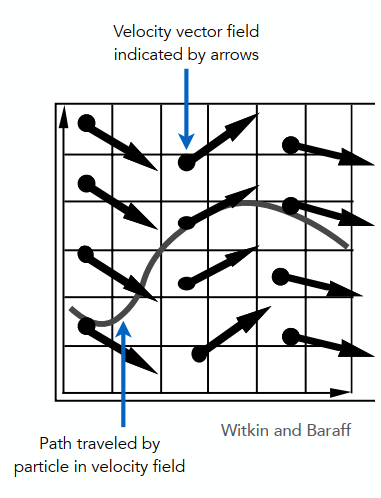
# Single particle simulation

首先研究单个粒子的运动，稍后推广到大量粒子

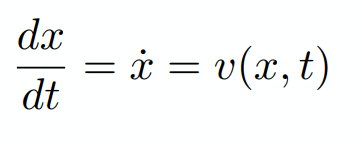
假设粒子的运动由作为位置和时间函数的速度矢量场确定：v(x, t)

假设把粒子放在某个速度场内，那么粒子应该会随着速度场而运动，那么就要模拟这个粒子速度随时间变化的过程。



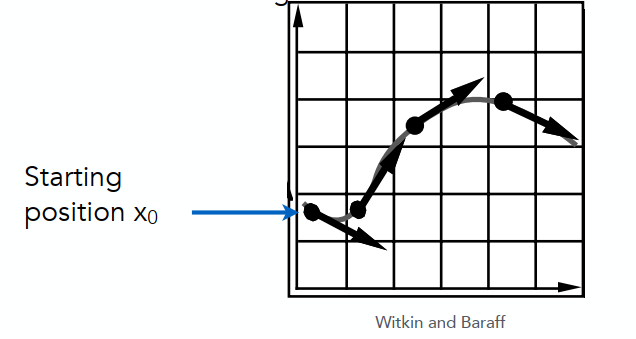
# Ordinary Differential Equation (ODE)

计算粒子随时间的位置需要求解一阶常微分方程：



“一阶”是指采用的一阶导数。

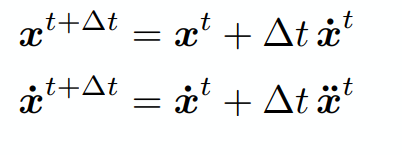
“普通”意味着没有“偏”导数，即 x 只是 t 的函数。



我们可以通过使用前向数值积分求解 ODE，服从给定的初始粒子位置 x0。这里用常微分方程去求速度。

## Euler’s Method (a.k.a. Forward Euler, Explicit Euler)

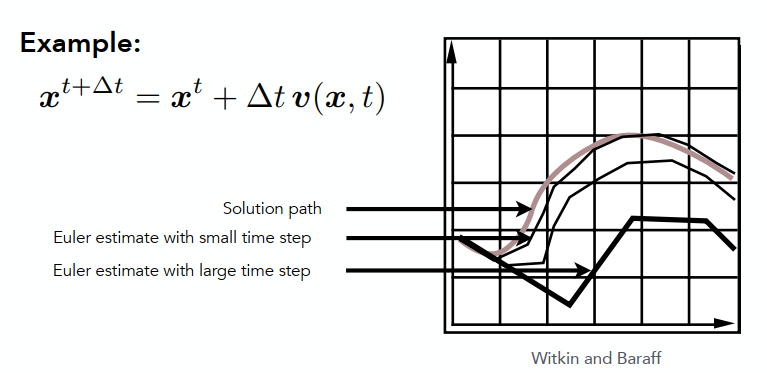
欧拉方法：最简单的一种方法就是把时间分成小段，每一个小块计算时间△t。知道某个时刻t的位置和速度，计算下一时刻t+△t的位置和速度。



