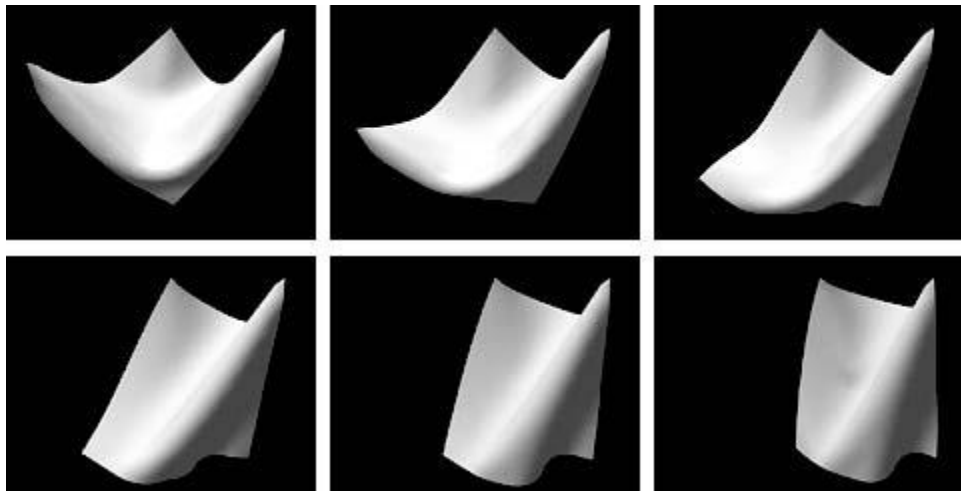


计算机图形学

作业3：物理模拟

在电影和游戏中, 物理仿真经常被用来制作各种各样物理过程和现象的动画, 包括爆炸、车祸、水、布等等。这种动画用关键帧技术来实现非常困难。但在给定支配运动的物理规律后, 再来模拟则相对容易。在本作业中, 将使用弹簧系统来构建一个视觉上吸引人的布料模拟, 如



图所示。

本文档组织如下：

1. [开始](#)
2. [需求概述](#)
3. [时间积分器](#)
4. [物理仿真](#)
5. [质点系统仿真布料](#)
6. [附加分](#)
7. [如何提交成果](#)

开始

查看提供示例解答演示程序,以了解您要完成的功能。

使用参数“e”、“t”或“r”运行a3soln文件(如: a3soln e),并观察“布料”掉落。您可以通过按r重置模拟,也可以通过按s来翻转布料。如果希望切换线框视图,请按s。你的任务是建立一个类似的模拟。

本目录中的其他文件提供了用OpenGL实现该应用的一个简单框架。请编译代码并运行它。切记,增量调试是好的做法!为了调试的目的,我们建议尝试让代码显示单个粒子,然后继续实现积分器。

本作业,你需要做设计。所提供的入门代码,仅给你完成作业提供基本引导,你可以随心所欲地更改代码。总的来说,需要花些时间思考写什么样的函数或类。

需求概述

同样,不必使用提供的所有初始代码(尽管我们推荐它)。本作业具有一定挑战性,需要你成为一名优秀的代码设计员和测试人员。为了确保部分附加作业,我们建议以下步骤。

首先,需要实现两种求解常微分方程的数值方法:Euler法和梯形法。需要使用课堂上讲过的简单一阶系统来测试。从系统中做出通用时间积分器是重要的。时间积分器应该足够通用,以便能对任何系统采取任何步骤者没有问题。此步与下一步独立,因为我们将为您提供一个合理的积分器(四阶龙格库塔RK4)。

接下来需要实现一个二阶系统,一个简单的摆,由两个带有弹簧的粒子组成。要你实现三种力:重力、粘性阻力和弹力。这些力在后面创建布模拟时都将用到。

其三,需要扩展简单单摆来创建一个具有四个粒子的弹簧串。这让你在开始装配布料之前增量测试弹簧的实现。

最后,用弹簧来组装一块布。它至少应由一个8*8的粒子网格构成。需要实现结构力,抗剪力和抗弯力的弹簧系统。

你的应用程序应该显示一个布的线框动画。实现类似于示例程序中的光滑渲染,将获得附加分。应用程序应该允许用户以某种方式移动布。它可以简单通过击键,使织物来回移动,就像示例程序中实现的一样。

