

现代物理与人工智能-II（第三部分）

韩子慕，王浩朴

1 序言

本复习资料针对《现代物理与人工智能》课程，主要内容忠实于授课PPT，并对PPT上难以理解（或有省略）的部分进行了补充说明。能给出例题和示例的部分，我们准备了一些练习题供大家参考。2025年考过的题目均以脚注的形式标明。

本复习资料由韩子慕和王浩朴编写。其中热力学与人工智能、统计物理与人工智能、物理仿真内容由韩子慕编写，量子力学与人工智能、混沌与分形部分由王浩朴编写。如有问题请联系对应编者。

2 物理模拟绪论

这一部分去年好像没怎么出题考试，所以简单总结了一些重要内容，大家可以随意看看，就当是在背文科了。

2.1 物理模拟

1. 物理模拟分为**实时的物理模拟和高精度的物理模拟**。
2. 物理模拟基于物理定律，将物理规律转化为数学模型，使用计算机进行物理状态更新计算与物理现象的可视化。
3. 物理模拟中的物态与物理意义上的物态并不相同，常见的“物态”划分往往是为了便于模拟和渲染，而非严格基于物理学上的相变和分子运动特性。计算机图形学的物理模拟中，物态分为：**刚体 (Rigid Bodies)**、**软体 (Soft Bodies)**、**流体 (Fluids)**、**颗粒/粒子系统 (Particle Systems & Granular Materials)**、其它扩展（布料、头发、羽毛）。

4. 布料：常用质点-弹簧或有限元壳体模型进行模拟。头发：将其抽象为一维弹性体或粒子链条。
5. 具身机器人模拟是一种通过高保真仿真环境训练机器人感知、决策与行动能力的技术体系。其核心目标是加速具身智能体（如人形机器人）在虚拟场景中的学习效率，并将训练成果迁移到现实世界，降低实体机器人的试错成本。具身智能仿真的软件工具：NVIDIA Isaac Sim、微软 AirSim、V-REP、PyBullet、MuJoCo、Sapien 等等。
6. 物理模拟的意义：具身智能的基础、构建世界模型、生成高质量训练数据、预测和理解自然现象、降低实验成本、提高安全性、推动工程设计和制造、加速科学发现。
7. 物理模拟是具身智能的基础。机器人、自动驾驶、无人机等有身体的 AI 系统，必须在物理世界中行动。物理模拟提供了：安全试错、高效训练、Corner Case 覆盖。

2.2 计算机图形学简介

计算机图形学 (Computer Graphics, 简称 CG) 是一种使用数学算法将二维或三维图形转化为计算机显示器的栅格形式的科学。简单地说，计算机图形学的主要研究内容就是研究如何在计算机中表示图形、以及利用计算机进行图形的计算、处理和显示的相关原理与算法。

几何或者几何处理是计算机图形学的一个重要分支。这部分主要介绍了网格 (Mesh)、点云 (PointCloud) 和体素 (Voxel) 三种几何表征方式。

2.2.1 网格

1. 网格包含：顶点 (v)、面 (f) 或者体。常见的网格有三角形网格、四边形网格。（给一个网格表示，需要看明白是什么意思，可以看 PPT 的示例）
2. 网格处理包含：重网格化、网格细分、网格优化。

2.2.2 点云

1. 点云就是空间中一群点的 (X,Y,Z) 坐标的集合。有些点云可能含有点的色彩信息 (R,G,B) 或物体反射面强度 (Intensity) 信息。

2. 点云的优势：最为简洁的 3D 表达方式：点的列表 (X,Y,Z) 可以表征任意复杂的几何。
3. 点云的难点：稀疏（远处的物体的点云数很少，这是激光雷达的性质所导致的）、数据量巨大；纹理信息的缺失；无序性（深度学习的困难，顺序不同的矩阵是完全不同的输入）¹

2.2.3 体素

1. 体素网格是对三维空间的规则化划分，类比于 2D 图像上的像素；每个体素单元上存储了不同的物理量。
2. 体素的优势：结构规则且表达直观、完整内部结构表示。
3. 体素的难点：大量的存储、分辨率与精度受限。

2.2.4 计算机图形学与具身智能

具身智能的仿真基于物理世界的规律，是实现算法改进与突破的关键。计算机图形学为物理世界仿真提供了基础支撑。

世界模型（World Model）是人工智能领域的一个概念，指智能体（如 AI 系统）通过学习环境中的规律和动态变化，构建一个内部模拟器来预测未来状态或理解环境如何运作。

3 运动模拟基础

3.1 运动建模

运动建模有三个步骤：**建模、初始条件、模拟**。建模是根据牛顿第二定律确定加速度，初始条件由题目给定，模拟的过程本质上是对连续积分过程进行离散化近似。本章主要介绍三种常用数值模拟方法：欧拉积分、结束速度法和平均速度法。

¹这里就可以看经典文章 PointNet，针对无序性的问题采用了 Max Pool 的方式解决，很天才的设计。

3.1.1 欧拉积分

欧拉积分是最基础的数值积分方法，其核心思想是：在一个足够小的时间步长 Δt 内，认为物体的加速度和速度均保持不变，从而利用当前时刻的信息推算下一时刻状态。

设在时刻 t_n ，物体的位置、速度分别为

$$\mathbf{x}_n, \quad \mathbf{v}_n$$

速度更新：

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t$$

位置更新：

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t$$

欧拉积分方法实现简单、计算量小，但由于位置更新只采用当前速度估计，误差较大，属于一阶精度方法，长时间模拟中容易产生累计误差，因此常作为基础方法或其他高阶方法的比较基准。

3.1.2 结束速度法

结束速度法在更新 $n+1$ 时刻的位移时，使用 $n+1$ 时刻的速度来计算，与欧拉积分法只有一个角标的差异。

速度更新：

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t$$

位置更新：

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1} \Delta t$$

3.1.3 平均速度法

平均速度法，故名思义，在更新 $n+1$ 时刻的位移时，使用 n 时刻的速度和 $n+1$ 时刻的速度的平均值进行更新。

速度更新：

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n \Delta t$$

位置更新：

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{2}(\mathbf{v}_{n+1} + \mathbf{v}_n) \Delta t$$

由高一物理知识可知，平均速度法在模拟匀加速直线运动时，所得的位移与精确解完全一样。

3.2 碰撞处理

碰撞处理有三个阶段：**碰撞检测**（是否碰撞）、**碰撞测定**（什么时间地点碰撞）、**碰撞响应**（碰撞产生什么效果）。²

3.2.1 碰撞检测

碰撞检测要判断碰撞是否发生。可以通过小球到平面的距离来判断是否碰撞，下面直接讨论三维场景的距离计算。设平面上一点 $\mathbf{p}$ ，那么到平面的距离是：

$$d = (\mathbf{x} - \mathbf{p}) \cdot \hat{\mathbf{n}}$$

再考虑小球的半径，如果小球在平面的正数侧（距离越远 d 越大），那么检测方法：

$$d = (\mathbf{x} - \mathbf{p}) \cdot \hat{\mathbf{n}} - r$$

如果小球在平面的负数侧（距离越远 d 越小），那么检测方法是：

$$d = (\mathbf{x} - \mathbf{p}) \cdot \hat{\mathbf{n}} + r$$

3.2.2 碰撞测定

碰撞测定就是计算是在何时何地发生碰撞。令 d_k 为第 k 个时间步小球到平面的距离，假如我们知道 $d_n > 0, d_{n+1} < 0$ 。那么可以推断碰撞应该在 $n, n+1$ 之间发生，但是我们需要进一步知道到底是在哪个时间点发生，也就是求解**时步分数** f 。碰撞发生在第 n 个时间步后经过 $f \cdot \Delta t$ 的时刻（ Δt 为一个时间步的长度）。我们假设时间步内物体是匀速的，由一个简单的相似三角形就可以得到：

$$f = \frac{d_n}{d_n - d_{n+1}}$$

因此，碰撞的位置为：

$$x_{\text{hit}} = (1 - f) \cdot x_n + f \cdot x_{n+1}$$

3.2.3 碰撞响应

碰撞响应的目标是确定碰撞发生后物体的速度变化。这部分主要依据经典力学中的动量守恒定律和恢复系数模型。

²2025 年考过“碰撞有哪三个阶段”填空题。

考虑两个质量分别为 m_1, m_2 的小球发生一维正碰（正碰是说两个球的相对速度方向在球心连线上），碰撞前速度分别为 v_1, v_2 ，碰撞后速度分别为 v'_1, v'_2 。根据动量守恒定律，有

$$m_1 v_1 + m_2 v_2 = m_1 v'_1 + m_2 v'_2$$

设碰撞恢复系数为 e ，其定义为碰撞后相对分离速度与碰撞前相对接近速度之比，即

$$e = \frac{v'_2 - v'_1}{v_1 - v_2}$$

其中 $0 \leq e \leq 1$ 。当 $e = 1$ 时为完全弹性碰撞（没有机械能损耗）；当 $e = 0$ 时为完全非弹性碰撞（质心系中相对动能完全损耗）。

联立动量守恒方程与恢复系数方程，可解得碰撞后的速度：

$$v'_1 = \frac{(m_1 - em_2)v_1 + (1 + e)m_2 v_2}{m_1 + m_2}$$

$$v'_2 = \frac{(m_2 - em_1)v_2 + (1 + e)m_1 v_1}{m_1 + m_2}$$

一些高中物理熟悉的结论：两个相同质量的物体发生完全弹性碰撞后，速度交换。一个物体与另一质量无穷大的物体（比如地面）发生完全弹性碰撞后，这个物体的速度大小不变，方向相反（反弹）……

