C_{distance} = \|\vec{p}_1 - \vec{p}_2\| - l = 0$$



$$C_{contact} = \|\vec{p}_1 - \vec{p}_2\| - 2r \geq 0$$

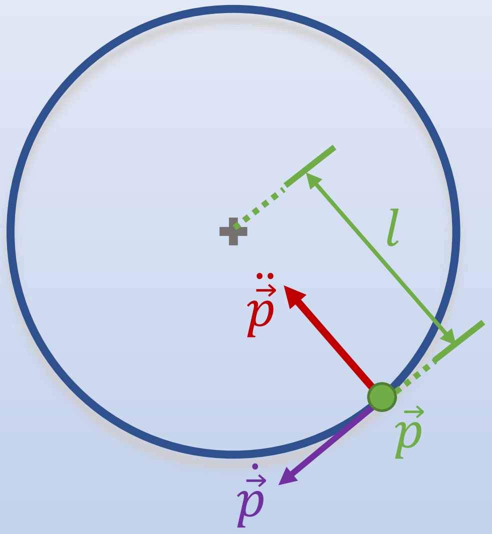
PBD的物理意义

- PBD 这个方法研究的是一个带约束的运动问题

- 高斯最小二乘约束原理(Gauss's principle of least constraint)
  - 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
  - $Z = \min \sum_i m_i \left\| \ddot{\vec{p}}_i - \frac{\vec{F}_{ext}}{m_i} \right\|^2$

约束对加速度的改变有多大

$\vec{p}_i$



高斯最小二乘约束原理应用

$$\underset{}{argmin}\sum_i m_i \left\| \ddot{\vec{p}}_i - \frac{\vec{F}_{ext}}{m_i} \right\|^2$$

$\vec{p}_i^t$ 和 $\vec{v}_i^t$ 质点i 在 t 时刻位置和速度，Δt 一个时间步长

质点i位置: $$\vec{p}_i^{t+\Delta t} = \vec{p}_i^t + \Delta t \left(\vec{v}_i^t + \Delta t \frac{\vec{F}_{ext}}{m_i} \right) + \Delta \vec{p}_i \quad (1)$$

质点i速度: $$\vec{v}_i^{t+\Delta t} = \frac{\vec{p}_i^{t+\Delta t} - \vec{p}_i^t}{\Delta t} = \vec{v}_i^t + \Delta t \frac{\vec{F}_{ext}}{m_i} + \frac{\Delta \vec{p}_i}{\Delta t} \quad (2)$$

质点i加速度: $$\ddot{\vec{p}}_i = \frac{\vec{v}_i^{t+\Delta t} - \vec{v}_i^t}{\Delta t} = \frac{\Delta \vec{p}_i}{\Delta t^2} + \frac{\vec{F}_{ext}}{m_i} \quad (3)$$

$$argmin \sum_i m_i \left\| \frac{\Delta \vec{p}_i}{\Delta t^2} \right\|^2$$

$$argmin \sum_i \frac{1}{2} m_i \|\Delta \vec{p}_i\|^2$$

$$argmin \frac{1}{2} \Delta \boldsymbol{p}^T \boldsymbol{M} \Delta \boldsymbol{p}$$

$$C(\boldsymbol{p}) = 0 \rightarrow C(\boldsymbol{p} + \Delta \boldsymbol{p}) = 0$$

$$\boldsymbol{p} = \begin{bmatrix} \vec{p}_1 \\ \vdots \\ \vec{p}_n \end{bmatrix} \quad \Delta \boldsymbol{p} = \begin{bmatrix} \Delta \vec{p}_1 \\ \vdots \\ \Delta \vec{p}_n \end{bmatrix} \quad \boldsymbol{M} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & m_n \end{bmatrix}$$