需要实现一个名为a3的可执行文件,它可取两个参数。第一个是一个字符(e, t, r),选择求解器(Euler法,梯形法或RK4)。第二个是可选的求解器步长。你的应用程序应该实现对键“t”的捕获,以便在显示布料、简单系统和单摆之间切换。或者,如果你太懒,不提供此功能,但同时显示所有的。

3 时间积分器(TimeStepper.xpp)

理解时间积分器和粒子系统之间的抽象是很重要的。时间积分器对系统的物理性质一无所知。它可以要求粒子系统使用evalF方法来计算导数。这个函数是系统和积分器之间的关键通信信道。它以状态向量作为输入，并返回这个特定状态的导数矢量，这两个矢量都表示为阵列，而不管粒子系统的精确类型（只有阵列的大小变化）。这样使积分器通用且可重用。

粒子系统存储其当前状态X，但是时间积分器可能需要另一状态的导数，特别是对于梯形法。粒子系统使用正确的状态（调用evalF所要求的状态）来计算导数是至关重要的。确保你理解系统内部状态和积分器所要求的状态的区别。

再次强调，你的积分器应该是模块化的，并且能够匹配任何系统，而你的粒子系统应该非常小心地计算在所要求状态下的力，此状态与其当前状态可能不同。在欧拉积分器中这两个状态是相同，但在梯形法时则不同，不要被前者误导。

3.1 欧拉法和梯形法的更新器

最简单的积分器是显式Euler法。对于Euler步，给定状态X，我们求X处的检查 $f(X, t)$ ，然后得到新状态值。这需要选择一个步长h，并且我们基于 $f(X, t)$ 做如下处理

$$X(t + h) = X + hf(X, t)$$

此方法，尽管实现容易，但除了对最简单的粒子系统外，对其他系统可能不稳定。其后果是，必须使用小的步长(h)来得到合理的结果。

还有很多其他方法可以提供高精度和稳定性。对这些问题集，我们将使用梯形法，其做法使用当前状态的力 f_0 和用欧拉法虚走一步h得到的力 f_1 的均值：

$$\begin{aligned} f_0 &= f(X, t) \\ f_1 &= f(X + hf_0, t + h) \\ X(t + h) &= X + (f_0 + f_1) h/2 \end{aligned}$$

能够同时求当前状态X和状态 $X+hf_0$ 处的力，这对粒子系统是至关重要的。粒子状态仅在最后的方程中被更新。

3.2 简单例子 20% (simpleSystem.cpp)

需要实现欧拉法和梯形法。使用粒子系统的evalF方法计算“适当”状态的导数。它应该将系统的状态设置为更新的值。可参考抽象类ParticleSystem, 它提供了获取和设置系统状态的方法。

需要对你实现用类似于课堂上讲的简单一阶ODE来进行测试。此系统仅有单个粒子，其状态由x-y-z坐标给定：

$$X_t = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

其右边是

$$f(X, t) = \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$$

这只是一阶系统，而ODE的右边不描述物理力。你不必用我们在课堂上使用的技巧把牛顿系统变成一阶系统。在粒子系统和积分器之间传递的向量长度为3。z坐标不做任何有趣的事情，但我们保留它，使此问题集中的各问题更加一致，并使用Vector3f。

实现抽象类ParticleSystem的子类SimpleSystem中的evalF。正如以上反复强调的，你需要把系统传递给takeStep方法，该方法来处理系统状态的更新。

