

1 物理模拟

万事万物的运动都遵循一定的原理. 经过物理学家们长久以来的努力, 我们已经可以用一系列物理方程描述物体的运动, 但这些方程往往复杂而难以求解. 于是, 人们开始寻求这些方程的近似解, 并且希望这些近似解足够精确, 以至于可以用来模拟现实世界中的物理现象. 这种通过数值方法求解物理方程, 并用计算机进行模拟的过程, 就称为**物理模拟 (Physical Simulation)**.

1.1 弹簧质点模型

在本节中, 我们将以弹簧质点模型为例介绍物理模拟的一般方法. 同时, 这一模型也是计算机图形学中常用的物理模拟模型之一, 研究其仿真办法也具有相当的实际意义.

1.1.1 物理模型

一般而言, 物理模拟的范围主要是宏观低速物体的运动, 因此主要使用牛顿运动定律进行运动的计算. 牛顿运动定律为

$$\mathbf{f} = m\mathbf{a} = m \frac{d\mathbf{v}}{dt} = m \frac{d^2\mathbf{x}}{dt^2}$$

1.1.2 物理模拟

在上一小节中, 我们的研究对象是一个质点, 并且它仅受一个力. 这个例子实在过于简单, 无法代表一般的情形. 通常而言, 物体包含无穷多粒子, 并且往往存在多个外力以及复杂的内力. 为了将这样的复杂的系统变为数值求解问题, 我们首先需要对物体进行离散化处理.

因此, 物理模拟的过程大致可以分为三个步骤: **空间离散化**, **时间离散化**以及**数值计算**.

1.1.3 空间离散化

通过一定的方法, 我们可以把物体抽象成由有限个质点组成的系统. 单个质点的状态由其位置 \mathbf{x}_i 和速度 \mathbf{v}_i , 而为了考虑其运动状态的变化, 我们还需通过其受的力 \mathbf{f}_i 和质量 m_i 得出其加速度. 因此, 一个质点模型可以由列表 $\{\mathbf{x}_i, \mathbf{v}_i, \mathbf{f}_i, m_i\}_{i=1}^N$ 表示.

在抽象成质点的情形下, 弹簧质点模型假定质点之间存在弹簧以传递内力. 质点 i 受的外力可以写做

$$\mathbf{f}_i = \sum_{j \in \Omega(i)} \mathbf{f}_{ij} + \mathbf{f}_i^{\text{ext}}$$

其中 $\mathbf{f}_i^{\text{ext}}$ 表示质点 i 受到的除去弹簧弹力之外的外力. 质点 i 与其邻接的质点 $j \in \Omega(i)$ 通过弹簧相连, 弹簧对质点 i 的弹力 \mathbf{f}_{ij} 可以通过胡克定律计算:

$$\mathbf{f}_{ij} = -k_{ij}(|\mathbf{x}_i - \mathbf{x}_j| - l_{ij}) \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}$$

其中 k_{ij} 为弹簧的弹性系数, l_{ij} 为弹簧的原长.

1.1.4 时间离散化

物体的运动状态是随时间连续变化的. 同样地, 我们也需要对时间进行离散化. 一般而言, 我们会在时间区间上均匀采样, 采样间隔为 Δt . 我们把 N 个质点的位置 \mathbf{x} 和速度 \mathbf{v} 写做 $3N$ 维的堆叠向量的形式, 在第 k 次采样 t_k 时有:

$$\mathbf{x}(t_k) = \begin{bmatrix} \mathbf{x}_1(t_k) \\ \dots \\ \mathbf{x}_n(t_k) \end{bmatrix}, \quad \mathbf{v}(t_k) = \begin{bmatrix} \mathbf{v}_1(t_k) \\ \dots \\ \mathbf{v}_n(t_k) \end{bmatrix}$$

其中 $\mathbf{x}_i(t)$ 表示 t 时刻质点 i 的位置, $\mathbf{v}_i(t)$ 同理. 根据简单的运动学知识, 我们可以得到下一次采样时的位置矩阵 $\mathbf{x}(t_{k+1})$ 和 $\mathbf{v}(t_{k+1})$:

$$\begin{aligned} \mathbf{x}(t_{k+1}) &= \mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{v}(t) dt \\ \mathbf{v}(t_{k+1}) &= \mathbf{v}(t_k) + \mathbf{M}^{-1} \int_{t_k}^{t_{k+1}} \mathbf{f}(t, \mathbf{x}(t), \mathbf{v}(t)) dt \end{aligned}$$

其中 $\mathbf{M} = \text{diag}\{m_1, m_1, m_1, \dots, m_N, m_N, m_N\}$ 为质点的质量矩阵.