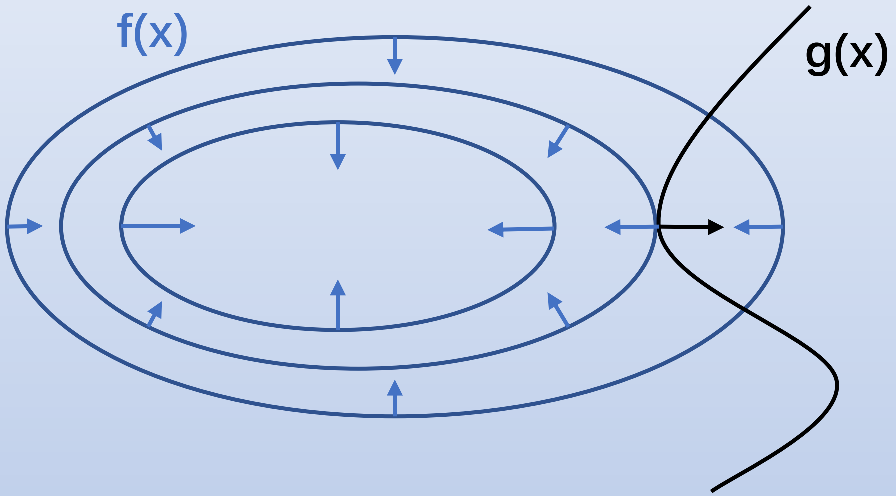
单个约束优化求解

- $$\begin{aligned} &\arg\min \tfrac{1}{2}\Delta \boldsymbol{p}^T \boldsymbol{M} \Delta \boldsymbol{p} \\ &s.t. \quad C(\boldsymbol{p}+\Delta \boldsymbol{p})=0 \end{aligned}$$
- $$\begin{aligned} &f(\boldsymbol{p})=\tfrac{1}{2}\Delta \boldsymbol{p}^T \boldsymbol{M} \Delta \boldsymbol{p} \\ &g(\boldsymbol{p})=C(\boldsymbol{p}) \end{aligned}$$
- 引入拉格朗日乘子 λ
  - $\nabla f(\boldsymbol{p}) + \lambda \nabla g(\boldsymbol{p}) = \boldsymbol{0}$
  - $\boldsymbol{M}\Delta \boldsymbol{p} + \lambda \nabla C(\boldsymbol{p}) = \boldsymbol{0}$

$$\boxed{\Delta \boldsymbol{p} = -\lambda \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p})}$$

一个方程两个未知数，怎么解?

- 拉格朗日乘子法
- $$\begin{aligned} &\arg\min \quad f(x) \\ &s.t. \quad g(x)=0 \end{aligned}$$
- $\nabla f(x) + \lambda \nabla g(x) = 0$
- $L(x,\lambda) = f(x) + \lambda g(x)$



单个约束优化求解

- $\Delta \boldsymbol{p} = -\lambda \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p})$
- $C(\boldsymbol{p} + \Delta \boldsymbol{p}) = 0$
- $C(\boldsymbol{p} + \Delta \boldsymbol{p}) \approx C(\boldsymbol{p}) + \nabla C(\boldsymbol{p}) \cdot \Delta \boldsymbol{p} = \boldsymbol{0}$
- $\begin{cases} \Delta \boldsymbol{p} = -\lambda \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p}) \\ C(\boldsymbol{p}) + \nabla C(\boldsymbol{p}) \cdot \Delta \boldsymbol{p} = 0 \end{cases}$ $\Rightarrow$ $\boxed{\begin{cases} \lambda = -\dfrac{\mathrm{C}(\boldsymbol{p})}{\nabla C(\boldsymbol{p})^T \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p})} \\ \Delta \boldsymbol{p} = \lambda \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p}) \end{cases}}$

多个约束优化求解

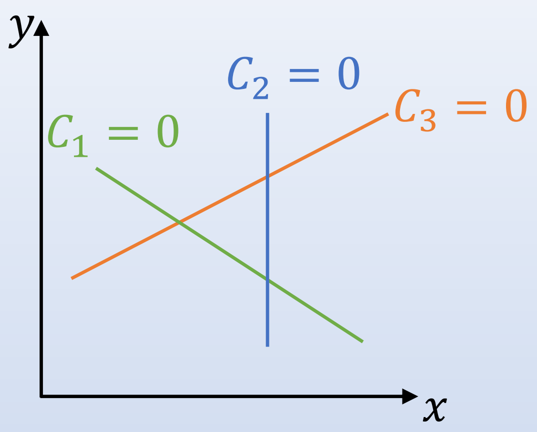
- N个粒子受1个约束的情况

$$\Delta \boldsymbol{p} = \lambda \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p})$$

$$\Delta \boldsymbol{p} = \begin{bmatrix} \Delta \vec{p}_1 \\ \vdots \\ \Delta \vec{p}_n \end{bmatrix}$$

