

CS-184: Computer Graphics

Lecture #21: Physically Based Animation Intro

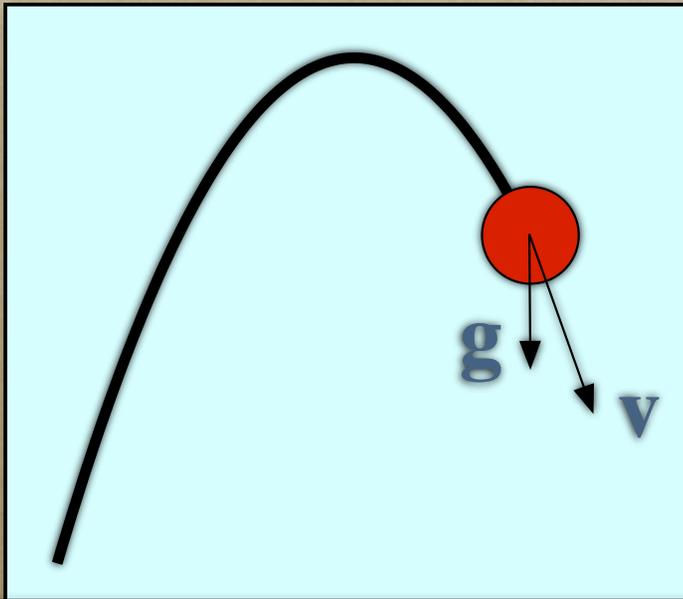
Prof. James O'Brien
University of California, Berkeley

Today

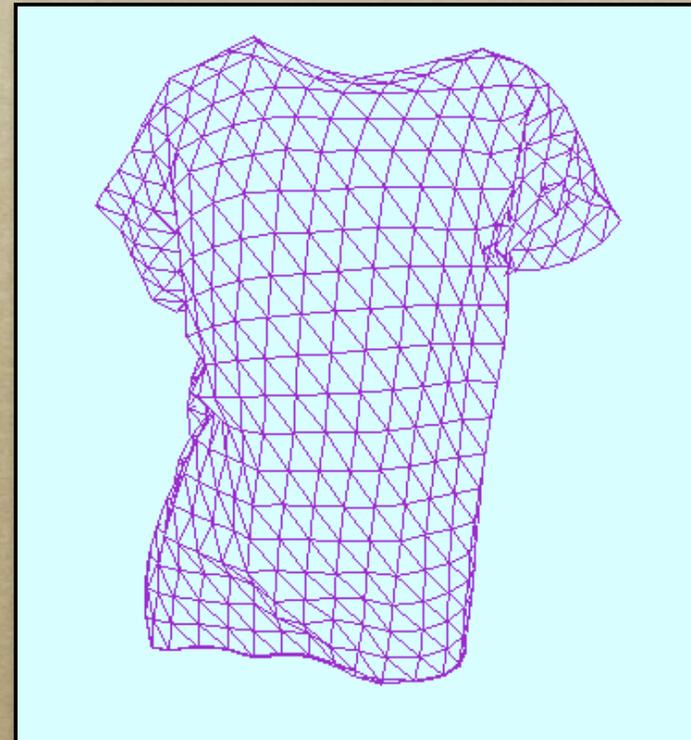
- Introduction to Simulation
 - Basic particle systems
 - Time integration (simple version)

