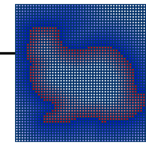
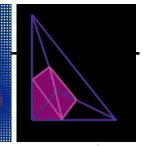
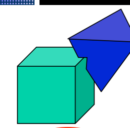
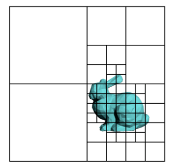
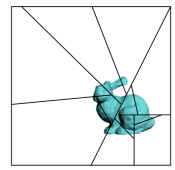
# Mass-Spring Systems

## Last Time?

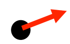
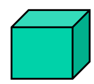
- Implicit Surfaces & Marching Cubes/Tetras
- Collision Detection & Conservative Bounding Regions
- Spatial Acceleration Data Structures
  - Octree, k-d tree, BSF tree

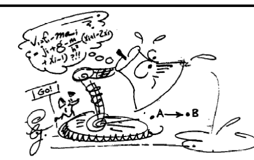







## Today

- Particle Systems
  - Equations of Motion (Physics)
  - Forces: Gravity, Spatial, Damping
  - Numerical Integration (Euler, Midpoint, etc.)
- Mass Spring System Examples
  - String, Hair, Cloth
- Stiffness
- Discretization

## Types of Dynamics

- Point 
- Rigid body 



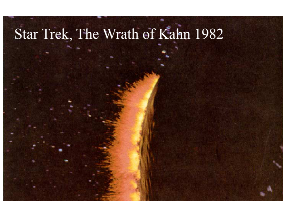
Witkin & Kass, "Spacetime Constraints", 1988.

- Deformable body (include clothes, fluids, smoke, etc.)

Carlson, Mucha, Van Horn, & Turk 2002

## What is a Particle System?

- Collection of many small simple particles that maintain *state* (position, velocity, color, etc.)
- Particle motion influenced by external *force fields*
- *Integrate* the laws of mechanics (ODE Solvers)
- To model: sand, dust, smoke, sparks, flame, water, etc.



Star Trek, The Wrath of Kahn 1982

Mark B. Allan
http://users.rcn.com/mba.dnai/software/flow/

## Particle Motion

- mass $m$, position $x$, velocity $v$
- equations of motion:

$$\frac{d}{dt}x(t) = v(t)$$

$$\frac{d}{dt}v(t) = \frac{1}{m}F(x,v,t) \qquad F = ma$$

- Analytic solutions can be found for some classes of differential equations, but most can't be solved analytically
- Instead, we will numerically approximate a solution to our *initial value problem*

## Higher Order ODEs

- Basic mechanics is a 2nd order ODE:

$$\frac{d^2}{dt^2} x = \frac{1}{m} F$$

- Express as 1st order ODE by defining $v(t)$:

$$\frac{d}{dt} x(t) = v(t)$$
$$\frac{d}{dt} v(t) = \frac{1}{m} F(x,v,t)$$

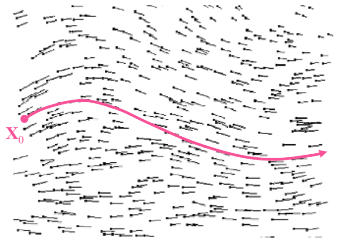
$$\mathbf{X} = \begin{pmatrix} x \\ v \end{pmatrix} \quad f(X,t) = \begin{pmatrix} v \\ \frac{1}{m} F(x,v,t) \end{pmatrix}$$

$X$ is a vector storing the *current state* of the particle

$f(X,t)$ describes how to *update* the state of the particle

## Path Through a Field

- $f(\mathbf{X},t)$ is a vector field defined everywhere
  - E.g. a velocity field which may change over time




Note: In the simplest particle systems, the particles do *not* interact with each other, only with external force fields

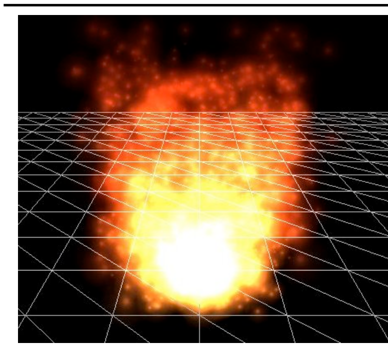
- $\mathbf{X}(t)$ is a path through the field