下面考虑球与固定面的碰撞。首先球的速度可以分解为法向速度： $\mathbf{v}_n = (\mathbf{v} \cdot \mathbf{n})\mathbf{n}$ ，和切向速度 $\mathbf{v}_t = \mathbf{v} - \mathbf{v}_n$ 。法向速度的更新取决于恢复系数 c_r （在这里换了一个记号）： $v_n^+ = -c_r v_n^-$ ，而切向速度的更新与摩擦力有关。下面讨论两种摩擦力模型：

1. 简单模型。我们定义一个摩擦系数 c_f ，切向速度更新： $v_t^+ = (1 - c_t)v_t^-$ 。
 $c_t = 1$ ，摩擦力使得切向速度直接为 0， $c_t = 0$ ，没有摩擦，切向速度不变。这个模型并不真实，因为实际上摩擦力的大小（或者说，碰撞过程中摩擦力的切向冲量大小）与应该与法向压力有关。
2. 库伦摩擦模型。切向冲量应该与法向冲量成正比，那么定义摩擦系数 μ ，切向速度的更新方程： $v_t^+ = v_t^- - \min(\mu \|v_n^-\|, \|v_t^-\|)$ ，这里做最小值的操作是为了保证切向速度不反向。³

³严谨地说，我认为这并不对。写成 $v_t^+ = v_t^- - \min(\mu(1 + e)\|v_n^-\|, \|v_t^-\|)$ 更合理一些，不过我想还是不必细究了。

3.2.4 静止条件

静止条件是一个用于判定物体处于静止状态并停止计算这个物体运动状态的条件。那么，为什么要设置静止条件？⁴

1. 如果没有静止条件，速度很小但不会结束模拟，反复抖动或振动造成**计算资源的浪费**。
2. 球不断的抖动或振动，产生**模拟的非真实感**（真实世界中抖动一段时间就会停下）。

如何进行静止测试？

1. 速度很小： $\|v\| < \epsilon_1$
2. 存在一个平面，使得小球到这个平面的距离很近 $d < \epsilon_2$
3. 施加于物体的合力指向平面： $F \cdot \hat{n} < \epsilon_3$
4. 摩擦力足够大，能够克服切向力，使得小球静止： $F_t < \mu \|F_n\|$

对大量物体准确的计算上述条件，是一个被证实的难以完成的任务（NP 完全问题），实际上采用近似方法处理。

3.3 多边形几何学

简单多边形：是一个有序的顶点集合，全部顶点都在同一平面上，并且按序连接顶点会组成一个无棱相交的闭环。不满足这个就是复杂多边形，比如五角星有边相交，就不是简单多边形。非平面多边形也不是简单多边形。

3.3.1 判断点是否与多边形碰撞

判断一个点是否与多边形相撞，分三步：

1. 求多边形的平面方程：设 p 是平面上一点，那么满足 $(x - p) \cdot \hat{n} = 0$ 的 x 在平面上。
2. 进行碰撞检测及（如需要）碰撞测定，求出粒子与该平面的相交点。在 i 时刻，物体的位置 x_i ，速度 v_i ，运动方程 $x = x_i + tv_i$ ，带入平面方程求解： $t_{\text{hit}} = \frac{(p - x_i) \cdot \hat{n}}{v_i \cdot \hat{n}}$ ，也就可以解得 $x_{\text{hit}} = x_i + t_{\text{hit}} v_i$ 。

⁴2025 年考过“为什么设置静止条件，如何进行静止测试”简答题。

□如果多边形为凸多边形

- ❖1. 通过投影所有点至二维坐标平面, 问题变成二维
- ❖2. 对每个顶点计算二维**边线矢量**, 即以当前顶点减去下一个顶点
- ❖3. 对每个顶点计算二维**碰撞矢量**, 即 x_{hit} 减去当前顶点
- ❖4. 对每个顶点构造一个 2×2 矩阵, 第一行是边线矢量, 第二行是碰撞矢量
- ❖5. 计算每个上述矩阵的行列式 (determinant)
- ❖6. 若所有行列式的符号都相同, 那么 x_{hit} 必定在多边形之内

$$\begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_{hit} - x_0 & y_{hit} - y_0 \end{vmatrix} = (x_1 - x_0)(y_{hit} - y_0) - (y_1 - y_0)(x_{hit} - x_0)$$

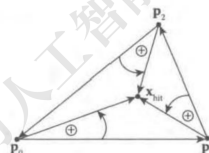


图 1: 判断点在多边形内 (凸多边形)

3. 判断相交点是否在多边形之内。因为 x_{hit} 与多边形是共平面的, 所以可以把三维问题转化为二维问题处理: 选择法向量中最大模的方向进行投影 (比如 $n_x > n_y > n_z$, 那么投影到 yz 平面)。投影之后就是二维平面中如何判断一个点在多边形之内的问题了, 这部分内容在下一节讲述。

3.3.2 判断点是否在多边形内

在凸多边形的情况下: 这里提供一个容易理解的思路, 本质上是对每条边和顶点到待判断点的连线做叉乘, 叉乘结果都指向同一侧, 则说明这个点在多边形内。

在一般的多边形情况下:

1. 射线法 (奇偶规则)。从待判断的点任意方向发射一条直线, 然后统计这条射线与多边形边相交的次数。如果射线与多边形边的交点数为奇数, 则判断该点在多边形内部; 如果交点数为偶数, 则判断该点在多边形外部。当然这个方法并不鲁棒。
2. 绕数法。计算多边形边绕待测点旋转的总角度, 即“绕数”。如果绕数不为零, 则点在多边形内部; 如果绕数为零, 则点在多边形外部。

3.4 物理引擎

物理引擎是一种计算机软件, 能够对特定物理系统 (通常指经典动力学系统) 进行近似模拟, 包括刚体动力学 (含碰撞检测)、软体动力学和流体

动力学等。它主要应用于计算机图形学、电子游戏和电影特效 (CGI) 领域。在视频游戏中 (通常作为中间件), 这类模拟需要实时运行。广义而言, 该术语也可泛指任何用于模拟物理现象的软件系统, 例如高性能科学仿真软件。

PhysX, Havok, Bullet 是全球范围内广泛使用的三大物理引擎, 它们分别归属于英伟达、微软和 AMD 开源计划。ODE (Open Dynamics Engine) 开源刚体动力学库, 适合机器人仿真和简单 3D 物理需求。

4 刚体模拟

刚体模拟的过程是不断更新刚体状态 $\mathbf{s}$ 随时间的变化。设 $t^{[0]}, t^{[1]}, \dots, t^{[k]}$ 为离散的时间点, 则物理引擎 (仿真器) 通过迭代计算, 由当前状态 $\mathbf{s}^{[n]}$ 推导出下一时刻的状态 $\mathbf{s}^{[n+1]}$ 。

刚体运动分为**平移运动**和**旋转运动**。

设刚体上某一点在局部坐标系中的位置为 $\mathbf{r}_i$, 刚体的质心位置为 $\mathbf{x}$, 旋转矩阵为 $\mathbf{R}$, 则该点在全局坐标系中的位置为 $\mathbf{x} + \mathbf{R}\mathbf{r}_i$ 。

其中局部坐标系 (参考坐标系) 以质心为原点, 全局坐标系 (世界坐标系) 为当前的绝对坐标系。

4.1 平移运动

平移运动的状态变量包含位置 $\mathbf{x}$ 和速度 $\mathbf{v}$ 。

速度更新公式 (由牛顿第二定律):

$$\mathbf{v}^{[1]} = \mathbf{v}^{[0]} + \mathbf{M}^{-1} \int_{t^{[0]}}^{t^{[1]}} \mathbf{f}(\mathbf{x}(t), \mathbf{v}(t), t) dt, \quad (1)$$

位置更新公式 (速度积分):

$$\mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt. \quad (2)$$

其中 $\mathbf{M}$ 为质量矩阵, $\mathbf{f}$ 为合外力。

4.2 数值积分

位置 $\mathbf{x}(t) = \int \mathbf{v}(t) dt$ 为 $\mathbf{v}(t)$ 曲线下的面积。数值积分的基本思想是用简单几何图形的面积近似代替曲线下面积。

4.2.1 显式欧拉积分

显式欧拉积分（一阶精度）用矩形的面积近似积分，矩形宽为 Δt ，高取区间左端点的速度值 $\mathbf{v}(t^{[0]})$ ：

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt \approx \Delta t \mathbf{v}(t^{[0]}). \quad (3)$$

对 $\mathbf{v}(t)$ 在 $t^{[0]}$ 处进行泰勒展开可分析误差：

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt = \Delta t \mathbf{v}(t^{[0]}) + \frac{\Delta t^2}{2} \mathbf{v}'(t^{[0]}) + \dots = \Delta t \mathbf{v}(t^{[0]}) + O(\Delta t^2). \quad (4)$$

误差为 $O(\Delta t^2)$ ，故称显式欧拉积分为一阶精度。

4.2.2 隐式欧拉积分

隐式欧拉积分（一阶精度）用矩形面积近似，高取区间右端点的速度值 $\mathbf{v}(t^{[1]})$ ：

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt \approx \Delta t \mathbf{v}(t^{[1]}). \quad (5)$$

泰勒展开分析误差：

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt = \Delta t \mathbf{v}(t^{[1]}) - \frac{\Delta t^2}{2} \mathbf{v}'(t^{[1]}) + \dots = \Delta t \mathbf{v}(t^{[1]}) + O(\Delta t^2). \quad (6)$$

4.2.3 中点积分

中点积分（二阶精度）用矩形面积近似，高取区间中点的速度值 $\mathbf{v}(t^{[0.5]})$ ：

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt \approx \Delta t \mathbf{v}(t^{[0.5]}). \quad (7)$$

对左半区间 $[t^{[0]}, t^{[0.5]}]$ ：

$$\int_{t^{[0]}}^{t^{[0.5]}} \mathbf{v}(t) dt = \frac{\Delta t}{2} \mathbf{v}(t^{[0.5]}) - \frac{(\Delta t/2)^2}{2} \mathbf{v}'(t^{[0.5]}) + O(\Delta t^3). \quad (8)$$