## Euler’s Method - Errors

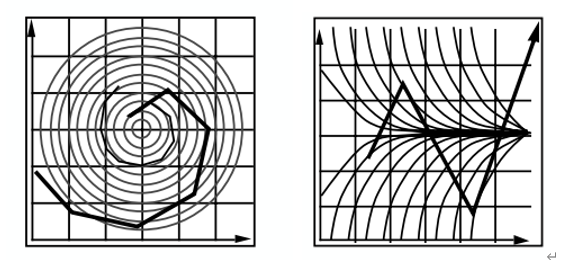
随着数值积分误差累积，欧拉积分特别糟糕

例子：



如图，用不同大小的步长△t，得到的结果会不同，△t分的越小，得到的结果就会越准。

## Instability of the Euler Method



对于这样一个环型速度场，无论取多小的步长△t，得到的轨迹都不可能沿着一个环走，因为速度的方向一直在改变，每一步总会有偏差，那么这个偏差是不断累积的，就会偏的越来越多最后飞出去。

## Euler Method two key problems:

不准确度随着时间步长 Δt 的增加而增加

不稳定性是一个常见的、严重的问题，会导致模拟发散

因此欧拉方法的两个问题就是：误差和不稳定

### Errors and Instability

通过有限差分的数值积分求解会导致两个问题：

误差：每个时间步的错误都会累积。随着仿真的进行，精度会降低。精度在图形应用程序中可能并不重要。

不稳定：误差可能会叠加，导致仿真发散。在仿真中缺乏稳定性是一个基本问题，不容忽视。

# Combating Instability

解决不稳定性的几种常用办法：

中点法/修正欧拉：起点和终点的平均速度

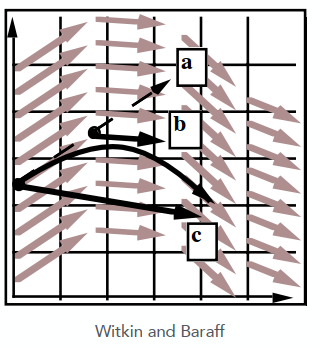
自适应步长：递归地比较一步和两个半步，直到误差可以接受

隐式方法：在下一个时间步使用速度（硬）

基于位置/Verlet 集成：在时间步后约束粒子的位置和速度

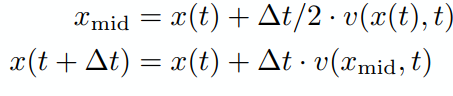
## 1、Midpoint Method

中点法

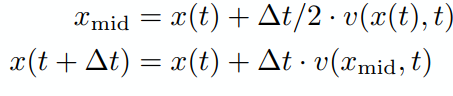


限定时间为t

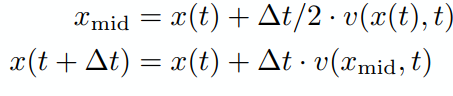
运用欧拉方法，粒子会从初始位置出发到达a点。但是a点的速度场并不稳定，粒子继续从a点运动很有可能发散逃出速度场。为此计算t/2时刻粒子b的位置Xmid，



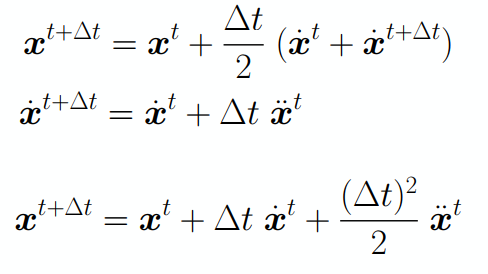
并由速度场及时间t可计算b的速度。



用中点的速度代替初始位置的速度去计算终点c的位置。c点比a点更为稳定。



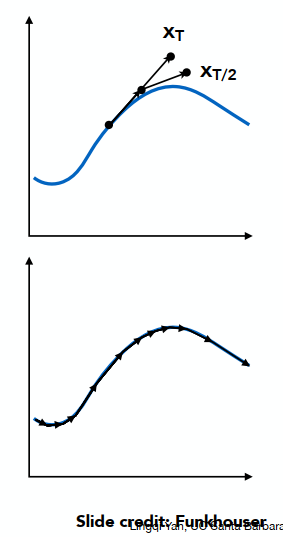
以下这种中点法(用原始位置速度和中点位置速度的均值替代初始位置的速度)会获得更好的结果：