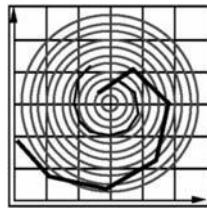
在本例中，仅有单个粒子，但下一个问题需要处理多个粒子，请确保你了解这些。我们通过将状态描述为一维Vector3f向量（查看课堂讲稿，看看应该如何表示系统的状态）给出了提示。给定系统状态X，evalF计算并返回f(X, t)（此处是速度）。evalF和积分器中的方法都不应该修改系统内的单个粒子。evalF以系统状态为参数并返回与该状态相关的导数。积分器方法应该在每个步骤自动化地修改系统的状态。

实现简单系统和欧拉积分器。尝试不同的h值，并观察精度变化。

正如在讲义中看到的，Euler法不稳定。精确解是圆，其方程为：

$$X(t) = \begin{pmatrix} r \cos(t + k) \\ r \sin(t + k) \end{pmatrix}$$

但是欧拉法使得解螺旋向外，无论h多小。在实现欧拉法后，你会看到二维空间的单个粒子螺旋向外，如下图所示。



接下来，实现梯形法。仔细思考接下来如何实现梯形法。它需要你计算在其他时间步及点位的导数f(X, t)。因此，你需要编写函数来求给定系统任意状态下的所有微分。记住，你的积分函数需要与系统分离并抽象出来。后面，你会对不同的系统使用这些积分函数，因此他们的模块化非常重要。

梯形法还是不稳定，但其以更慢的速度发散。你能比较你的欧拉法和梯形法的实现，观察粒子以不同的速率向外发散。检查你的欧拉法和梯形法如你所期望的一样工作，其中欧拉螺旋向外并发散，而梯形法也一样，只是会慢很多。你的应用程序的命令行应该让用户可以选择求解器和步长。每种算法的实现可得10%。

4 物理仿真

在本节中，需要实现一个简单的双粒子摆，并将其扩展到多个粒子链。这将需要你实现不

同类型的力（重力、粘性阻力和弹簧力）。已经为您提供了一个四阶Runge Kutta（RK4）积分器，因为你之前实现的积分器是不稳定的。

4.1 力

粒子系统模拟的核心是力。在给定粒子的位置 x_i ，速度 x_i' 和质量 m_i 。则重力可如下表示：

$$F(x_i, x_i', m_i) = m_i g$$

而粘性阻尼力(给定阻尼系数k)：

$$F(x_i, x_i', m_i) = -k x_i'$$

同样可以将力表达为涉及其他粒子的表达式。例如，如果用无粘阻的弹簧连接二个粒子i和j，弹簧的静态长度r弹性系数k, 将产生如下力：

$$F(x_i, x_i', m_i) = -k(\|d\| - r) \frac{d}{\|d\|}, \text{其中 } d = x_i - x_j$$

将所有力相加得到合力，用质量除合力得到加速度 x_i''

所有粒子的运动可以用二阶常微分方程来描述：

$$x'' = F(x, x')$$

在此表达式中，x描述所有粒子的位置（x有3n个元素，其中n是粒子数）。函数F对所有力求和并除以粒子的质量。

数值法求解此方程的典型方法是转换为一阶常微分方程。通过引入变换 $v=x'$ 来实现。将得到：

$$\begin{bmatrix} x' \\ v' \end{bmatrix} = \begin{bmatrix} v \\ F(x, v) \end{bmatrix}$$

总之，可以定义状态为系统中所有粒子的位置和速度

$$X = \begin{pmatrix} x \\ v \end{pmatrix}$$

由此得到：

$$\frac{d}{dt} X = f(X, t) = \begin{pmatrix} v \\ F(x, v) \end{pmatrix}$$

给定这些系统特性，你应该能用你的时间积分器来近似此系统。在下一节，你要实现一个简单的单摆和多粒子链来测试你的实现。注意，你不应该修改你的欧拉和梯形代码。你的积分器代码应该模块化并足够抽象，以便能处理源于任何系统的任何状态！当然，你应该仔细思考如何

存储状态。一个简单的选择是存储一个大小为 $2n$ 的大Vector3f数组，其中位置存储在偶数下标中，而速度存储在奇数下标中。编写一个助手函数可能对读粒子 i 的位置和速度有帮助。