Physically Based Animation

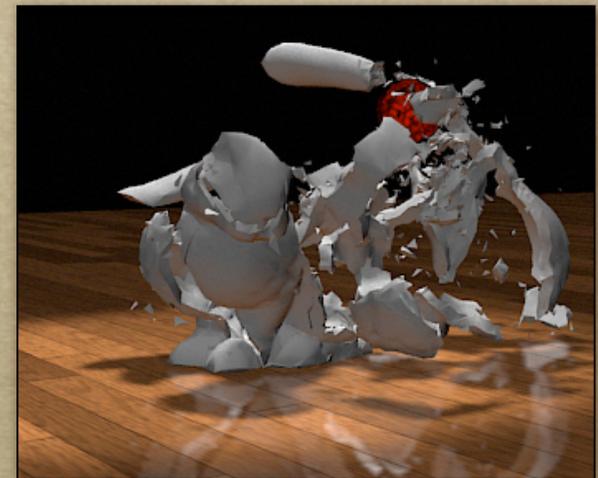
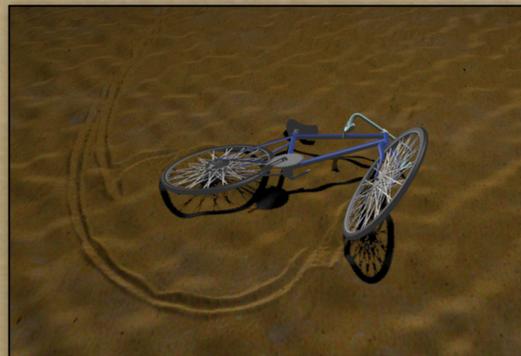
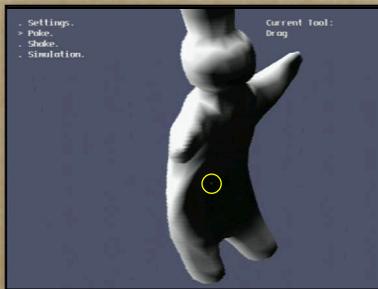
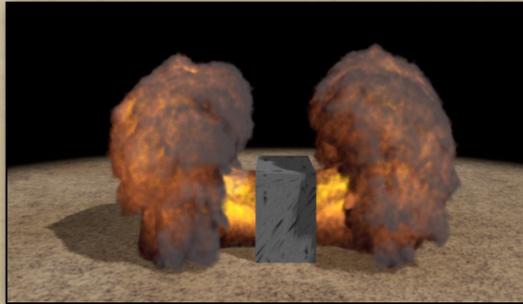
- Generate motion of objects using numerical simulation methods



$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \Delta t \mathbf{v}^t + \frac{1}{2} \Delta t^2 \mathbf{a}^t$$



Physically Based Animation



Particle Systems

- Single particles are very simple
- Large groups can produce interesting effects
- Supplement basic ballistic rules
 - Collisions
 - Interactions
 - Force fields
 - Springs
 - Others...



Karl Sims, SIGGRAPH 1990

Particle Systems

- Single particles are very simple
- Large groups can produce interesting effects
- Supplement basic ballistic rules
 - Collisions
 - Interactions
 - Force fields
 - Springs
 - Others...



Feldman, Klingner, O'Brien, SIGGRAPH 2005

Basic Particles

- Basic governing equation $\ddot{x} = \frac{1}{m}f$
 - f is a sum of a number of things
 - Gravity: constant downward force proportional to mass
 - Simple drag: force proportional to negative velocity
 - Particle interactions: particles mutually attract and/or repel
 - Beware $O(n^2)$ complexity!
 - Force fields
 - Wind forces
 - User interaction

Basic Particles

- Properties other than position
 - Color
 - Temp
 - Age
- Differential equations also needed to govern these properties
- Collisions and other constrains directly modify position and/or velocity

Integration

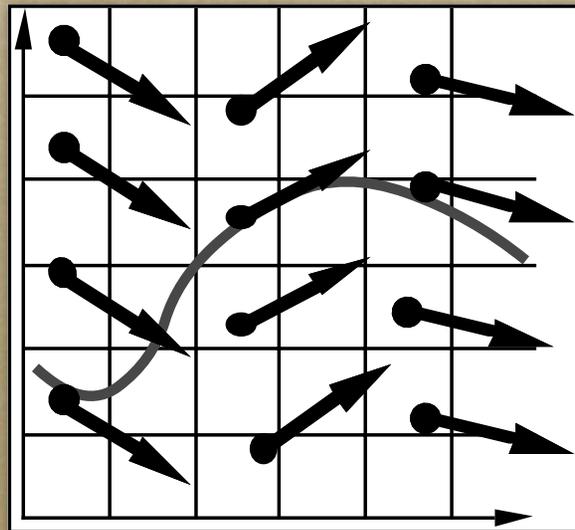
- Euler's Method
 - Simple
 - Commonly used
 - Very inaccurate
 - Most often goes unstable

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

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

Integration

- For now let's pretend $f = mv$
 - *Velocity* (rather than acceleration) is a function of force