将式子整理写开可以发现，中点法比欧拉方法多了一个二次的项，相当于在模拟抛物线，因此更准确

## 2、Adaptive Step Size

自适应方法核心在于自适应步长，是一种基于误差估计选择步长的技术。这种方法依赖于非常小的步长，需要不断重复直到误差低于阈值。

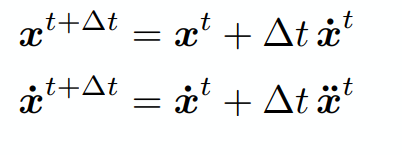


原始点用欧拉方法计算经过△t后运动到了xT点，显然不是很准。这时候把时间减半变成△t/2，原始点用欧拉方法计算经过△t/2后运动到了中点，中点再用欧拉方法计算经过△t/2后运动到了xT/2。然后比较xT和xT/2的位置，如果差的很远，那么就应该把时间再分的更小去计算。直到两个点的位置差不多，说明时间已经分的足够小了，这时候得到的结果是比较准确的。

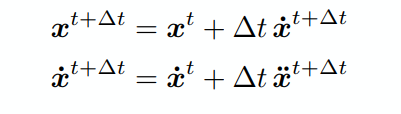
## 3、Implicit Euler Method

隐式欧拉方法：(后向方法)用未来推现在。

首先回忆下显示欧拉方法：下一个时间步的位置依赖于当前位置及速度，下一个时间步的速度依赖于当前的速度及加速度。



隐式欧拉方法最大的区别在于：下一个时间步的位置依赖于当前位置及下一时间步的速度，下一个时间步的速度依赖于当前的速度及下一时间步的加速度。



用未知推已知的方法要依赖于微分方程组，使用寻根算法，例如牛顿法进行迭代求解。

虽然隐式欧拉方法难解，但是可以提供很好的稳定性。

#### How to determine / quantize “stability”?

如何定义这个方法是不是稳定的以及多么稳定？

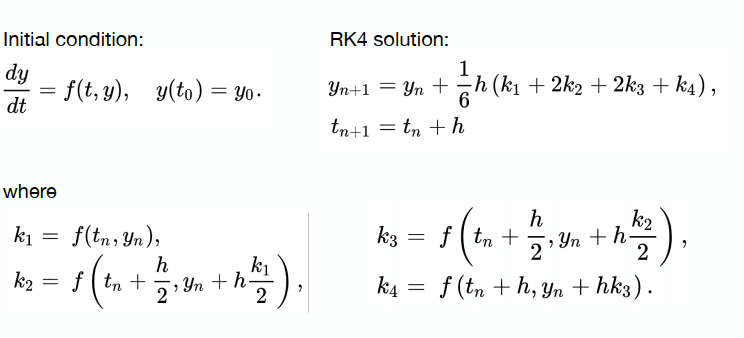
通常对数值方法会定义两个概念：局部截断误差（每一步会产生的误差）和整体误差（每一步的局部截断误差的累积）(死去的数值分析开始攻击我)。研究这两个数没有太大意义，但是研究他们的阶（误差与△t的关系）很有意义。隐式欧拉方法是一阶的，也就是局部误差是O(h2)，整体误差是O(h)。这里h指的是取的时间步长△t。

如何理解O(h)？

如果把h（△t）减小一半，那么期望误差也会跟着减小1/2。对于 O(h2)来讲，如果把h（△t）减小一半，那么误差会减小到1/4。如果有O(h3)，那么如果把h（△t）减小一半，那么误差会减小到1/8。因此阶数越高误差越小。