## For a Collection of 3D particles…

$$\mathbf{X} = \begin{pmatrix} p_x^{(1)} \\ p_y^{(1)} \\ p_z^{(1)} \\ v_x^{(1)} \\ v_y^{(1)} \\ v_z^{(1)} \\ p_x^{(2)} \\ p_y^{(2)} \\ p_z^{(2)} \\ v_x^{(2)} \\ v_y^{(2)} \\ v_z^{(2)} \\ \vdots \end{pmatrix} \quad f(\mathbf{X},t) = \begin{pmatrix} v_x^{(1)} \\ v_y^{(1)} \\ v_z^{(1)} \\ \frac{1}{m_1} F_x^{(1)}(\mathbf{X},t) \\ \frac{1}{m_1} F_y^{(1)}(\mathbf{X},t) \\ \frac{1}{m_1} F_z^{(1)}(\mathbf{X},t) \\ v_x^{(2)} \\ v_y^{(2)} \\ v_z^{(2)} \\ \frac{1}{m_2} F_x^{(2)}(\mathbf{X},t) \\ \frac{1}{m_2} F_y^{(2)}(\mathbf{X},t) \\ \frac{1}{m_2} F_z^{(2)}(\mathbf{X},t) \\ \vdots \end{pmatrix}$$

more generally, we can define $X$ as a huge vector storing the current state of *all* particles in a system

## Questions?



http://en.wikipedia.org/wiki/File:Particle_sys_fire.jpg

Note: current state $X$ can also include color!

and $f(X,t)$ can animate changes in color over time!

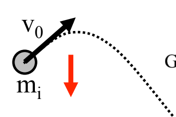
## Today

- Particle Systems
  - Equations of Motion (Physics)
  - Forces: Gravity, Spatial, Damping
  - Numerical Integration (Euler, Midpoint, etc.)
- Mass Spring System Examples
  - String, Hair, Cloth
- Stiffness
- Discretization

## Forces: Gravity

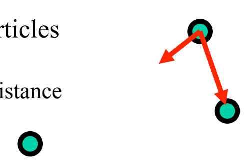
For smoke, flame: make gravity point up!

- Simple gravity: depends only on particle mass




Gravity: $f^{(i)} = \begin{pmatrix} 0 \\ 0 \\ -m_i G \end{pmatrix}$

- N-body problem: depends on all other particles
  - Magnitude inversely proportional to square distance
  - $F_{ij} = G\, m_i m_j / r^2$



*Quickly gets impractical to compute analytically, and expensive to numerically approximate too!*

## Forces: Spatial Fields

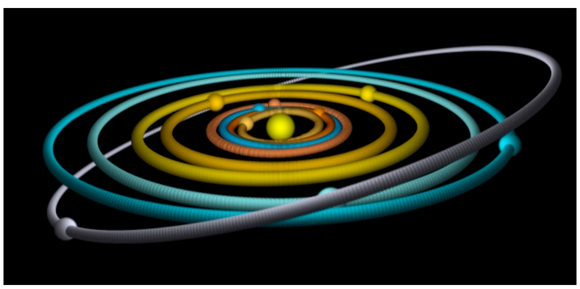
- Force on particle $i$ depends only on position of $i$
  - wind
  - attractors
  - repulsers
  - vortices
- Can depend on time (e.g., wind gusts)
- Note: these forces will generally add energy to the system, and thus may need damping…

## Forces: Damping

$$f^{(i)} = -dv^{(i)}$$

- Force on particle $i$ depends only on velocity of $i$
- Force opposes motion
  - A hack mimicking real-world friction/drag
- Removes energy, so system can settle
- Small amount of damping can stabilize solver
- Too much damping makes motion too glue-like