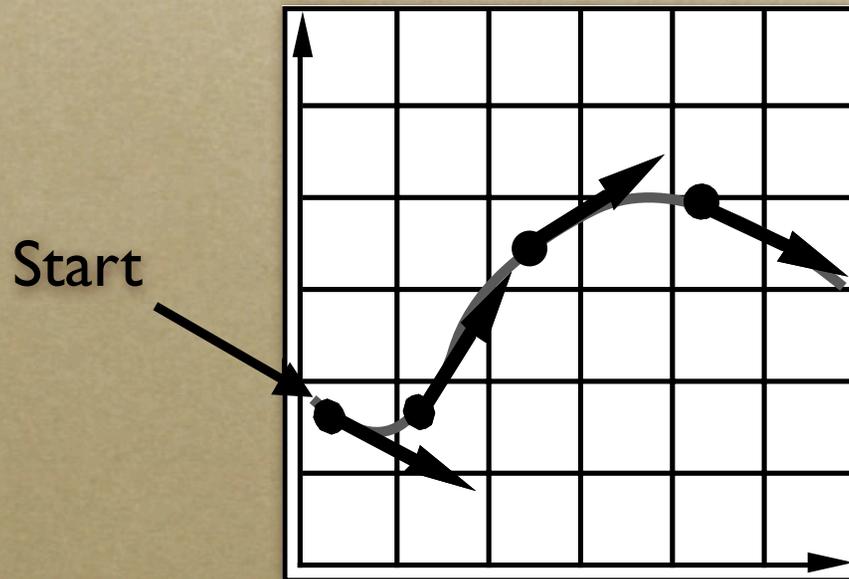
Witkin and Baraff

$$\dot{x} = f(x, t)$$

Note: Second order ODEs can be turned into first order ODEs using extra variables.

Integration

- For now let's pretend $f = mv$
 - *Velocity* (rather than acceleration) is a function of force

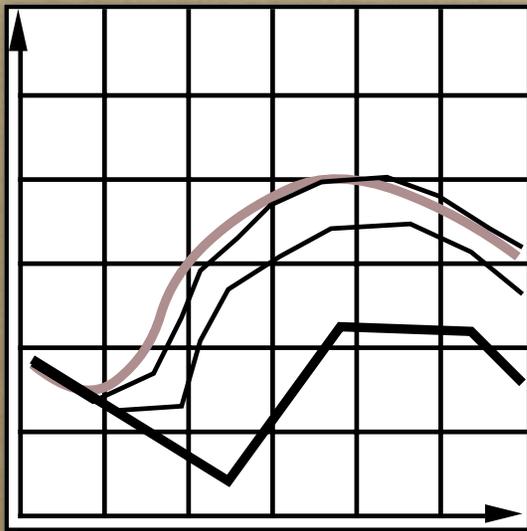


Witkin and Baraff

$$\dot{x} = f(x, t)$$

Integration

- With numerical integration, errors accumulate
- Euler integration is particularly bad

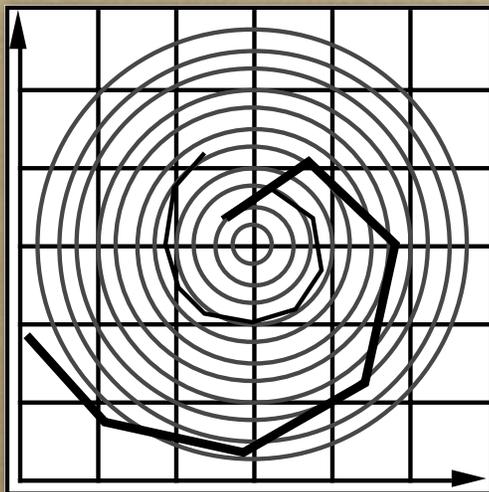


Witkin and Baraff

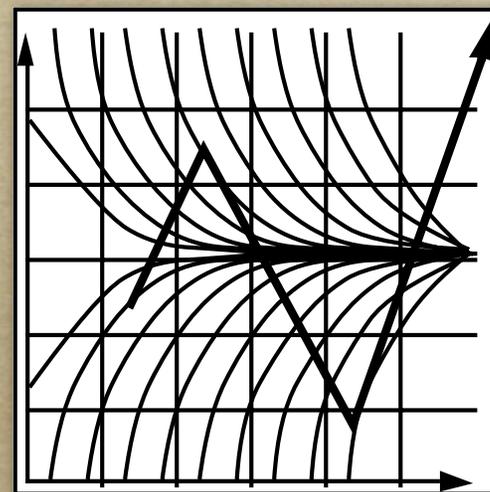
$$\mathbf{x} := \mathbf{x} + \Delta t \mathbf{f}(\mathbf{x}, t)$$