## 4、Runge-Kutta Families

龙格-库塔方法：龙格库塔方法特别适合解微分方程，尤其是对于非线性的。它的四阶版本是使用最广泛的，又名 RK4。



## 5、Position-Based / Verlet Integration

主要思想：

修改欧拉前步法，约束粒子的位置以防止发散、不稳定行为

使用约束位置来计算速度

这两种想法都会消耗能量，稳定性。

优点：

快速简单

不基于物理，耗散能量（误差）

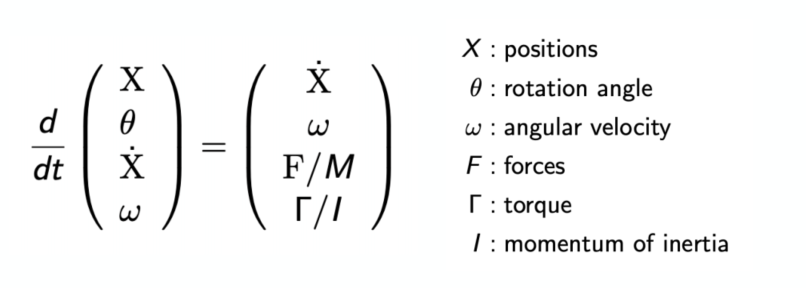
缺点

不基于物理的方法，通过调整不同位置使得满足某种限制。但是由于不是基于物理的方法，因此不满足能量守恒。

## 6、Rigid Body Simulation

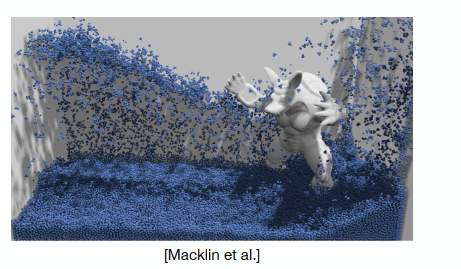
刚体模拟案例：

对于刚体的模拟类似于对单个粒子的模拟，但是会更多的考虑刚体本身的物理量。



# Fluid Simulation

流体模拟：一种简单的基于位置的方法



关键思想：

整个水体是由很多不可压缩的刚体小球组成。通过模拟这些小球的运动位置，来模拟整个水体的运动，最后再渲染出来。

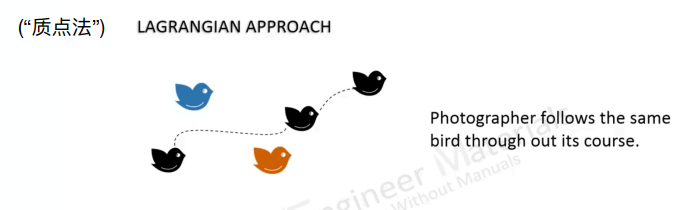
这里假设整个水是不可压缩的。也就是在任何一个时刻任何一个地方的密度相同。

通过任何一个时刻小球的分布都可以知道某个小球周围的密度。如果有任何一个地方的密度和之前平静的水在这个位置的密度不一样，那么就需要通过移动小球把这个地方的密度修正回来。通过不断的根据小球分布计算不同位置的密度，不断的修正，就可以将水模拟出来。这种方式是不基于物理的。

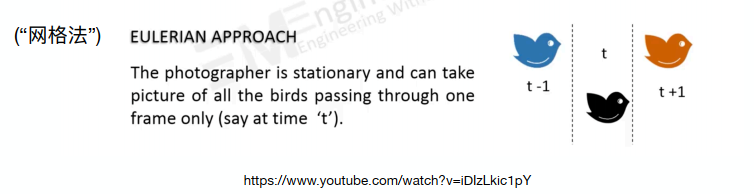
## Eulerian vs. Lagrangian

在物理中模拟大规模物质的两种方法：

质点法（拉格朗日方法）：如刚才对水体的模拟，认为水是由各个圆形的小水滴组成的，通过模拟各个小水滴的运动来模拟整个的水体运动。



网格法（欧拉方法）：把空间划分成网格，计算不同网格中不同时间密度的变化来模拟。



## Material Point Method (MPM)



结合拉格朗日方法和欧拉法：首先认为不同的粒子具有某些材质属性（如上图这是一只会融化的兔子，粒子间有粘性、粒子有质量等），然后融化过程通过网格去计算，算出来之后再把信息写回每个粒子中去。