对右半区间 $[t^{[0.5]}, t^{[1]}]$ ：

$$\int_{t^{[0.5]}}^{t^{[1]}} \mathbf{v}(t) dt = \frac{\Delta t}{2} \mathbf{v}(t^{[0.5]}) + \frac{(\Delta t/2)^2}{2} \mathbf{v}'(t^{[0.5]}) + O(\Delta t^3). \quad (9)$$

两段相加：

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t) dt = \Delta t \mathbf{v}(t^{[0.5]}) + O(\Delta t^3). \quad (10)$$

中点积分的误差为 $O(\Delta t^3)$ ，因此称为二阶精度。

4.2.4 平移运动积分离散化

平移运动的状态包含位置 $\mathbf{x}$ 和速度 $\mathbf{v}$ 。将积分用显式欧拉方法离散化：

$$\mathbf{v}^{[1]} = \mathbf{v}^{[0]} + \Delta t \mathbf{M}^{-1} \mathbf{f}^{[0]}, \quad (\text{显式}) \quad (11)$$

$$\mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t \mathbf{v}^{[1]}. \quad (12)$$

其中质量矩阵 $\mathbf{M}$ 和时间步长 Δt 是由用户指定的变量。

5 旋转表示与旋转运动

5.1 旋转矩阵

旋转矩阵是一种直接的旋转表示方法。二维旋转矩阵为

$$\mathbf{R} = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix}. \quad (13)$$

三维旋转矩阵为

$$\mathbf{R} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}. \quad (14)$$

旋转矩阵 $\mathbf{R}$ 是一个正交矩阵，满足以下两个性质：

1. 行列式为 1（保证是纯旋转，无缩放或反射）；
2. 正交性： $\mathbf{R}^T \mathbf{R} = \mathbf{R} \mathbf{R}^T = \mathbf{I}$ （保证旋转后向量长度不变）。

设 $\mathbf{R}_1$ 和 $\mathbf{R}_2$ 是两个旋转矩阵，证明 $\mathbf{R} = \mathbf{R}_1 \mathbf{R}_2$ 仍然是旋转矩阵：

- 行列式： $\det(\mathbf{R}) = \det(\mathbf{R}_1) \det(\mathbf{R}_2) = 1 \cdot 1 = 1$ ；
- 正交性： $\mathbf{R}^T \mathbf{R} = (\mathbf{R}_1 \mathbf{R}_2)^T (\mathbf{R}_1 \mathbf{R}_2) = \mathbf{R}_2^T \mathbf{R}_1^T \mathbf{R}_1 \mathbf{R}_2 = \mathbf{R}_2^T \mathbf{I} \mathbf{R}_2 = \mathbf{I}$ 。

几何上，矩阵的乘法表示旋转的连续施加， $\mathbf{R} = \mathbf{R}_1 \mathbf{R}_2$ 表示先执行旋转 $\mathbf{R}_2$ ，再执行旋转 $\mathbf{R}_1$ 。

5.2 二维旋转矩阵

二维旋转矩阵表示绕原点旋转角度 θ :

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (15)$$

5.3 三维旋转矩阵（绕坐标轴旋转）

绕 x 轴的主动旋转定义为:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}. \quad (16)$$

绕 y 轴的主动旋转定义为:

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (17)$$

绕 z 轴的主动旋转定义为:

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

旋转矩阵表示法存在以下不足:

- 冗余: 二维旋转实际只有一个自由度, 却用四个元素表示; 三维旋转实际只有三个自由度, 却用九个元素表示;
- 物理含义不直观;
- 旋转时间微分 (角速度) 计算复杂。

5.4 欧拉角表示

欧拉角表示法用三个轴向旋转的组合来表示任意旋转。在 Unity 引擎中, 欧拉角的旋转顺序依次是: 先绕 Z 轴旋转 γ , 再绕 X 轴旋转 β , 最后绕 Y 轴旋转 α :

$$\mathbf{R} = \mathbf{R}_z(\gamma) \mathbf{R}_x(\beta) \mathbf{R}_y(\alpha). \quad (19)$$

欧拉角表示法直观，每个轴向旋转只需要一个角度参数即可描述，在设计和控制领域十分常用。但这种表示法不适合动力学模拟，原因在于：

- 会在特定状态下丢失自由度（即**万向节锁**问题）；
- 难以准确定义其时间导数（即旋转角速度）。

因此，欧拉角表示法虽然在界面中直观易用，但在物理引擎内部通常会转换为四元数进行实际计算。

5.5 四元数表示

四元数（Quaternion）是另一种旋转表示方法。复数系统用两个数表示二维点（具有乘法和除法运算），类比地，四元数用四个数表示三维旋转（具有乘法和除法运算）。

四元数由标量部分和三维向量部分组成。设 $\mathbf{q} = [s, \mathbf{v}]$ ，其中 s 为标量部分， $\mathbf{v} = (v_x, v_y, v_z)$ 为三维向量部分。

四元数的基本运算：

数乘：

$$a\mathbf{q} = [a s, a \mathbf{v}]. \quad (20)$$

加減：

$$\mathbf{q}_1 \pm \mathbf{q}_2 = [s_1 \pm s_2, \mathbf{v}_1 \pm \mathbf{v}_2]. \quad (21)$$

乘法（格拉斯曼积）：

$$\mathbf{q}_1 \times \mathbf{q}_2 = [s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2]. \quad (22)$$

模：

$$|\mathbf{q}| = \sqrt{s^2 + \mathbf{v} \cdot \mathbf{v}}. \quad (23)$$

四元数与旋转轴、旋转角度的关系：四元数与旋转轴 $\mathbf{u}$ （单位向量）和旋转角度 θ 的关系为：

$$\mathbf{q} = \cos \frac{\theta}{2} + \mathbf{u} \sin \frac{\theta}{2}. \quad (24)$$

展开为标量和向量形式：

$$\mathbf{q} = \left[\cos \frac{\theta}{2}, \mathbf{u} \sin \frac{\theta}{2} \right]. \quad (25)$$

归一化四元数满足 $|\mathbf{q}| = 1$ ，此时：

$$|\mathbf{q}|^2 = \cos^2 \frac{\theta}{2} + \sin^2 \frac{\theta}{2} = 1. \quad (26)$$

标量部分为 $s = \cos \frac{\theta}{2}$ ，向量部分的模为 $|\mathbf{v}| = \sin \frac{\theta}{2}$ 。四元数是 Unity、虚幻引擎等中默认的旋转表达方式，因为它避免了万向节锁问题且插值计算方便。

四元数表达的旋转矩阵：设归一化四元数 $\mathbf{q} = [s, (x, y, z)]$ ，对应的旋转矩阵为

$$\mathbf{R} = \begin{bmatrix} s^2 + x^2 - y^2 - z^2 & 2(xy - sz) & 2(xz + sy) \\ 2(xy + sz) & s^2 - x^2 + y^2 - z^2 & 2(yz - sx) \\ 2(xz - sy) & 2(yz + sx) & s^2 - x^2 - y^2 + z^2 \end{bmatrix}. \quad (27)$$

四元数的乘法：四元数的乘法对应于三维空间中旋转的合成。设 $\mathbf{q}_1$ 表示旋转 R_1 ， $\mathbf{q}_2$ 表示旋转 R_2 ，则 $\mathbf{q} = \mathbf{q}_1 \times \mathbf{q}_2$ 表示先执行 R_2 再执行 R_1 的组合旋转（注意乘法顺序与矩阵乘法一致）。

在 Unity 引擎中，旋转表示的转换关系为：界面中显示欧拉角（直观），内部存储使用四元数（计算稳定），进行具体坐标变换时转换为矩阵（便于批量乘法运算）。

5.6 旋转运动

旋转运动使用四元数 $\mathbf{q}$ 表示旋转（例如当前时刻局部坐标系到世界坐标系的转换），使用三维向量 $\boldsymbol{\omega}$ 表示**角速度**。角速度 $\boldsymbol{\omega}$ 的方向表示旋转轴的方向，模值表示角速度的数值大小。

旋转运动的状态更新公式：

$$\boldsymbol{\omega}^{[1]} = \boldsymbol{\omega}^{[0]} + \Delta t \mathbf{I}^{-1} \boldsymbol{\tau}^{[0]}, \quad (28)$$

四元数的更新公式：

$$\mathbf{q}^{[1]} = \mathbf{q}^{[0]} + \Delta \mathbf{q}, \quad \Delta \mathbf{q} = \left[0, \frac{\boldsymbol{\omega}^{[1]}}{2} \right] \times \mathbf{q}^{[0]}. \quad (29)$$

其中 $\boldsymbol{\tau}$ 为力矩， $\mathbf{I}$ 为惯性张量。四元数更新后通常需要进行归一化以保持单位长度。

平移运动与旋转运动的对应关系如表所示：

平移运动		旋转运动	
速度	$\mathbf{v}$	角速度	$\boldsymbol{\omega}$
位置	$\mathbf{x}$	四元数	$\mathbf{q}$
质量	M	惯性张量	$\mathbf{I}$
力	$\mathbf{f}$	力矩	$\boldsymbol{\tau}$

平移运动公式:

$$\mathbf{v}^{[1]} = \mathbf{v}^{[0]} + \Delta t \mathbf{M}^{-1} \mathbf{f}^{[0]}, \quad (30)$$

$$\mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t \mathbf{v}^{[1]}. \quad (31)$$

旋转运动公式:

$$\boldsymbol{\omega}^{[1]} = \boldsymbol{\omega}^{[0]} + \Delta t (\mathbf{I}^{[0]})^{-1} \boldsymbol{\tau}^{[0]}, \quad (32)$$

$$\mathbf{q}^{[1]} = \mathbf{q}^{[0]} + \left[0, \frac{\boldsymbol{\omega}^{[1]}}{2} \right] \times \mathbf{q}^{[0]}. \quad (33)$$