Integration

- Stability issues can also arise
 - Occurs when errors lead to larger errors
 - Often more serious than error issues



$$\dot{\mathbf{x}} = [-\sin(\omega t) , -\cos(\omega t)]$$



Witkin and Baraff

Integration

- Modified Euler

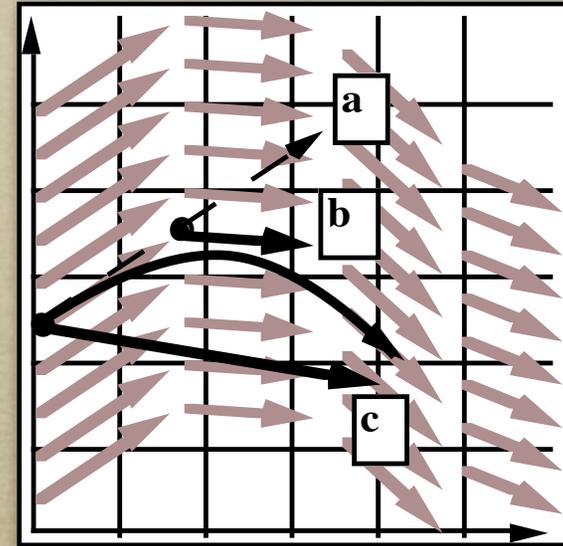
$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \frac{\Delta t}{2} (\dot{\mathbf{x}}^t + \dot{\mathbf{x}}^{t+\Delta t})$$

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

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

Integration

- Midpoint method
 - a. Compute half Euler step
 - b. Eval. derivative at halfway
 - c. Retake step
- Other methods
 - Verlet
 - Runge-Kutta
 - And *many* others...



Witkin and Baraff

Integration

- Implicit methods
 - Informally (incorrectly) called backward methods
 - Use derivatives in the future for the current step

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

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

$$\dot{\mathbf{x}}^{t+\Delta t} = \mathbf{V}(\mathbf{x}^{t+\Delta t}, \dot{\mathbf{x}}^{t+\Delta t}, t + \Delta t)$$

$$\ddot{\mathbf{x}}^{t+\Delta t} = \mathbf{A}(\mathbf{x}^{t+\Delta t}, \dot{\mathbf{x}}^{t+\Delta t}, t + \Delta t)$$

Integration

- Implicit methods

- Informally (incorrectly) called backward methods

- Use derivatives in the future for the current step

$$\dot{\mathbf{x}}^{t+\Delta t} = \dot{\mathbf{x}}^t + \Delta t \mathbf{V}(\mathbf{x}^{t+\Delta t}, \dot{\mathbf{x}}^{t+\Delta t}, t + \Delta t)$$

$$\dot{\mathbf{x}}^{t+\Delta t} = \dot{\mathbf{x}}^t + \Delta t \mathbf{A}(\mathbf{x}^{t+\Delta t}, \dot{\mathbf{x}}^{t+\Delta t}, t + \Delta t)$$

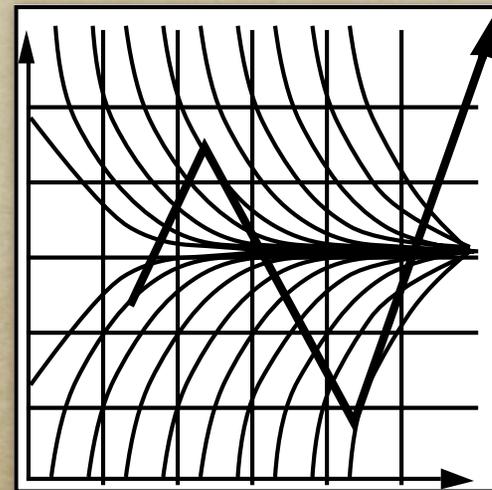
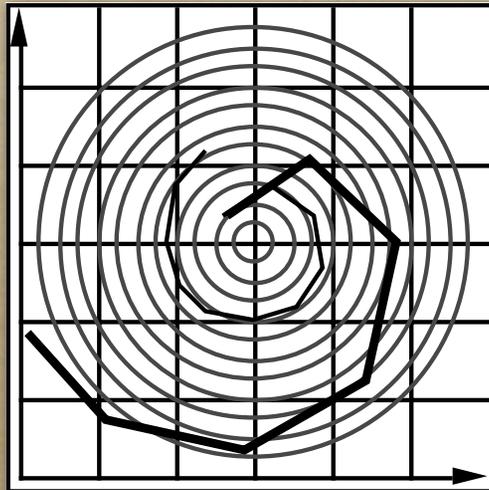
- Solve nonlinear problem for $\mathbf{x}^{t+\Delta t}$ and $\dot{\mathbf{x}}^{t+\Delta t}$

- This is fully implicit backward Euler

- Many other implicit methods exist...