- N个粒子受M个约束的情况
  - 解可能不存在
  - 可能存在很多个解

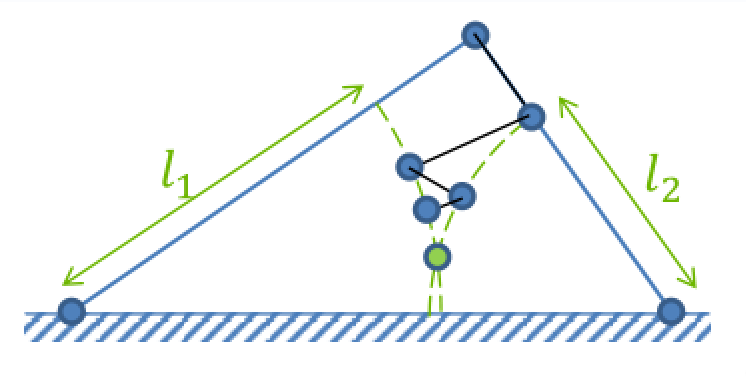
$$\begin{gathered} \Delta \boldsymbol{p}_1 = \lambda_1 \boldsymbol{M}^{-1} \nabla C_1(\boldsymbol{p}) \\ \Delta \boldsymbol{p}_2 = \lambda_2 \boldsymbol{M}^{-1} \nabla C_2(\boldsymbol{p}) \\ \vdots \\ \Delta \boldsymbol{p}_M = \lambda_M \boldsymbol{M}^{-1} \nabla C_M(\boldsymbol{p}) \end{gathered}$$

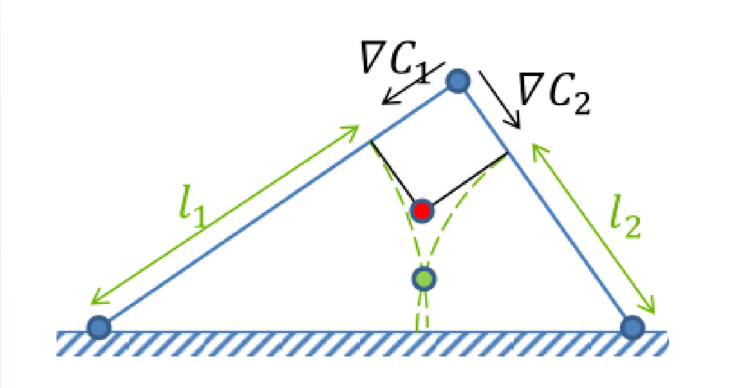


- 迭代法求解：
  - 高斯-赛德尔迭代
  - 雅可比迭代

约束求解器

- 高斯-赛德尔迭代
  - 收敛速度较慢
  - 不可并行
- 雅可比迭代
  - 收敛速度慢
  - 可并行
- 平均雅可比迭代 $\Delta\vec{p}_i = \frac{1}{n_i}\Delta\vec{p}_i$ （ n_i 为影响 i 的粒子数）
- 加入超松因子(SOR) $\Delta\vec{p}_i = \frac{\omega}{n_i}\Delta\vec{p}_i \ (1 \le \omega \le 2)$





约束求解优先级

- 约束类型分组
- 相同类型一组
- 优先级高的先处理，把 $\Delta\vec{p}_i$ 累加到 $\vec{p}_i$ 上，再处理低优先级
- 例:
  - Process Collision Constraint
  - Process Density Constraint

目录

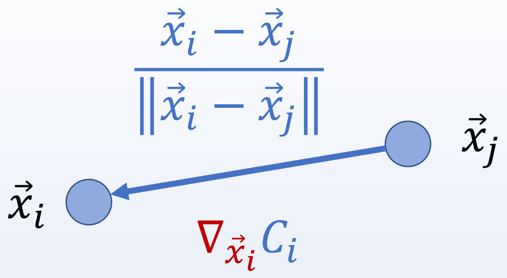
- 流体基础
- 基于力的流体模拟
- 基于约束的流体模拟
  - Position Based Dynamics
  - Position Based Fluid

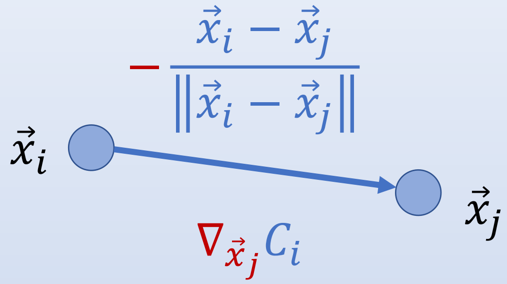
PBF——流体的密度约束

- $C_i(\vec{x}_1, \ldots, \vec{x}_n) = \frac{\rho_i}{\rho_0} - 1 = 0$
- ρ_0 为静止密度 (1000kg/m^3)
- 怎么求 ρ_i ?
  - 物理量 $A_i = \sum_j \frac{m_j}{\rho_j} A_j W(\vec{x}_i - \vec{x}_j, h)$
  - 密度 $\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W(\vec{x}_i - \vec{x}_j, h) = \sum_j m_j W(\vec{x}_i - \vec{x}_j, h)$
- 梯度： $\nabla_{\vec{x}_k} C_i = \frac{1}{\rho_0} \sum_j \nabla_{\vec{x}_k} W\left(\vec{x}_i - \vec{x}_j, h\right)$ (k 包括自身 i 和邻居 j)