6 刚体动力学

6.0.1 力矩

力矩是在旋转的情形下与力等价的物理量, 它描述了由力引起的旋转趋势。

设作用在刚体上第 i 个点的力为 $\mathbf{f}_i$, 该点在局部坐标系中的位置为 $\mathbf{r}_i$, 则该力对质心产生的力矩为

$$\boldsymbol{\tau}_i = (\mathbf{R}\mathbf{r}_i) \times \mathbf{f}_i. \quad (34)$$

其中 $\mathbf{R}\mathbf{r}_i$ 为该点在全局坐标系中相对于质心的位置向量。

力矩向量 $\boldsymbol{\tau}_i$ 垂直于 $(\mathbf{R}\mathbf{r}_i)$ 和 $\mathbf{f}_i$ 两个向量所张成的平面, 其大小与两个向量的长度以及它们夹角的正弦成正比:

$$|\boldsymbol{\tau}_i| = |\mathbf{R}\mathbf{r}_i| |\mathbf{f}_i| \sin \theta. \quad (35)$$

其中 θ 为 $\mathbf{R}\mathbf{r}_i$ 与 $\mathbf{f}_i$ 之间的夹角。

总力矩为各力矩之和:

$$\boldsymbol{\tau} = \sum_i \boldsymbol{\tau}_i = \sum_i (\mathbf{R}\mathbf{r}_i) \times \mathbf{f}_i. \quad (36)$$

力 $\mathbf{f}$ 是平移运动中的推动量，力矩 $\boldsymbol{\tau}$ 是旋转运动中的推动量；质量 M 是平移运动的阻力，惯性张量 $\mathbf{I}$ 是旋转运动的阻力。

6.0.2 惯性张量

惯性张量是描述刚体对旋转运动阻力的物理量。在局部坐标系（参考坐标系）中，惯性张量的表达式为

$$\mathbf{I}_{\text{ref}} = \sum_i m_i (\mathbf{r}_i^T \mathbf{r}_i \mathbf{1} - \mathbf{r}_i \mathbf{r}_i^T), \quad (37)$$

其中 $\mathbf{1}$ 为 3×3 单位矩阵， m_i 为第 i 个质点的质量， $\mathbf{r}_i$ 为该质点在局部坐标系中的位置向量。

在连续情况下：

$$\mathbf{I}_{\text{ref}} = \int_V \rho(\mathbf{r}) (\mathbf{r}^T \mathbf{r} \mathbf{1} - \mathbf{r} \mathbf{r}^T) dV. \quad (38)$$

6.0.3 惯性张量的变换

类似于质量，惯性张量描述了由力矩引起的旋转趋势的阻力。但与质量不同的是，惯性张量并非恒定不变，它随着刚体的旋转而变化。

在当前状态（全局坐标系）下，惯性张量由参考坐标系中的惯性张量通过旋转矩阵相似变换得到：

$$\mathbf{I} = \sum_i m_i \left((\mathbf{R} \mathbf{r}_i)^T (\mathbf{R} \mathbf{r}_i) \mathbf{1} - (\mathbf{R} \mathbf{r}_i) (\mathbf{R} \mathbf{r}_i)^T \right) \quad (39)$$

$$= \sum_i m_i \left(\mathbf{r}_i^T \mathbf{R}^T \mathbf{R} \mathbf{r}_i \mathbf{1} - \mathbf{R} \mathbf{r}_i \mathbf{r}_i^T \mathbf{R}^T \right) \quad (40)$$

$$= \sum_i m_i \mathbf{R} \left(\mathbf{r}_i^T \mathbf{r}_i \mathbf{1} - \mathbf{r}_i \mathbf{r}_i^T \right) \mathbf{R}^T \quad (41)$$

$$= \mathbf{R} \mathbf{I}_{\text{ref}} \mathbf{R}^T. \quad (42)$$

由于 $\mathbf{R}$ 为正交矩阵（ $\mathbf{R}^T \mathbf{R} = \mathbf{1}$ ），上式中 $(\mathbf{R} \mathbf{r}_i)^T (\mathbf{R} \mathbf{r}_i) = \mathbf{r}_i^T \mathbf{r}_i$ 。

惯性张量是一个矩阵，其逆矩阵表示物体对旋转的阻力（类似于质量对平移运动的阻力）。同一个刚体绕不同轴旋转时，旋转阻力不同，这正是惯性张量为矩阵而非标量的原因。

6.1 刚体动力学

角速度更新公式可由力矩与角加速度的关系 $\boldsymbol{\tau} = \mathbf{I}\boldsymbol{\alpha}$ 推导得到：

$$\boldsymbol{\omega}^{[1]} = \boldsymbol{\omega}^{[0]} + \Delta t (\mathbf{I}^{[0]})^{-1} \boldsymbol{\tau}^{[0]}. \quad (43)$$

四元数的微分方程为

$$\frac{d\mathbf{q}}{dt} = \left[0, \frac{\boldsymbol{\omega}}{2} \right] \times \mathbf{q}. \quad (44)$$

用显式欧拉方法离散化得到四元数更新公式：

$$\mathbf{q}^{[1]} = \mathbf{q}^{[0]} + \left[0, \frac{\boldsymbol{\omega}^{[1]}}{2} \right] \times \mathbf{q}^{[0]}. \quad (45)$$

四元数更新后通常需要归一化，以消除数值误差导致的模长漂移。

刚体物理模拟完整流程：

在刚体物理模拟中，需要更新的物理状态量为 $\mathbf{s} = \{\mathbf{v}, \mathbf{x}, \boldsymbol{\omega}, \mathbf{q}\}$ 。每步模拟的完整流程如下：

1. 从四元数计算旋转矩阵： $\mathbf{R} \leftarrow \text{Matrix.Rotate}(\mathbf{q})$;
2. 对每个施力点 i 计算力矩： $\boldsymbol{\tau}_i \leftarrow (\mathbf{R}\mathbf{r}_i) \times \mathbf{f}_i$;
3. 计算总力矩： $\boldsymbol{\tau} \leftarrow \sum \boldsymbol{\tau}_i$;
4. 对每个施力点 i 计算外力： $\mathbf{f}_i \leftarrow \text{Force}(\mathbf{x}_i, \mathbf{v}_i)$;
5. 计算合外力： $\mathbf{f} \leftarrow \sum \mathbf{f}_i$;
6. 更新惯性张量： $\mathbf{I} \leftarrow \mathbf{R}\mathbf{I}_{\text{ref}}\mathbf{R}^T$;
7. 更新线速度： $\mathbf{v} \leftarrow \mathbf{v} + \Delta t \mathbf{M}^{-1} \mathbf{f}$;
8. 更新角速度： $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} + \Delta t \mathbf{I}^{-1} \boldsymbol{\tau}$;
9. 更新位置： $\mathbf{x} \leftarrow \mathbf{x} + \Delta t \mathbf{v}$;
10. 更新四元数： $\mathbf{q} \leftarrow \mathbf{q} + \left[0, \frac{\boldsymbol{\omega}}{2} \right] \times \mathbf{q}$ ，并归一化。

7 刚体碰撞

研究刚体碰撞问题，首先研究如何检测刚体碰撞。对于复杂几何物体进行**逐点碰撞检测**。刚体上各顶点在全局坐标系中的位置由局部坐标系到世界坐标系的转换得到：

$$\mathbf{x}_i \leftarrow \mathbf{x} + \mathbf{R}\mathbf{r}_i. \quad (46)$$

可见检测刚体碰撞是较为容易的，只是对刚体的每一个点做检测即可。之后要讨论的内容就是刚体的碰撞响应问题了。

7.1 有符号距离函数

有符号距离函数 (Signed Distance Function, SDF) 定义点 $\mathbf{x}$ 到某一曲面的有符号距离。符号的正负对应于点 $\mathbf{x}$ 位于曲面的外侧还是内侧：

$$\phi(\mathbf{x}) \begin{cases} > 0 & (\text{点在曲面外侧}) \\ = 0 & (\text{点在曲面上}) \\ < 0 & (\text{点在曲面内侧}) \end{cases} \quad (47)$$

曲面的法线方向由 $\phi(\mathbf{x})$ 的梯度给出： $\mathbf{N} = \nabla \phi(\mathbf{x})$ 。

常见的有符号距离函数示例：

- 平面 (法线 $\mathbf{n}$ 过点 $\mathbf{p}$): $\phi(\mathbf{x}) = (\mathbf{x} - \mathbf{p}) \cdot \mathbf{n}$;
- 球 (球心 $\mathbf{c}$, 半径 r): $\phi(\mathbf{x}) = |\mathbf{x} - \mathbf{c}| - r$;
- 胶囊体 (轴线过点 $\mathbf{p}$, 方向 $\mathbf{n}$, 半径 r): $\phi(\mathbf{x}) = |\mathbf{x} - \mathbf{p}|_2 - |(\mathbf{x} - \mathbf{p}) \cdot \mathbf{n}| - r$, 其中 $|\cdot|_2$ 表示欧几里得范数。

7.2 惩罚方法

惩罚方法是一种处理碰撞的方法，通过在物体穿透时施加惩罚力来阻止穿透。

二次惩罚方法: 惩罚势能类似于弹簧的弹性势能，定义为

$$E_p(\mathbf{x}) = \frac{k}{2} (\min\{0, \phi(\mathbf{x})\})^2. \quad (48)$$

对应的惩罚力为

$$\mathbf{f} = -k \min\{0, \phi(\mathbf{x})\} \mathbf{N} = -k \min\{0, \phi(\mathbf{x})\} \nabla \phi(\mathbf{x}), \quad (49)$$