## Questions?



http://www.lactamme.polytechnique.fr/Mosaic/images/NCOR.U1.2048.D/display.html

## Today

- Particle Systems
  - Equations of Motion (Physics)
  - Forces: Gravity, Spatial, Damping
  - Numerical Integration (Euler, Midpoint, etc.)
- Mass Spring System Examples
  - String, Hair, Cloth
- Stiffness
- Discretization

## Euler's Method

- Examine $f(\mathbf{X},t)$ at (or near) current state
- Take a step of size $h$ to new value of $\mathbf{X}$:

$$t_1 = t_0 + h$$
$$\mathbf{X}_1 = \mathbf{X}_0 + h\, f(\mathbf{X}_0, t_0)$$

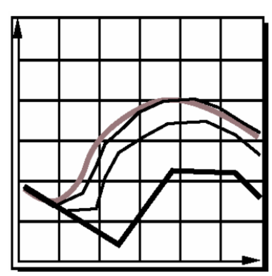
$$\mathbf{X} = \begin{pmatrix} x \\ v \end{pmatrix} \qquad f(X,t) = \begin{pmatrix} v \\ \frac{1}{m} F(x,v,t) \end{pmatrix}$$

update the position by adding a little bit of the *current* velocity
&
update the velocity by adding a little bit of the *current* acceleration

- Piecewise-linear approximation to the curve

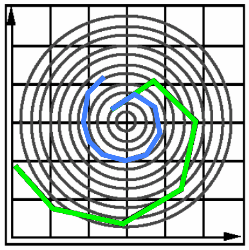
## Effect of Step Size

- Step size controls accuracy
- Smaller steps more closely follow curve
- For animation, we may want to take many small steps per frame
  - How many frames per second for animation?
  - How many steps per frame?



## Euler's Method: Inaccurate

- Simple example: particle in stable circular orbit around planet (origin)
- Current velocity is always tangent to circle
- Force is perpendicular to circle
- Euler method will spiral outward no matter how small *h* is



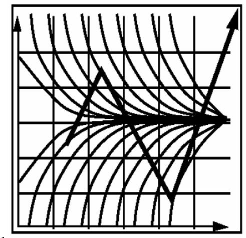
## Euler's Method: Unstable

- Problem: $f(x,t) = -kx$
- Solution: $x(t) = x_0 e^{-kt}$
- Limited step size:

$$x_1 = x_0(1 - hk)$$

$$\begin{cases} h \le 1/k & \text{ok} \\ h > 1/k & \text{oscillates} \pm \\ h > 2/k & \text{explodes} \end{cases}$$

- If *k* is big, *h* must be small



## Analysis using Taylor Series

- Expand exact solution **X***(t)*

$$\mathbf{X}(t_0+h) = \mathbf{X}(t_0) + h\left(\tfrac{d}{dt}\mathbf{X}(t)\right)\Big|_{t_0} + \tfrac{h^2}{2!}\left(\tfrac{d^2}{dt^2}\mathbf{X}(t)\right)\Big|_{t_0} + \tfrac{h^3}{3!}(\cdots) + \cdots$$

- Euler's method:

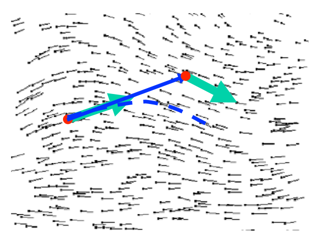
$$\mathbf{X}(t_0+h) = \mathbf{X}_0 + h\, f(\mathbf{X}_0, t_0) \qquad \ldots + O(h^2)\,\text{error}$$

$$h \to h/2 \;\Rightarrow\; error \to error/4 \,\text{per step} \times \text{twice as many steps}$$
$$\to error/2$$

- First-order method: Accuracy varies with *h*
  - To get 100x better accuracy need 100x more steps

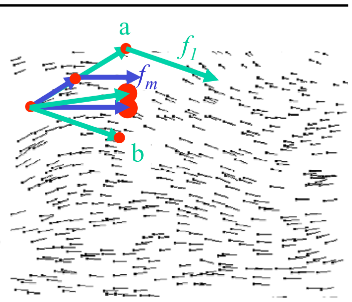
## Can we do better than Euler's Method?