现在, 我们的目标就是求解上述积分方程. 典型的计算时间积分的办法有**显式欧拉 (Explicit Euler)** 积分和**隐式欧拉 (Implicit Euler)** 积分.

显式欧拉积分 在复杂的情形中, 上述积分方程可能不存在解析表达. 好在当取样的时间间隔较小时, 我们可以将被积函数视作常数, 取值即为每个时间间隔开始的时刻, 于是即有

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \mathbf{v}(t_k) \Delta t$$

$$\mathbf{v}(t_{k+1}) = \mathbf{v}(t_k) + \mathbf{M}^{-1} \mathbf{f}(t_k) \Delta t$$

这样, 对于弹簧质点系统的模拟就归结于上述迭代过程. 显式欧拉积分的优点在于计算和逻辑简单, 每一步只需要计算当前时刻的力即可.

然而, 这样的简单近似也有其缺点. 当时间步长较大时, $\mathbf{v}(t)$ 和 $\mathbf{f}(t)$ 可能在时间间隔内变化较大, 从而导致解得的 $\mathbf{x}(t)$ 和 $\mathbf{v}(t)$ 偏离实际情况, 使得系统能量升高. 这又进一步导致 $\mathbf{v}(t)$ 和 $\mathbf{f}(t)$ 的偏离, 最终可能导致系统发散, 无法继续模拟下去.

解决上述问题的一种办法是使用更小的时间步长, 但这会大大增加计算量, 反而得不偿失. 因此, 我们需要一种更稳定的办法.

隐式欧拉积分 与显式欧拉积分不同, 隐式欧拉积分在计算下一个时间步长的状态时, 使用的是每个时间间隔结束的時刻的力. 即有

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \mathbf{v}(t_{k+1}) \Delta t$$

$$\mathbf{v}(t_{k+1}) = \mathbf{v}(t_k) + \mathbf{M}^{-1} \mathbf{f}(t_{k+1}) \Delta t$$

这样, 我们需要解一个隐式方程组. 将 $\mathbf{v}(t_{k+1})$ 代入 $\mathbf{x}(t_{k+1})$ 中可得

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + (\mathbf{v}(t_k) + \mathbf{M}^{-1}\mathbf{f}(t_{k+1})\Delta t) \Delta t$$

为了方便考虑, 我们把力 \mathbf{f} 分成内力和外力两部分:

$$\mathbf{f}(t_{k+1}) = \mathbf{f}_{\text{int}}(t_{k+1}) + \mathbf{f}_{\text{ext}}$$

其中 \mathbf{f}_{ext} 作为外力与时间和质点的位置无关. 于是进一步整理可得

$$\mathbf{x}(t_{k+1}) = [\mathbf{x}(t_k) + \Delta t \mathbf{v}(t_k) + (\Delta t)^2 \mathbf{M}^{-1} \mathbf{f}_{\text{ext}}] + (\Delta t)^2 \mathbf{M}^{-1} \mathbf{f}_{\text{int}}(t_{k+1})$$

等号右边前半部分全部为已知量, 不妨记作 $\mathbf{y}(t_k)$. 于是方程组即为

$$\mathbf{x}(t_{k+1}) - \mathbf{y}(t_k) - (\Delta t)^2 \mathbf{M}^{-1} \mathbf{f}_{\text{int}}(t_{k+1}) = \mathbf{0}$$

接下来, 我们需要将内力 $\mathbf{f}_{\text{int}}(t_{k+1})$ 表示成位置 $\mathbf{x}(t_{k+1})$ 的函数. 对于质点 i 与 j 之间的弹簧, 其弹性势能

$$E_{ij} = \frac{1}{2} k_{ij} (\|\mathbf{x}_j - \mathbf{x}_i\| - l_{ij})^2$$

于是

$$\mathbf{f}_{ij} = -\frac{\partial E_{ij}}{\partial \mathbf{x}_i}$$

从而对于质点 i , 其内力 $\mathbf{f}_{i,\text{int}}$ 就可以写作

$$\mathbf{f}_{i,\text{int}} = -\sum_{j \in \Omega(i)} \frac{\partial E_{ij}}{\partial \mathbf{x}_i} = -\frac{\partial E}{\partial \mathbf{x}_i}$$