其中 k 为惩罚强度系数。当 $\phi(\mathbf{x}) \geq 0$ 时不施加力；当 $\phi(\mathbf{x}) < 0$ （穿透）时施加指向外侧的力。

带缓冲区的二次惩罚方法：为了减少穿透现象，可引入缓冲带 ε 。当 $\phi(\mathbf{x}) < \varepsilon$ 时即开始施加力：

$$\mathbf{f} = k(\varepsilon - \phi(\mathbf{x})) \mathbf{N}. \quad (50)$$

k 值较小时，穿透依然可能发生； k 值较大时，可能发生过冲（overshooting）现象。

对数障碍函数惩罚方法：对数障碍函数惩罚方法中力随着距离变小而增大。惩罚力恒为

$$\mathbf{f} = \frac{\rho}{\phi(\mathbf{x})} \mathbf{N}, \quad (51)$$

其中 ρ 为障碍强度。该方法假设 $\phi(\mathbf{x}) < 0$ （穿透）永远不会发生，需要通过调整时间步长 Δt 来保证这一点。对数障碍函数的力在物体接近曲面时迅速增大，从而有效防止穿透。

7.3 冲量法

冲量法假设碰撞会瞬间改变物体的位置和速度，是另一种处理碰撞的方法。

位置修正：当检测到穿透（ $\phi(\mathbf{x}) < 0$ ）时，首先修正位置：

$$\mathbf{x}^{\text{new}} \leftarrow \mathbf{x} - \phi(\mathbf{x}) \nabla \phi(\mathbf{x}) = \mathbf{x} + \phi(\mathbf{x}) \mathbf{N}. \quad (52)$$

速度修正：位置修正后，还需要对速度进行修正。设顶点 $\mathbf{x}_i$ 处的法线为 $\mathbf{N}$ ，顶点速度为 $\mathbf{v}_i$ 。

首先将速度分解为法线方向和切线方向：

$$\mathbf{v}_{N,i} = (\mathbf{v}_i \cdot \mathbf{N}) \mathbf{N}, \quad (53)$$

$$\mathbf{v}_{T,i} = \mathbf{v}_i - \mathbf{v}_{N,i}. \quad (54)$$

若 $\mathbf{v}_i \cdot \mathbf{N} < 0$ （顶点正在朝曲面内部运动），则修正法线速度：

$$\mathbf{v}_{N,i}^{\text{new}} \leftarrow -\mu_N \mathbf{v}_{N,i}, \quad (55)$$

其中 μ_N 为恢复系数。

切线速度受摩擦力影响。设摩擦系数为 μ ，则首先计算缩放系数⁵

$$a \leftarrow \max\left(1 - \mu_T \frac{1 + \mu_N}{|\mathbf{v}_{T,i}|} |\mathbf{v}_{N,i}|, 0\right). \quad (56)$$

切线速度更新为

$$\mathbf{v}_{T,i}^{\text{new}} \leftarrow a \mathbf{v}_{T,i}. \quad (57)$$

合成后的速度为

$$\mathbf{v}_i^{\text{new}} = \mathbf{v}_{N,i}^{\text{new}} + \mathbf{v}_{T,i}^{\text{new}}. \quad (58)$$

以上推导结果是针对于发生碰撞的那一个顶点而言的，但是我们无法直接修改某个顶点的 $\mathbf{x}_i$ 和 $\mathbf{v}_i$ ，因为它们并非状态变量，而是通过全局状态变量（质心位置 $\mathbf{x}$ 、线速度 $\mathbf{v}$ 、角速度 $\boldsymbol{\omega}$ ）间接确定的。**所以，我们的思路是通过施加冲量来修改全局状态变量，而冲量由碰撞顶点的速度变化来确定。**

刚体上第 i 个顶点的速度由线速度和角速度共同贡献：

$$\mathbf{v}_i = \mathbf{v} + \boldsymbol{\omega} \times (\mathbf{R}\mathbf{r}_i). \quad (59)$$

如果在顶点 $\mathbf{v}_i$ 施加一个冲量 $\mathbf{j}$ ($\mathbf{j} = \mathbf{f}\Delta t$)，则全局状态变量更新为

$$\mathbf{v}^{\text{new}} \leftarrow \mathbf{v} + \frac{\mathbf{j}}{M}, \quad (60)$$

$$\boldsymbol{\omega}^{\text{new}} \leftarrow \boldsymbol{\omega} + \mathbf{I}^{-1}((\mathbf{R}\mathbf{r}_i) \times \mathbf{j}). \quad (61)$$

施加冲量后，该顶点的速度变为

$$\mathbf{v}_i^{\text{new}} = \mathbf{v}^{\text{new}} + \boldsymbol{\omega}^{\text{new}} \times (\mathbf{R}\mathbf{r}_i) \quad (62)$$

$$= \mathbf{v} + \frac{\mathbf{j}}{M} + (\boldsymbol{\omega} + \mathbf{I}^{-1}((\mathbf{R}\mathbf{r}_i) \times \mathbf{j})) \times (\mathbf{R}\mathbf{r}_i). \quad (63)$$

叉积 $\mathbf{r} \times \mathbf{q}$ 可以转化为矩阵乘法的形式 $\mathbf{r}^* \mathbf{q}$ ，其中 $\mathbf{r}^*$ 为 $\mathbf{r}$ 的反对称矩阵：

$$\mathbf{r}^* = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \quad (64)$$

即

$$\mathbf{r} \times \mathbf{q} = \begin{bmatrix} r_y q_z - r_z q_y \\ r_z q_x - r_x q_z \\ r_x q_y - r_y q_x \end{bmatrix} = \mathbf{r}^* \mathbf{q}. \quad (65)$$

⁵这个式子取最大值是防止 a 小于 0，另外一个上界由 $\|\mathbf{v}_{T,i}^{\text{new}} - \mathbf{v}_{T,i}\| \leq \mu_t \|\mathbf{v}_{T,i}^{\text{new}} - \mathbf{v}_{T,i}\|$ 确定。

利用叉积矩阵形式，顶点速度变化可写为

$$\mathbf{v}_i^{\text{new}} = \mathbf{v}_i^{\text{old}} + \frac{\mathbf{j}}{M} - (\mathbf{R}\mathbf{r}_i)^* \mathbf{I}^{-1} (\mathbf{R}\mathbf{r}_i)^* \mathbf{j}. \quad (66)$$

定义碰撞矩阵 $\mathbf{K}$:

$$\mathbf{K} \leftarrow \frac{1}{M} \mathbf{1} - (\mathbf{R}\mathbf{r}_i)^* \mathbf{I}^{-1} (\mathbf{R}\mathbf{r}_i)^*. \quad (67)$$

则速度变化为

$$\mathbf{v}_i^{\text{new}} - \mathbf{v}_i^{\text{old}} = \mathbf{K}\mathbf{j}. \quad (68)$$

7.3.1 冲量法完整算法

冲量法处理刚体碰撞的完整流程如下:

1. 对每个顶点，计算全局坐标: $\mathbf{x}_i \leftarrow \mathbf{x} + \mathbf{R}\mathbf{r}_i$;
2. 检测是否穿透: 若 $\phi(\mathbf{x}_i) < 0$ ，则发生碰撞;
3. 计算顶点速度: $\mathbf{v}_i \leftarrow \mathbf{v} + \boldsymbol{\omega} \times (\mathbf{R}\mathbf{r}_i)$;
4. 检测是否朝内运动: 若 $\mathbf{v}_i \cdot \mathbf{N} < 0$ ，则需要处理;
5. 计算期望速度 $\mathbf{v}_i^{\text{new}}$:
 - 法线分量: $\mathbf{v}_{N,i} \leftarrow (\mathbf{v}_i \cdot \mathbf{N}) \mathbf{N}$, $\mathbf{v}_{N,i}^{\text{new}} \leftarrow -\mu \mathbf{v}_{N,i}$;
 - 切线分量: $\mathbf{v}_{T,i} \leftarrow \mathbf{v}_i - \mathbf{v}_{N,i}$;
 - 缩放系数: $a \leftarrow \max\left(1 - \mu \frac{1 + \mu}{|\mathbf{v}_{T,i}|} |\mathbf{v}_{N,i}|, 0\right)$;
 - 切线速度更新: $\mathbf{v}_{T,i}^{\text{new}} \leftarrow a \mathbf{v}_{T,i}$;
6. 计算碰撞矩阵: $\mathbf{K} \leftarrow \frac{1}{M} \mathbf{1} - (\mathbf{R}\mathbf{r}_i)^* \mathbf{I}^{-1} (\mathbf{R}\mathbf{r}_i)^*$;
7. 计算冲量: $\mathbf{j} \leftarrow \mathbf{K}^{-1} (\mathbf{v}_i^{\text{new}} - \mathbf{v}_i)$;
8. 更新全局状态:

$$\mathbf{v} \leftarrow \mathbf{v} + \frac{\mathbf{j}}{M}, \quad (69)$$

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} + \mathbf{I}^{-1} ((\mathbf{R}\mathbf{r}_i) \times \mathbf{j}). \quad (70)$$

8 基于物理的布料模拟

8.1 布料模拟的基本概念

布料是软体模拟的一个重要子类。由于布料具有**薄壳特性**（即二维曲面嵌入三维空间中），其力学行为与三维软体存在本质差异，因此需要单独归类研究。布料的厚度相对于其表面尺寸可以忽略不计，这种几何特性决定了其变形主要由面内拉伸、面外弯曲以及剪切行为主导。在工程实践中，布料模拟通常采用两种建模方式：**质点-弹簧系统**和**有限元壳体模型**。

8.2 布料模拟的常用方法

布料模拟根据应用场景对物理精度和计算效率的不同要求，形成了三种主流方法。

8.2.1 质点-弹簧系统

核心思想：将布料离散化为由质点构成的网格结构，相邻质点之间通过弹簧连接。系统通过模拟弹簧的弹力、阻尼力以及外力（如重力）的作用，利用数值积分方法迭代更新每个质点的位置和速度。