- Modified Euler is *partially* implicit as is Verlet

Temp Slide



Need to draw reverse diagrams....

Integration

- Semi-Implicit

- Approximate with linearized equations

$$V(\mathbf{x}^{t+\Delta t}, \dot{\mathbf{x}}^{t+\Delta t}) \approx V(\mathbf{x}^t, \dot{\mathbf{x}}^t) + \mathbf{A} \cdot (\Delta \mathbf{x}) + \mathbf{B} \cdot (\Delta \dot{\mathbf{x}})$$

$$\mathbf{A}(\mathbf{x}^{t+\Delta t}, \dot{\mathbf{x}}^{t+\Delta t}) \approx \mathbf{A}(\mathbf{x}^t, \dot{\mathbf{x}}^t) + \mathbf{C} \cdot (\Delta \mathbf{x}) + \mathbf{D} \cdot (\Delta \dot{\mathbf{x}})$$

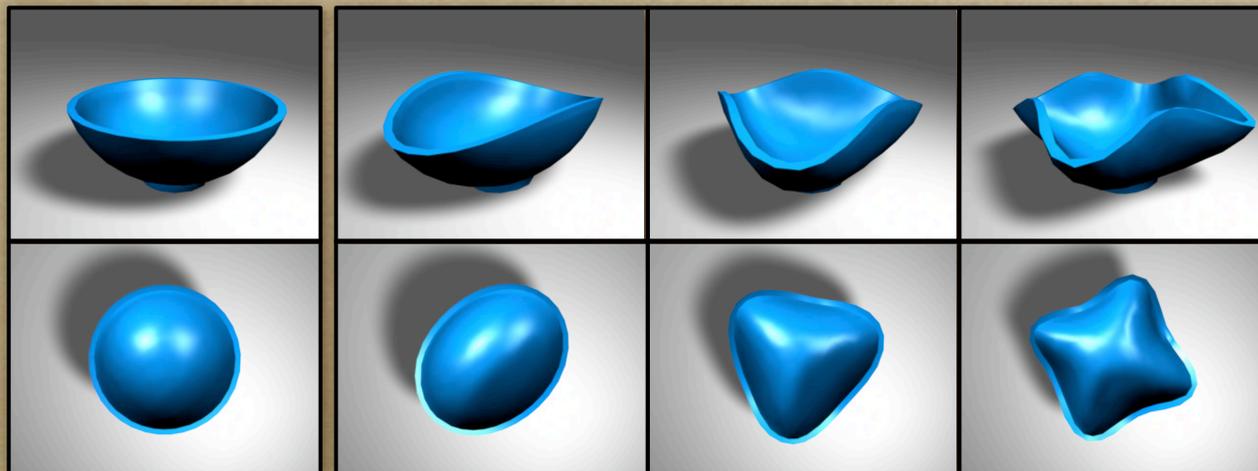
$$\begin{bmatrix} \mathbf{x}^{t+\Delta t} \\ \dot{\mathbf{x}}^{t+\Delta t} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^t \\ \dot{\mathbf{x}}^t \end{bmatrix} + \Delta t \left(\begin{bmatrix} \dot{\mathbf{x}}^t \\ \ddot{\mathbf{x}}^t \end{bmatrix} + \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \dot{\mathbf{x}} \end{bmatrix} \right)$$

Integration

- Explicit methods can be conditionally stable
 - Depends on time-step and *stiffness* of system
- Fully implicit can be **un**conditionally stable
 - May still have large errors
- Semi-implicit can be conditionally stable
 - Nonlinearities can cause instability
 - Generally more stable than explicit
 - Comparable errors as explicit
 - Often show up as excessive damping

Integration

- Integrators can be analyzed in modal domain
- System have different component behaviors
- Integrators impact components differently



Suggested Reading

- Physically Based Modeling: Principles and Practice
 - Andy Witkin and David Baraff
 - <http://www-2.cs.cmu.edu/~baraff/sigcourse/index.html>
- Numerical Recipes in C++
 - Chapter 16
- Any good text on integrating ODE's