4.2 简单单摆20%

下面要实现重力，粘性阻尼和弹簧力。首先用单个粒子连接到固定点的弹簧（基本上就是一个单摆）来测试，代码在pendulumSystem.cpp中。evalF应返回 $f(x, t)$ ，需要计算作用在粒子上的重力、粘阻和弹力

建议认真思考弹簧的表示，因为需要跟踪每个粒子上的弹力。可以存储一个弹簧的链表，每个弹簧中有其所作用到的二个粒子，静态长度和弹性系数。另一种方法是，对每个粒子存储其所连接的粒子，弹性系数和静态长度。

(选做)为了辅助调试，实现一个函数让你看到哪个弹簧连接到指定的粒子。用户以命令行参数指定数 i ，以渲染连接到粒子 i 的弹簧。(当有更多粒子时，这将变得更有用)。

你应该确保粒子的运动是正确的。请注意，特别是使用欧拉法时，需要提供合理的阻尼，否则系统会爆炸。梯形法会更稳定，但是你仍然需要一点粘性阻力使运动稳定。到目前为止，如果所有的东西都是正确的，那么你可得到40%的分数。

4.3 多粒子链20%

下一步是将测试扩展到多个粒子。试着用弹簧连接四个粒子形成一条链，并固定其中一个端点（可以通过将施加到粒子上的合力调零来固定一个粒子）。确保你可以模拟这个弹簧链的运动。一般来说，粒子和弹簧越多，不稳定性越大，所以必须仔细选择参数（弹簧常数、阻力系数、台阶尺寸）以避免爆炸。如果目前的实现都正确的，可以得到60%的分数。

5 粒子系统布料 40%

上一节描述了如何模拟一组粒子受重力、粘性阻力和弹力作用。本节中，我们将描述这些力如何组合以产生合理（但不一定精确）的布料模型。

在继续作业之前，推荐大家做一个代码的快照，以防止完整的布料实现无法工作。推荐使用Git进行版本控制。尽管使用版本控制系统会花费一点时间，但，总的说来是值得的。

你需要扩展ParticleSystem或者pendulumSystem，并对布料模拟实现自己的ClothSystem类。

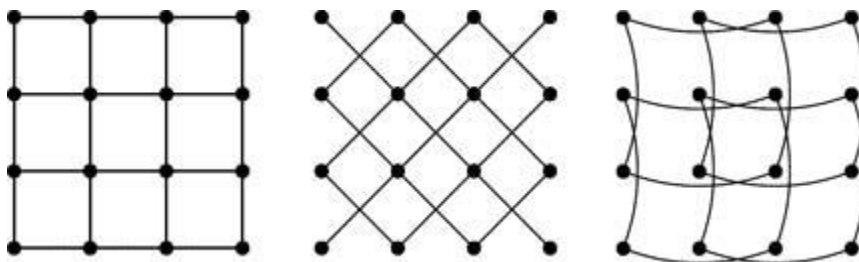


图1: 从左到右: 结构弹簧, 抗剪弹簧, 抗弯弹簧

我们从质点规则网格开始, 这些质点仅水平和铅直方向用弹簧连接其邻居。这些弹簧的作用是保持粒子网格在一起, 称其为结构弹簧。然后, 我们加入抗剪弹簧以防止布料沿对角线崩塌。最后, 我们加入抗弯弹簧以防止布料自折叠。注意抗弯弹簧画成曲线是为了清晰, 使其跨过一个粒子, 他们本身是“直”弹簧, 其力的方程也与前面给的一致。

如果所设计的代码足够好, 添加必须的弹簧不会很困难。请确保使用合理的静态长度, 最开始可以小一点。此处请特别小心地编写代码, 很容易产生错误并且连接弹簧到并不存在的粒子。推荐你创建一个辅助函数`indexOf`, 该函数输入为 $n \times n$ 布料的下标 i, j , 输出为粒子向量的线性下标。