优点：实现简单，计算效率高，适合实时应用（如游戏引擎中的布料效果）。

缺点：弹簧刚度参数需要手动调整，缺乏直接的物理对应关系；难以精确模拟复杂的布料行为，如各向异性（经纬方向力学性质不同）和真实拉伸特性。

应用：实时动画、游戏中的简单布料效果。

8.2.2 有限元法

核心思想：将布料视为连续介质，将其划分为若干小单元（通常为三角形或四边形单元）。基于连续介质力学理论，计算每个单元的应力、应变和弹性势能，通过求解偏微分方程来模拟布料的变形过程。

优点：物理精度高，能够准确模拟复杂材料行为，包括各向异性、非线性本构关系等。

缺点：计算量大，系统矩阵的组装和求解开销高，难以满足实时应用的需求。

应用：电影特效、高精度工业仿真。

8.2.3 基于位置的动力学 (Position-Based Dynamics, PBD)

核心思想：直接通过几何约束（如距离约束、弯曲约束、拉伸约束）来调整质点的位置，跳过传统力学中力的计算和积分过程。通过迭代投影的方式使质点位置逐步满足所有约束条件。

优点：计算高效且数值稳定，适合实时交互应用（如 VR/AR 游戏中的布料模拟）。

缺点：物理真实性较低，模拟结果依赖于约束参数的设置，难以保证与真实物理行为的一致性。

应用：实时布料模拟、交互式场景。

9 质点-弹簧模型

9.1 理想弹簧的力学基础

9.1.1 胡克定律与弹簧势能

理想弹簧遵循**胡克定律**：弹簧产生的恢复力与其相对于自然长度的形变成正比，方向指向恢复原长的方向。

设弹簧两端点位置为 $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^d$ ($d = 2$ 或 3)，自然长度 (rest length) 为 s ，弹簧刚度系数为 k 。定义当前长度为

$$\ell = \|\mathbf{x}_1 - \mathbf{x}_0\| = \sqrt{(\mathbf{x}_1 - \mathbf{x}_0)^T (\mathbf{x}_1 - \mathbf{x}_0)}. \quad (71)$$

弹簧的**弹性势能**定义为

$$E(\mathbf{x}_0, \mathbf{x}_1) = \frac{k}{2}(\ell - s)^2 = \frac{k}{2}(\|\mathbf{x}_1 - \mathbf{x}_0\| - s)^2. \quad (72)$$

弹簧作用于端点 $\mathbf{x}_0$ 的**弹性力**为势能的负梯度：

$$\mathbf{f}_0(\mathbf{x}_0, \mathbf{x}_1) = -\nabla_{\mathbf{x}_0} E = -k(\ell - s) \frac{\mathbf{x}_0 - \mathbf{x}_1}{\ell} = k(\ell - s) \frac{\mathbf{x}_1 - \mathbf{x}_0}{\ell}. \quad (73)$$

同理，作用于端点 $\mathbf{x}_1$ 的力为 $\mathbf{f}_1 = -\mathbf{f}_0$ ，满足牛顿第三定律。

9.1.2 梯度与 Hessian 的定义

对于一个定义在 $\mathbb{R}^n$ 上的标量函数 $f(\mathbf{x})$ ，其**梯度** (gradient) 是一个向量，包含函数在所有自变量方向上的偏导数：

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T. \quad (74)$$

对于一个二次可微函数 $f(\mathbf{x})$ ，其**Hessian 矩阵**是一个 $n \times n$ 的对称矩阵，包含所有二阶偏导数：

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (75)$$

9.1.3 辅助函数的梯度与 Hessian

为了推导弹簧系统的梯度与 Hessian，首先计算以下辅助函数。

设 $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0$ ， $\ell = \|\mathbf{x}\|$ ，考虑 $\ell^2 = \mathbf{x}^T \mathbf{x}$ ：

$$\nabla_{\mathbf{x}}(\ell^2) = \nabla_{\mathbf{x}}(\mathbf{x}^T \mathbf{x}) = 2\mathbf{x}^T, \quad \nabla_{\mathbf{x}}^2(\ell^2) = 2\mathbf{I}. \quad (76)$$

对于 $\ell = \sqrt{\mathbf{x}^T \mathbf{x}}$ ，利用链式法则：

$$\nabla_{\mathbf{x}} \ell = \nabla_{\mathbf{x}} \sqrt{\mathbf{x}^T \mathbf{x}} = \frac{1}{2\ell} \cdot 2\mathbf{x} = \frac{\mathbf{x}^T}{\ell}. \quad (77)$$

9.2 弹簧的梯度与 Hessian 推导

9.2.1 一端位置固定的弹簧

设弹簧一端 $\mathbf{x}_0$ 固定，另一端 $\mathbf{x}_1$ 为变量。

能量：

$$E(\mathbf{x}_1) = \frac{k}{2} (\|\mathbf{x}_1 - \mathbf{x}_0\| - s)^2. \quad (78)$$

受力:

$$\mathbf{f}(\mathbf{x}_1) = -\nabla_{\mathbf{x}_1} E = -k(\|\mathbf{x}_1 - \mathbf{x}_0\| - s)(\nabla_{\mathbf{x}_1} \|\mathbf{x}_1 - \mathbf{x}_0\|)^T \quad (79)$$

$$= -k(\|\mathbf{x}_1 - \mathbf{x}_0\| - s) \frac{\mathbf{x}_1 - \mathbf{x}_0}{\|\mathbf{x}_1 - \mathbf{x}_0\|}. \quad (80)$$

Tangent Stiffness Matrix (Hessian 矩阵)⁶:

$$\mathbf{K}(\mathbf{x}_1) = -\nabla_{\mathbf{x}_1} \mathbf{f} = \nabla_{\mathbf{x}_1}^2 E. \quad (81)$$

对 $\mathbf{f}$ 求梯度, 下面为了书写的简便, 令 $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0, \ell = \|\mathbf{x}\|$:

$$\nabla_{\mathbf{x}_1} \left(\frac{\mathbf{x}}{\ell} \right) = \frac{1}{\ell} \mathbf{I} - \frac{1}{\ell^2} \mathbf{x} \left(\frac{\mathbf{x}}{\ell} \right)^T \quad (82)$$

$$= \frac{1}{\ell} \mathbf{I} - \frac{1}{\ell^3} \mathbf{x} \mathbf{x}^T. \quad (83)$$

因此:

$$\nabla_{\mathbf{x}_1} \mathbf{f} = k \left[\left(1 - \frac{s}{\ell} \right) \mathbf{I} + \frac{s}{\ell^3} \mathbf{x} \mathbf{x}^T \right]. \quad (84)$$

最终得到 tangent stiffness matrix:

$$\boxed{\mathbf{K}(\mathbf{x}_1) = -\nabla_{\mathbf{x}_1} \mathbf{f} = k \left[\left(\frac{s}{\ell} - 1 \right) \mathbf{I} - \frac{s}{\ell^3} \mathbf{x} \mathbf{x}^T \right]}. \quad (85)$$

9.2.2 两端位置均为变量的弹簧

设弹簧两端 $\mathbf{x}_0, \mathbf{x}_1$ 均为变量。定义 $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0, \ell = \|\mathbf{x}\|$, s 为弹簧原始长度。

能量:

$$E(\mathbf{x}_0, \mathbf{x}_1) = \frac{k}{2} (\|\mathbf{x}_1 - \mathbf{x}_0\| - s)^2. \quad (86)$$

力:

作用于 $\mathbf{x}_0$ 的力:

$$\mathbf{f}_0 = -\nabla_{\mathbf{x}_0} E = -k(\ell - s) \frac{\mathbf{x}_0 - \mathbf{x}_1}{\ell} = k(\ell - s) \frac{\mathbf{x}_1 - \mathbf{x}_0}{\ell}. \quad (87)$$

作用于 $\mathbf{x}_1$ 的力:

$$\mathbf{f}_1 = -\nabla_{\mathbf{x}_1} E = -k(\ell - s) \frac{\mathbf{x}_1 - \mathbf{x}_0}{\ell} = -\mathbf{f}_0. \quad (88)$$

⁶最后一节课说这部分推导要考。

Tangent Stiffness Matrix: 将 $\mathbf{x}_0$ 和 $\mathbf{x}_1$ 合并为状态向量 $\mathbf{q} = [\mathbf{x}_0^T, \mathbf{x}_1^T]^T$, Hessian 具有 $2d \times 2d$ 的分块结构。

令 $\hat{\mathbf{x}} = \mathbf{x}/\ell = (\mathbf{x}_1 - \mathbf{x}_0)/\ell$ 为单位方向向量。定义投影矩阵 $\mathbf{P} = \mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T$ (满足 $\mathbf{P}\hat{\mathbf{x}} = \mathbf{0}$)，则：

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{00} & \mathbf{K}_{01} \\ \mathbf{K}_{10} & \mathbf{K}_{11} \end{bmatrix} = k \begin{bmatrix} \frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T - \mathbf{P} & -\frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T + \mathbf{P} \\ -\frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T + \mathbf{P} & \frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T - \mathbf{P} \end{bmatrix}. \quad (89)$$

利用 $\mathbf{P} = \mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T$ ，上式可简化为：

$$\mathbf{K}_{00} = \mathbf{K}_{11} = -\mathbf{K}_{01} = -\mathbf{K}_{10} = k \left[\left(\frac{s}{\ell} - 1 \right) \hat{\mathbf{x}}\hat{\mathbf{x}}^T - \mathbf{I} + \hat{\mathbf{x}}\hat{\mathbf{x}}^T \right] = k \left[\frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T - \mathbf{I} \right]. \quad (90)$$