PBF——流体的密度约束

- $\nabla_{\vec{x}_k} C_i = \begin{cases} \sum_j \nabla_{\vec{x}_k} W(\vec{x}_i - \vec{x}_j, h) & if\ k = i \\ -\nabla_{\vec{x}_k} W(\vec{x}_i - \vec{x}_j, h) & if\ k = j \end{cases}$
- $\lambda = -\dfrac{C(\boldsymbol{x})}{\nabla_{\boldsymbol{x}} C(\boldsymbol{x})^T \nabla_{\boldsymbol{x}} C(\boldsymbol{x})} = -\dfrac{C(\boldsymbol{x})}{\|\nabla_{\boldsymbol{x}} C(\boldsymbol{x})\|^2}$
- $\lambda_i = -\dfrac{C_i(\vec{x}_i, \cdots, \vec{x}_n)}{\sum_k \left\|\nabla_{\vec{x}_k} C_i\right\|^2}$

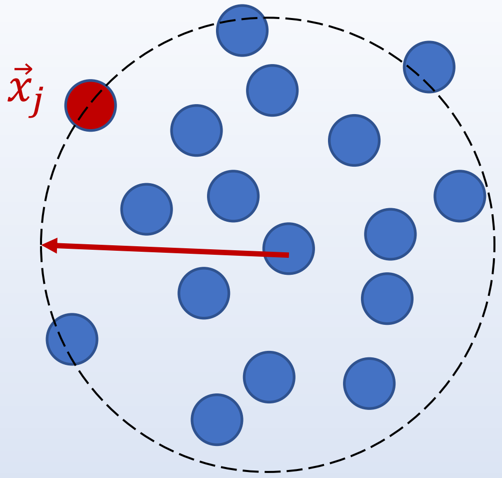




$W(\vec{x}_i - \vec{x}_j, h)$: poly6
$\nabla_{\vec{x}_k} W(\vec{x}_i - \vec{x}_j, h)$: spiky

PBF——拉格朗日乘子中的除0问题

- $\vec{r} = \vec{x}_i - \vec{x}_j$
- 当 $\|\vec{r}\| = h$ 时， $\nabla W(\vec{r}, h) = 0$
  - $\nabla_{\vec{x}_j} C_i = -\frac{1}{\rho_0} \nabla_{\vec{x}_j} W(\vec{r}, h) = 0$
  - $\nabla_{\vec{x}_i} C_i = \frac{1}{\rho_0} \sum_j \nabla_{\vec{x}_i} W(\vec{r}, h) = 0$
- 加入松弛因子ε

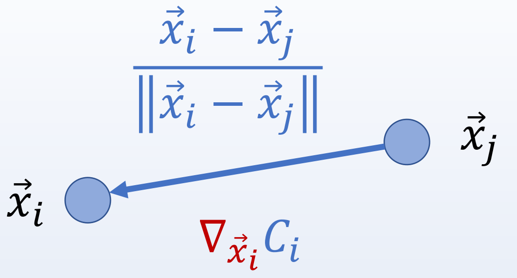


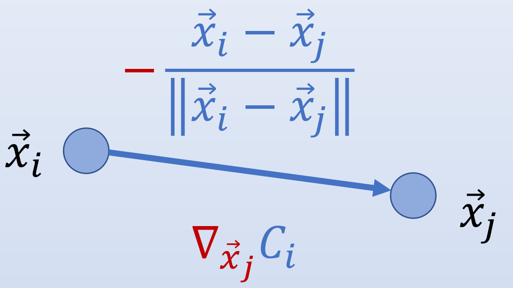
$$\lambda_i = -\frac{C_i(\vec{x}_i, \cdots, \vec{x}_n)}{\sum_{\boldsymbol{k}} \left\|\nabla_{\vec{x}_k} C_{\boldsymbol{i}}\right\|^2} \Rightarrow \lambda_i = -\frac{C_i(\vec{x}_1, \cdots, \vec{x}_n)}{\sum_k \left\|\nabla_{\vec{x}_k} C_i\right\|^2 + \varepsilon}$$