首先, 实现结构弹簧。画出弹簧以确保你添加了正确的进去。确保看起来与你期望的一样, 然后往下做。运行仿真软件, 你将得到某个看起来象网的东西。与之前一样, 粘性阻尼可帮你防止爆炸。为了能够快速调试, 可使用较小的网格, 如 3×3 个粒子。

在确保结构弹簧正确后, 添加抗前弹簧。同样进行增量测试来防止错误, 最后添加抗弯弹簧。

显示布料的最简单方式是画出格网或者结构弹簧。(这个在调试结构弹簧实现时你已经完成了!)。为了得到附加分, 你可以像示例程序中一样画出光滑表面。如果你真的要用光滑表面来画布料, 你需要求出每个点的法向量。

如果你的初次实现看起来很可怕, 或者由于不稳定而飞起来了, 不要灰心。此时, 你的欧拉求解器除了对最小的步长之外, 没有任何效果, 几乎没有例外的, 你应该使用梯形求解器。

如果你完成了移动框架布料, 那么你完成了90%。余下的工作只是添加必要的用户接口, 如渲染布料, 移动等等。这样就完成100%了。

你会发现David Baraff写的基于物理建模的[这些笔记](#)会有些帮助, 特别是粒子系统动力学。

6 附加分

下面列出的附加分是一组可能的内容。总的来说, 可视化模拟技术从很多工程学科中获益, 并且利益于广泛的数值分析技术。请随意试验这里没有列出的想法和思路。同样请确保在加入附加特性后你的代码对前面的内容仍然可用。

6.1 易

- 添加随机风力到布料模拟中模拟微风。使用某键来切换有风和无风。
- 除了用线框网格来显示布料, 实现光滑渲染。此部分最具挑战是在每个结点定义法向, 这

需要使用邻近粒子的位置来进行近似。

- 用同样技术实现不同物体。例如，通过扩展粒子网格为三维网格，可做出摇晃明胶。
- 给用户供基于鼠标的接口与布料交互。例如，您可以允许用户点击布的某部分并拖动。
- 用简单的图元（如球体）实现布的无摩擦碰撞。这比起初听起来要简单：只是检查粒子是否“在”球体内；如果是，就把点投影回表面。
- 实现4阶Runge-Kutta积分器，这和我们提供的是一样的。

6.2 中等

- 实现自适应求解器方案(参考自适应Runge-Kutta-Fehlberg技术或参考MATLAB ode45函数)。
- 实现[此文中](#)描述的隐式积分方法。此技术能对微分方程系统刚度提供极强的稳定性，如在布料模拟中引起的问题。例如，隐式欧拉积分技术，和你用显式方法实现的一样不精确。但是，这里的不精确趋向于将解往稳定方向偏，因此允许更大的步长。示例解演示了些技术，可以在命令行输入i来启用。
- 扩展粒子系统以支持约束，如此[文档](#)所描述的。

6.3 难

- 实现更可靠的布料模型，如[此文](#)中描述的
- 模拟[刚体动力学](#)，[可变形体模型](#)，或[流体](#)。理论上，粒子系统可以用来得到类似效果。但是，通过更复杂的物理和数学模型，可以获得更高精度和效率。

7 如何提交成果

需要编写README.txt文件来回答下列问题：

- 如何编译和运行代码？如果你的可执行文件需要某些参数才能正常工作，请确保这些参数被指定。
- 和班上的谁合作过吗？如果是，请说明和谁交谈，以及你给予或接受了什么样的帮助。
- 有什么参考文献（书，文章，网站等）你觉得对你完成作业特别有帮助，请列出来。
- 你的代码有些什么已知问题？这很重要，如果你帮助我们理解发生了些什么，会给你加分
- 实现了什么附加分吗？如果是，告诉我们如何使用你设计的附加特性。如果涉及较多工作，请描述你是如何做的。
- 对本作业有些什么可分享的建议和意见？
- 和前面作业一样，你需要创建单个压缩文件（.tar.gz or .zip），其中包括：
 - 所有编译你的作业要用到原代码。
 - 编译好的可执行文件a3.
 - README.txt 文件