其中 E 为弹簧系统的总能量 (与 i 不相连的弹簧的势能对 \mathbf{x}_i 的梯度为 $\mathbf{0}$, 因此可以并入求和项中), 其自变量为各质点的位置向量 \mathbf{x} . 将所有质点的内力堆叠起来, 我们就有

$$\mathbf{x}(t_{k+1}) - \mathbf{y}(t_k) + (\Delta t)^2 \mathbf{M}^{-1} \frac{\partial E(\mathbf{x}(t_{k+1}))}{\partial \mathbf{x}} = \mathbf{0}$$

于是, 可以构造辅助函数

$$g(\mathbf{x}) = \frac{1}{2(\Delta t)^2} (\mathbf{x} - \mathbf{y}(t_k))^t \mathbf{M} (\mathbf{x} - \mathbf{y}(t_k)) + E(\mathbf{x})$$

则有

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{x}(t_{k+1}) - \mathbf{y}(t_k) + (\Delta t)^2 \mathbf{M}^{-1} \frac{\partial E(\mathbf{x}(t_{k+1}))}{\partial \mathbf{x}} = \mathbf{0}$$

于是, 目标就转化为最小化问题

$$\mathbf{x}(t_{k+1}) = \arg \min_{\mathbf{x}} g(\mathbf{x})$$

对于上述最小化问题, 使用梯度下降法可能不是一个很好的选择, 因为 $g(\mathbf{x})$ 很可能是非凸的. 我们采取牛顿法解决上述问题.

牛顿法的基本思想是在每轮迭代中使用一个二次函数逼近目标函数, 然后求解该二次函数的极小值作为下一次迭代的点. 迭代过程每次通过尝试解 $\mathbf{x}^{(k)}$ 计算下一次

1.2 流体模拟

1.2.1 流体的基本性质

定理 1.1 Navier-Stokes 方程 流体的运动遵循 Navier-Stokes 方程, 其表达式为

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}$$

其中 $\frac{Dv}{Dt}$ 为流体的随体导数, 其定义为

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v}$$

定理 1.2 不可压性质 流体中的质量守恒方程为

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v})$$

一般的流体模拟中总是假设流体不可压缩/膨胀, 于是密度 ρ 在各处和任意时刻均为常数. 于是上式即为

$$\nabla \cdot \mathbf{v} = 0$$

即流体的速度场是无散的, 也即流入任意一点的流体体积总是等于流出该点的流体体积.

1.2.2 描述流体的两种方式

上述 N-S 方程是以**欧拉视角**描述流体的运动的. 欧拉视角是指将流体的所有物理量看成空间上的一个场, 然后描述这个场随时间的变化. 这好比在水中插有无限多的木桩, 每个木桩上装有检测水的流速, 密度, 压强等的传感器, 我们描述的是所有传感器上的数值变化.

另一种描述流体的方式是**拉格朗日视角**, 即将流体看成由无数个质点组成, 然后描述每个质点随时间的运动. 这好比在水中放入无数悬浮的小球, 其上也装有传感器, 我们描述的是每个小球的位置, 速度等随时间的变化.

1.2.3 光滑粒子流体

我们首先采用拉格朗日视角来描述流体的运动.

定理 1.3 光滑粒子流体模型 光滑粒子流体 (Smoothed Particle Hydrodynamics, SPH) 模型将流体看成有限个流体微团 (视作粒子) 组成的系统. 每个粒子 i 具有位置 \mathbf{x}_i , 速度 \mathbf{v}_i , 质量 m_i .

为了根据上述信息计算流体的宏观性质, 例如求解 N-S 方程必不可少的密度场和压强场, 我们采用一种经典的办法: **核函数**.

核函数 我们已经经历了很多采样问题以从离散的样本点重建连续函数. 对于流体而言, 如果采取简单的近邻法划分空间并计算目标函数, 往往会在边界上造成函数值的跳变, 因而结果交叉. 因此, 我们需要采取一定的办法使得这种不连续性被平滑. 一种自然的想法就是考虑目标点附近的所有粒子对该点性质的影响, 并且距离越近的粒子影响越大. 这就可以通过引入核函数 $W(\mathbf{r}, h)$ 进行加权平均而解决.