其中对角块 $\mathbf{K}_{ii}$ 描述第 i 个端点处力的变化对自身位置的影响，非对角块 $\mathbf{K}_{ij}$ ($i \neq j$) 描述第 j 个端点位移对第 i 个端点处力的影响。

9.3 多个弹簧的系统

当系统中存在多个弹簧时，根据能量的可叠加性，系统的总能量等于各弹簧能量之和：

$$E_{\text{total}} = \sum_{e=0}^{n-1} E_e = \sum_{e=0}^{n-1} \frac{k_e}{2} (\|\mathbf{x}_{j_e} - \mathbf{x}_{i_e}\| - s_e)^2. \quad (91)$$

其中 n 为弹簧总数，第 e 条弹簧连接顶点 i_e 和 j_e ，刚度系数为 k_e ，自然长度为 s_e 。

系统作用在第 i 个顶点上的总力为所有与该顶点相连弹簧产生的力的叠加：

$$\mathbf{f}_i = -\nabla_{\mathbf{x}_i} E_{\text{total}} = - \sum_{e \in \mathcal{N}(i)} k_e (\ell_e - s_e) \frac{\mathbf{x}_i - \mathbf{x}_{j_e}}{\ell_e}. \quad (92)$$

其中 $\mathcal{N}(i)$ 表示与顶点 i 相连的所有弹簧的索引集合。

9.4 弹簧网格结构

9.4.1 结构化的弹簧网格

对于规则网格（如矩形网格），通常使用三类弹簧来模拟布料的不同力学行为：⁷

- **Horizontal and Vertical Springs (结构弹簧)**：连接水平和垂直相邻的质点，用于抵抗拉伸变形。
- **Diagonal Springs (剪切弹簧)**：连接对角线方向的相邻质点，用于抵抗剪切变形。
- **Bending Springs (弯曲弹簧)**：连接相隔一个质点的位置（如 (i, j) 与 $(i + 2, j)$ ），用于抵抗面外弯曲变形。

9.4.2 非结构化的弹簧网格

对于不规则的三角网格，可以通过以下方式转化为弹簧网络：

- **边弹簧**：三角网格的每条边对应一个弹簧，用于抵抗边方向上的拉伸。
- **弯曲弹簧**：对于每一对共享边的相邻三角形，添加一个弯曲约束（或等效的弯曲弹簧），用于抵抗面外弯曲。

9.5 三角网格的表示与拓扑构建

9.5.1 基本表示方法

三角网格使用两种基本数据结构表示：

顶点列表：存储所有顶点的三维坐标

$$\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n-1}\}, \quad \mathbf{v}_i \in \mathbb{R}^3. \quad (93)$$

三角形列表：存储每个三角形对应的三个顶点索引

$$\{\{i_0, j_0, k_0\}, \{i_1, j_1, k_1\}, \dots\}. \quad (94)$$

每个三角形有三条边，但相邻三角形共享边，因此边存在重复。

⁷这部分在 PPT 上是英文呈现的，中文翻译为笔者自行翻译，可能不准确，建议以英文描述为准。

9.5.2 拓扑构建方法

拓扑构建的核心在于对三角形边三元组进行排序和去重。每个三元组包含三个要素：

- 边的起始顶点索引 (index 0)
- 边的终止顶点索引 (index 1), 要求起始索引小于终止索引 (index 0 < index 1)
- 该边所属的三角形索引

构建步骤：

1. 遍历所有三角形，为每条边生成三元组 $\{v_{\min}, v_{\max}, t\}$ 。
2. 将所有三元组按字典序排序（先比较 $v_{\min}$ ，再比较 $v_{\max}$ ）。
3. 排序后，相同的边会相邻出现。去重后得到**唯一的边列表**。
4. 对于弯曲弹簧的构建：记录共享同一条边的三角形对，即得到**相邻三角形列表**。

10 质点-弹簧系统的数值计算方法

10.1 弹簧系统的显式积分

10.1.1 显式欧拉积分算法

显式积分采用前向欧拉方法，利用当前时刻的状态直接计算下一时刻的状态。设顶点质量为 m_i ，时间步长为 Δt 。

算法流程：

1. 计算弹簧力：对每条弹簧 $e = (i, j)$ ，计算其产生的力并累加到对应顶点。
2. 更新速度（显式）： $\mathbf{v}_i^{t+1} \leftarrow \mathbf{v}_i^t + \frac{\Delta t}{m_i} \mathbf{f}_i$ 。
3. 更新位置（显式）： $\mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+1}$ 。

其中 $\mathbf{f}_i$ 为作用在顶点 i 上的合外力（包括弹簧力、重力、阻尼力等）。

10.1.2 数值稳定性分析

显式积分法存在严重的**数值不稳定性**问题：当弹簧刚度 k 过大或时间步长 Δt 过大时，系统的解会出现过冲现象，导致能量指数增长、模拟发散。

从物理角度理解，显式积分假设在当前时间步内力保持不变，当弹簧刚度大时，恢复力的变化非常迅速，这一假设不再成立。若采用减小 Δt 的简单方案，受限于 CFL 条件，需要极小的步长才能保证稳定，这会导致仿真效率显著降低。

10.2 弹簧系统的隐式积分

10.2.1 隐式积分的基本方程

隐式积分通过对位置和速度均采用隐式离散来求解数值不稳定性问题。核心思想是利用下一时刻的力来计算下一时刻的状态。

速度的隐式更新：

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}^{t+1}). \quad (95)$$

位置的隐式更新：

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t \mathbf{v}^{t+1}. \quad (96)$$

联立上述两式消去 $\mathbf{v}^{t+1}$ ：

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t \mathbf{v}^t + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}^{t+1}). \quad (97)$$

10.2.2 等价优化问题

在假设力场为**保守力场**的前提下，上述隐式方程等价于求解以下优化问题：

$$\boxed{\mathbf{x}^{t+1} = \arg \min_{\mathbf{x}} \Phi(\mathbf{x})} \quad (98)$$

其中目标函数 $\Phi(\mathbf{x})$ 定义为：

$$\boxed{\Phi(\mathbf{x}) = \frac{1}{2\Delta t^2} (\mathbf{x} - \mathbf{x}^t - \Delta t \mathbf{v}^t)^T \mathbf{M} (\mathbf{x} - \mathbf{x}^t - \Delta t \mathbf{v}^t) - (\mathbf{x} - \mathbf{x}^t)^T \mathbf{f}_{\text{ext}} + E_{\text{int}}(\mathbf{x})}. \quad (99)$$

式中各项含义如下：

- 第一项为**惯性项** (inertia term), 惩罚位置偏离显式预测位置的程度, 其中 $\mathbf{M}$ 为质量矩阵。
- 第二项为外力 (如重力) 做的功, $\mathbf{f}_{\text{ext}}$ 为外力。⁸
- 第三项 $E_{\text{int}}(\mathbf{x})$ 为系统的内部弹性势能。

等价性证明: 对 $\Phi(\mathbf{x})$ 求梯度并令其为零:

$$\nabla \Phi(\mathbf{x}) = \frac{1}{\Delta t^2} \mathbf{M}(\mathbf{x} - \mathbf{x}^t - \Delta t \mathbf{v}^t) - \mathbf{f}_{\text{ext}} + \nabla E_{\text{int}}(\mathbf{x}) \quad (100)$$

$$= \frac{1}{\Delta t^2} \mathbf{M}(\mathbf{x} - \mathbf{x}^t - \Delta t \mathbf{v}^t) - \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}(\mathbf{x}) = 0. \quad (101)$$

整理得:

$$\mathbf{x} = \mathbf{x}^t + \Delta t \mathbf{v}^t + \Delta t^2 \mathbf{M}^{-1} [\mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{int}}(\mathbf{x})] = \mathbf{x}^t + \Delta t \mathbf{v}^t + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}). \quad (102)$$

这与隐式积分方程完全一致, 因此两种表述等价。

值得注意的是, 这一方法适用于所有力场为保守力场的物理系统, 不仅限于质点-弹簧系统。

10.3 牛顿迭代方法

10.3.1 一维牛顿法

牛顿迭代法用于求解优化问题 $\min_x g(x)$, 即寻找满足 $g'(x) = 0$ 的极值点。⁹

核心思想: 在当前迭代点 $x^{(n)}$ 附近对 $g(x)$ 进行二阶泰勒展开:

$$g(x) \approx g(x^{(n)}) + g'(x^{(n)})(x - x^{(n)}) + \frac{1}{2}g''(x^{(n)})(x - x^{(n)})^2. \quad (103)$$

对上式关于 x 求导并令其为零, 得到更新方向:

$$\Delta x = -\frac{g'(x^{(n)})}{g''(x^{(n)})}. \quad (104)$$

算法流程:

1. 初始化 $x^{(0)}$ 。

⁸这里给出的是更普适的形式, 比如考虑重力场作用下的质点弹簧系统, PPT 的公式上没有外力做功这一项。

⁹这部分内容在 CSAI-2 中讲过。

2. 对于 $n = 0, 1, 2, \dots$:

- (a) 计算 $\Delta x \leftarrow -\frac{g'(x^{(n)})}{g''(x^{(n)})}$ 。
- (b) 若 $|\Delta x|$ 足够小, 则终止迭代。
- (c) 更新 $x^{(n+1)} \leftarrow x^{(n)} + \Delta x$ 。

收敛性分析: 牛顿法能够找到极值点, 但该极值可能是极小值或极大值。

- 在极小值点 x^* 处, $g''(x^*) > 0$ 。
- 在极大值点 x^* 处, $g''(x^*) < 0$ 。
- 若 $g''(x)$ 处处大于零, 则 $g(x)$ 为严格凸函数, 不存在极大值, 且仅有一个极小值点。

