

这节随便听听，再做一下作业 8

Final Project 开个开源库。

嗯哼，动画。

## Mass Spring System 质点弹簧系统

Idealized spring




Diagram showing two points,  $a$  and  $b$ , connected by a spring. The force exerted by the spring on point  $b$  is given by:

$$f_{a \rightarrow b} = k_s(b - a)$$
$$f_{b \rightarrow a} = -f_{a \rightarrow b}$$

Force pulls points together

Strength proportional to displacement (Hooke's Law)

$k_s$  is a spring coefficient: stiffness

改变  $k_s$ ，就是对不同材质的模拟，一种材质内部可能有不同的  $k_s$

Damp only the internal, spring-driven motion

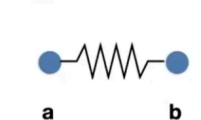


Diagram showing two points,  $a$  and  $b$ , connected by a spring. The damping force applied on point  $b$  is given by:

$$f_b = -k_d \frac{b - a}{\|b - a\|} (\dot{b} - \dot{a}) \cdot \frac{b - a}{\|b - a\|}$$

Annotations: "Relative velocity of b, assuming a is static (vector)" points to  $\dot{b} - \dot{a}$ . "Damping force applied on b" points to  $f_b$ . "Relative velocity projected to the direction from a to b (scalar)" points to the dot product term. "Direction from a to b" points to  $\frac{b - a}{\|b - a\|}$ .

- Viscous drag only on change in spring length
  - Won't slow group motion for the spring system (e.g. global translation or rotation of the group)
- Note: This is only one specific type of damping

图 1 还要考虑摩擦力，否则系统不会停止运动

## Particle System 粒子系统

说了挺多哈哈，懒得记了。

### 逆运动学

难题

1. 难解
2. 解不唯一

## Euler's Method

Euler's Method (a.k.a. Forward Euler, Explicit Euler)

- Simple iterative method
- Commonly used
- Very inaccurate
- Most often goes **unstable**

$$\begin{aligned}x^{t+\Delta t} &= x^t + \Delta t \dot{x}^t \\ \dot{x}^{t+\Delta t} &= \dot{x}^t + \Delta t \ddot{x}^t\end{aligned}$$

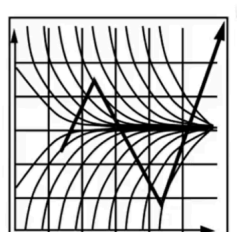
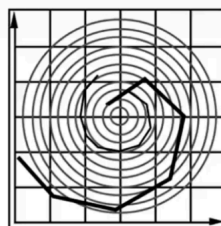
## Instability of the Euler Method

The Euler method (explicit / forward)

$$x^{t+\Delta t} = x^t + \Delta t v(x, t)$$

Two key problems:

- Inaccuracies increase as time step  $\Delta t$  increases
- Instability is a common, serious problem that can cause simulation to diverge



Witkin and Baraff

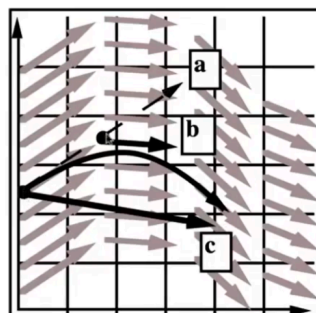
问题：除了采样率之外，误差会被无限放大才是更致命的。

## Midpoint Method

Midpoint method

- Compute Euler step (a)
- Compute derivative at midpoint of Euler step (b)
- Update position using midpoint derivative (c)

$$\begin{aligned}x_{\text{mid}} &= x(t) + \Delta t/2 \cdot v(x(t), t) \\ x(t + \Delta t) &= x(t) + \Delta t \cdot v(x_{\text{mid}}, t)\end{aligned}$$



Witkin and Baraff

## Modified Euler

### Modified Euler

- Average velocity at start and end of step
- Better results

$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \frac{\Delta t}{2} (\dot{\mathbf{x}}^t + \dot{\mathbf{x}}^{t+\Delta t})$$

$$\dot{\mathbf{x}}^{t+\Delta t} = \dot{\mathbf{x}}^t + \Delta t \ddot{\mathbf{x}}^t$$

$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \Delta t \dot{\mathbf{x}}^t + \frac{(\Delta t)^2}{2} \ddot{\mathbf{x}}^t$$



## Adaptive Step Size

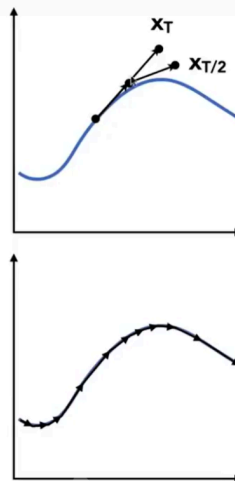
既然反正都要算这个.....

### Adaptive step size

- Technique for choosing step size based on error estimate
- Very practical technique
- But may need very small steps!

Repeat until error is below threshold:

- Compute  $\mathbf{x}_T$  an Euler step, size  $T$
- Compute  $\mathbf{x}_{T/2}$  two Euler steps, size  $T/2$
- Compute error  $\|\mathbf{x}_T - \mathbf{x}_{T/2}\|$
- If (error > threshold) reduce step size and try again



### How to determine / quantize "stability"?

- We use the local truncation error (every step) / total accumulated error (overall)
- Absolute values do not matter, but the orders w.r.t. step
- Implicit Euler has order 1, which means that
  - Local truncation error:  $O(h^2)$  and
  - Global truncation error:  $O(h)$  (h is the step, i.e.  $\Delta t$ )

图 2 梦回渐进理论

# Runge-Kutta

## Runge-Kutta Families

被遗忘的角落

A family of advanced methods for solving ODEs

- Especially good at dealing with non-linearity
- It's order-four version is the most widely used, a.k.a. RK4

Initial condition:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

RK4 solution:

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4),$$
$$t_{n+1} = t_n + h$$

where

$$k_1 = f(t_n, y_n),$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right),$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right),$$

$$k_4 = f(t_n + h, y_n + hk_3).$$

其实就是高级一点的中点法

---

完结撒花。

做一下作业 8

嗯，环境有问题，那就不做了。

后续的 final project，也是我带专生涯的毕业设计，会继续更新，放在 [https://gitee.com/isirin1131\\_admin/specialized-graduation-program](https://gitee.com/isirin1131_admin/specialized-graduation-program)

告一段落喽。

有什么感言呢？我只是感觉有些无聊了，完成我的生成函数手稿后，就会开始 games001 的学习。

之后就是 games 200 系列了？或许会开始搞 AI 了，但那太远了，还是关注在短期目标上吧。