PBF——位置修正

$$\boxed{\Delta \boldsymbol{p} = \lambda \boldsymbol{M}^{-1} \nabla C(\boldsymbol{p})}$$

$$\begin{aligned}
\Delta \vec{x}_i &= \lambda_i \nabla_{\vec{x}_i} C_i + \sum_j \lambda_j \nabla_{\vec{x}_j} C_i \\
&= \frac{1}{\rho_0} \sum_j \lambda_i \nabla_{\vec{x}_i} W(\vec{r}, h) + \left(-\frac{1}{\rho_0} \sum_j \lambda_j \nabla_{\vec{x}_j} W(\vec{r}, h)\right) \\
&= \frac{1}{\rho_0} \sum_j \lambda_i \nabla_{\vec{x}_i} W(\vec{r}, h) + \frac{1}{\rho_0} \sum_j \lambda_j \nabla_{\vec{x}_i} W(\vec{r}, h) \\
&= \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla_{\vec{x}_i} W(\vec{r}, h)
\end{aligned}$$





$k = i$ 时，$\nabla_{\vec{x}_i} C_i$表示约束函数 C_i关于$\vec{x}_i$的梯度，方向为 $\frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|}$

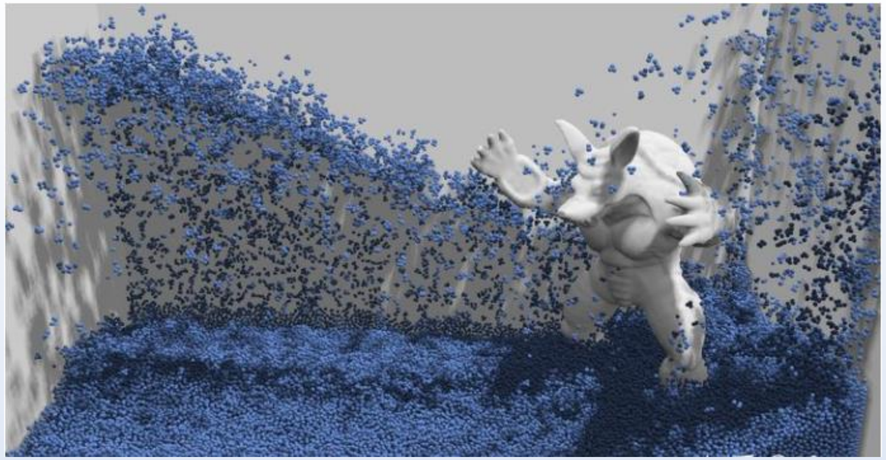
$k = j$ 时，$\nabla_{\vec{x}_j} C_i$表示约束函数 C_i关于$\vec{x}_j$的梯度，方向为 $-\frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|}$

PBF——Tensile Instability

➤邻居粒子不足导致 $\rho_i < \rho_0$

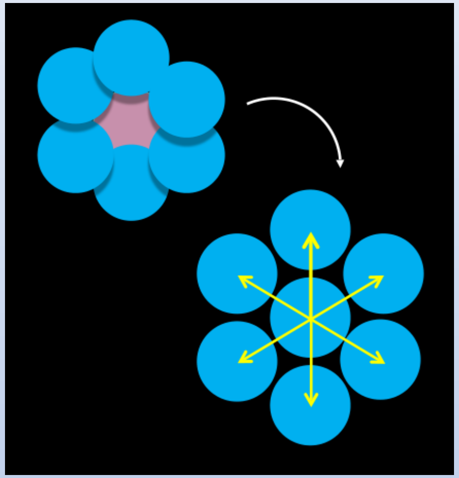
➤负压导致粒子间压力变成吸引力

➤吸引力产生不符合真实情况的凝聚



➤解决方法：

- ➤添加一种排斥力，避免粒子凝聚
- ➤$C_i(\vec{x}_1, \ldots, \vec{x}_n) = \frac{\rho_i}{\rho_0} - 1 \leq 0$



后续...

- Unified particle physics for real-time applications (UPP)
- 混合欧拉-拉格朗日方法
- 各种约束的实现
- 并行计算
- 涡流、湍流的模拟
- 流体渲染
- ……

参考文献

- Müller, Matthias, et al. "Position based dynamics." Journal of Visual Communication and Image Representation 18.2 (2007): 109-118.
- Macklin, Miles, and Matthias Müller. "Position based fluids." ACM Transactions on Graphics (TOG) 32.4 (2013): 1-12.
- Macklin, Miles, et al. "Unified particle physics for real-time applications." ACM Transactions on Graphics (TOG) 33.4 (2014): 1-12.
- https://zhuanlan.zhihu.com/p/48737753
- https://zhuanlan.zhihu.com/p/49536480
- 《计算流体力学入门》
- 《Fluid Engine Development》
- 《Fluid Simulation for Computer Graphics》

Q&A