10.3.2 高维牛顿法

将牛顿法应用于高维优化问题 $\min_{\mathbf{x} \in \mathbb{R}^n} g(\mathbf{x})$ 。

在当前迭代点 $\mathbf{x}^{(n)}$ 附近进行二阶泰勒展开:

$$g(\mathbf{x}) \approx g(\mathbf{x}^{(n)}) + \nabla g(\mathbf{x}^{(n)})^T (\mathbf{x} - \mathbf{x}^{(n)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(n)})^T \nabla^2 g(\mathbf{x}^{(n)}) (\mathbf{x} - \mathbf{x}^{(n)}). \quad (105)$$

令梯度为零得到线性方程组:

$$\nabla g(\mathbf{x}) \approx \nabla g(\mathbf{x}^{(n)}) + \nabla^2 g(\mathbf{x}^{(n)}) (\mathbf{x} - \mathbf{x}^{(n)}) = 0. \quad (106)$$

解得更新方向:

$$\Delta \mathbf{x} = -[\nabla^2 g(\mathbf{x}^{(n)})]^{-1} \nabla g(\mathbf{x}^{(n)}). \quad (107)$$

算法流程:

1. 初始化 $\mathbf{x}^{(0)}$ 。
2. 对于 $n = 0, 1, 2, \dots$:
 - (a) 求解 $\nabla^2 g(\mathbf{x}^{(n)}) \Delta \mathbf{x} = -\nabla g(\mathbf{x}^{(n)})$ 。
 - (b) 若 $\|\Delta \mathbf{x}\|$ 足够小, 则终止迭代。
 - (c) 更新 $\mathbf{x}^{(n+1)} \leftarrow \mathbf{x}^{(n)} + \Delta \mathbf{x}$ 。

10.4 基于牛顿法的物理模拟

将牛顿法应用于隐式积分的优化问题。定义目标函数：

$$\Phi(\mathbf{x}) = \frac{1}{2\Delta t^2} (\mathbf{x} - \mathbf{x}^t - \Delta t \mathbf{v}^t)^T \mathbf{M} (\mathbf{x} - \mathbf{x}^t - \Delta t \mathbf{v}^t) + E_{\text{int}}(\mathbf{x}) - \mathbf{x}^T \mathbf{f}_{\text{ext}}. \quad (108)$$

为简化表达，令 $\mathbf{y} = \mathbf{x}^t + \Delta t \mathbf{v}^t$ （显式预测位置），则：

$$\Phi(\mathbf{x}) = \frac{1}{2\Delta t^2} (\mathbf{x} - \mathbf{y})^T \mathbf{M} (\mathbf{x} - \mathbf{y}) + E_{\text{int}}(\mathbf{x}) - \mathbf{x}^T \mathbf{f}_{\text{ext}}. \quad (109)$$

梯度：

$$\nabla \Phi(\mathbf{x}) = \frac{1}{\Delta t^2} \mathbf{M} (\mathbf{x} - \mathbf{y}) + \nabla E_{\text{int}}(\mathbf{x}) - \mathbf{f}_{\text{ext}}. \quad (110)$$

Hessian：

$$\nabla^2 \Phi(\mathbf{x}) = \frac{1}{\Delta t^2} \mathbf{M} + \nabla^2 E_{\text{int}}(\mathbf{x}). \quad (111)$$

牛顿迭代公式：

$$\Delta \mathbf{x} = - \left[\frac{1}{\Delta t^2} \mathbf{M} + \nabla^2 E_{\text{int}}(\mathbf{x}^{(n)}) \right]^{-1} \left(\frac{1}{\Delta t^2} \mathbf{M} (\mathbf{x} - \mathbf{y}) + \nabla E_{\text{int}}(\mathbf{x}) - \mathbf{f}_{\text{ext}} \right). \quad (112)$$

完整算法流程：

1. 初始化： $\mathbf{x}^{(0)} \leftarrow \mathbf{y} = \mathbf{x}^t + \Delta t \mathbf{v}^t$ （或使用其他初始猜测值）。
2. 对于 $n = 0, 1, 2, \dots$ ：
 - (a) 计算梯度： $\mathbf{g} = \nabla \Phi(\mathbf{x}^{(n)})$ 。
 - (b) 计算 Hessian（或近似）： $\mathbf{H} = \nabla^2 \Phi(\mathbf{x}^{(n)})$ 。
 - (c) 求解线性方程组： $\mathbf{H} \Delta \mathbf{x} = -\mathbf{g}$ 。
 - (d) 若 $\|\Delta \mathbf{x}\|$ 足够小，终止迭代。
 - (e) 更新： $\mathbf{x}^{(n+1)} \leftarrow \mathbf{x}^{(n)} + \Delta \mathbf{x}$ 。
3. 最终更新速度： $\mathbf{v}^{t+1} \leftarrow (\mathbf{x}^{t+1} - \mathbf{x}^t) / \Delta t$ 。

11 弹簧系统的 Hessian 矩阵正定性分析

11.1 单弹簧系统 Hessian 的正定性分析

回顾单端固定弹簧的 Hessian：

$$\nabla^2 E = k \left[\left(1 - \frac{s}{\ell} \right) \mathbf{I} + \frac{s}{\ell} \hat{\mathbf{x}} \hat{\mathbf{x}}^T \right]. \quad (113)$$

该矩阵由两项组成。将其改写为投影矩阵的形式更加直观。令 $\mathbf{P} = \mathbf{I} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T$ 为垂直于弹簧方向的投影矩阵（满足 $\mathbf{P}\hat{\mathbf{x}} = \mathbf{0}$ ），则：

$$\nabla^2 E = k \left[\left(1 - \frac{s}{\ell}\right) \mathbf{I} + \frac{s}{\ell} (\mathbf{I} - \mathbf{P}) \right] = k \left[\mathbf{I} - \frac{s}{\ell} \mathbf{P} \right]. \quad (114)$$

或者使用原始形式：

$$\nabla^2 E = \underbrace{k \left(1 - \frac{s}{\ell}\right) \mathbf{I}}_{\text{各向同性部分}} + \underbrace{k \frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T}_{\text{轴向部分}}. \quad (115)$$

对于任意非零向量 $\mathbf{v} \in \mathbb{R}^d$ ，考察二次型：

$$\mathbf{v}^T (\nabla^2 E) \mathbf{v} = k \left[\left(1 - \frac{s}{\ell}\right) \|\mathbf{v}\|^2 + \frac{s}{\ell} (\mathbf{v}^T \hat{\mathbf{x}})^2 \right]. \quad (116)$$

利用 $(\mathbf{v}^T \hat{\mathbf{x}})^2 \leq \|\mathbf{v}\|^2$ （柯西-施瓦茨不等式）：

$$\mathbf{v}^T (\nabla^2 E) \mathbf{v} \geq k \left(1 - \frac{s}{\ell}\right) \|\mathbf{v}\|^2. \quad (117)$$

- 当弹簧被**拉伸**时（ $\ell > s$ ），有 $1 - \frac{s}{\ell} > 0$ ，此时 $\nabla^2 E$ 是**正定**的。
- 当弹簧被**压缩**时（ $\ell < s$ ），有 $1 - \frac{s}{\ell} < 0$ ，此时 Hessian 矩阵**可能失去正定性**。

11.2 多弹簧系统 Hessian 的正定性问题

整个弹簧系统的 Hessian 矩阵为

$$\nabla^2 \Phi = \frac{1}{\Delta t^2} \mathbf{M} + \nabla^2 E_{\text{int}}. \quad (118)$$

其中质量矩阵项 $\frac{1}{\Delta t^2} \mathbf{M}$ 是正定的，但当某些弹簧被压缩时，内部能量的 Hessian $\nabla^2 E_{\text{int}}$ 可能在某些方向上为负定，导致系统总 Hessian 失去正定性。

当 Hessian 矩阵不正定时，优化问题可能存在多个局部极小值，对应系统的不同平衡状态。牛顿法在这种情况下可能收敛到非期望的局部极小值，或者由于 Hessian 的非正定性导致线性求解器失效。

注意：一维弹簧系统的势能函数为 $E(x) = \frac{k}{2}(x - s)^2$ ，其二阶导数 $E''(x) = k > 0$ 恒为正。因此一维系统不存在正定性问题。二维和三维系统中正定性问题的产生根源在于长度函数 $\ell = \|\mathbf{x}\|$ 的非线性特性。

若 Hessian 矩阵 $\nabla^2\Phi$ 在求解线性方程组 $\nabla^2\Phi \Delta\mathbf{x} = -\nabla\Phi$ 时不正定, 部分线性求解器 (如共轭梯度法、Cholesky 分解) 可能失效或产生错误结果。

11.3 强制正定方案

一种简单有效的解决方案是: 当 $\ell < s$ (弹簧被压缩) 时, 直接舍弃 Hessian 中的末端项, 即:

$$\boxed{\nabla^2 E \approx k \left(1 - \frac{s}{\ell}\right) \mathbf{I}} \quad (\text{当 } \ell < s \text{ 时, 或更简单地取为单位矩阵的倍数}). \quad (119)$$

更实用的方案是:

$$\nabla^2 E \approx \begin{cases} k \cdot \mathbf{I} & (\text{简化方案}) \\ k \left[1 - \frac{s}{\ell}\right] \mathbf{I} + k \frac{s}{\ell} \hat{\mathbf{x}}\hat{\mathbf{x}}^T & (\text{当 } \ell \geq s \text{ 时}) \end{cases} \quad (120)$$

在 Unity 实验实现中, 采用了更简化的方案: 将 Hessian 近似为对角矩阵, 即每个顶点的更新独立进行:

$$\Delta\mathbf{x}_i = -\frac{g_i}{H_{ii}} \quad (121)$$

其中 g_i 为第 i 个顶点对应的梯度分量, H_{ii} 为 Hessian 的对角元 (取足够大的正值以保证正定性)。