- Problem: *f* has varied along the step
- Idea: look at *f* at the arrival of the step and compensate for variation



## 2nd-Order Methods

- Midpoint:
  - ½ Euler step
  - evaluate $f_m$
  - full step using $f_m$
- Trapezoid:
  - Euler step (a)
  - evaluate $f_1$
  - full step using $f_1$ (b)
  - average (a) and (b)

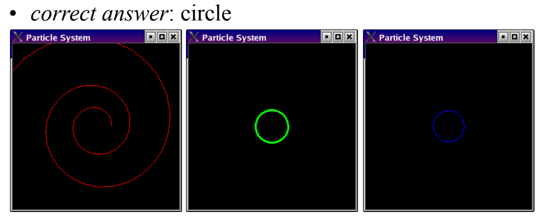



- Midpoint & trapezoid do not yield exactly the same result, but they have same order of accuracy

## Comparison: Euler, Midpoint, Runge-Kutta

A 4th order method!

- *initial position*: (1,0,0)
- *initial velocity*: (0,5,0)
- *force field*: pulls particles to origin with magnitude proportional to distance from origin
- *correct answer*: circle

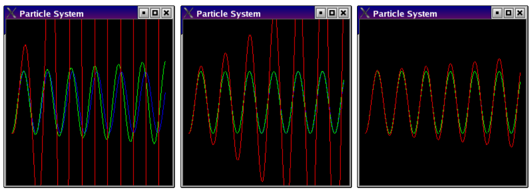



Euler will always diverge (even with small dt)

## Comparison: Euler, Midpoint, Runge-Kutta

A 4th order method!

- *initial position*: (0,-2,0)
- *initial velocity*: (1,0,0)
- *force field*: pulls particles to line y=0 with magnitude proportional to distance from line
- *correct answer*: sine wave




Decreasing the timestep (dt) improves the accuracy

## Questions?

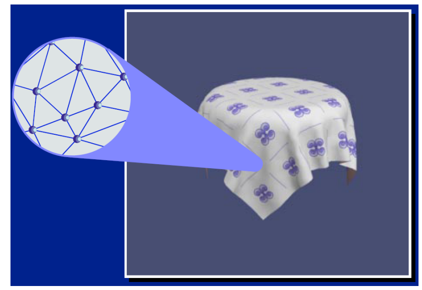


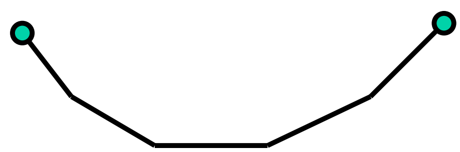
Image by
Baraff, Witkin, Kass

## Today

- Particle Systems
  - Equations of Motion (Physics)
  - Numerical Integration (Euler, Midpoint, etc.)
  - Forces: Gravity, Spatial, Damping
- Mass Spring System Examples
  - String, Hair, Cloth
- Stiffness
- Discretization

## How would you simulate a string?

- Each particle is linked to two particles
- Forces try to keep the distance between particles constant
- What force?

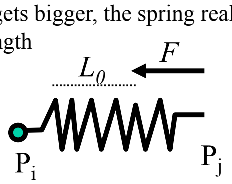


## Spring Forces

- Force in the direction of the spring and proportional to difference with rest length $L_0$

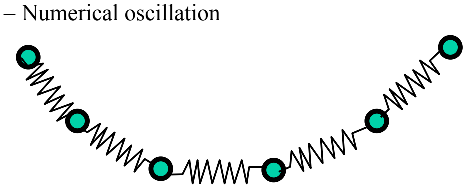
$$F(P_i, P_j) = K(L_0 - ||\vec{P_i P_j}||)\frac{\vec{P_i P_j}}{||\vec{P_i P_j}||}$$

- K is the stiffness of the spring
  - When K gets bigger, the spring really wants to keep its rest length



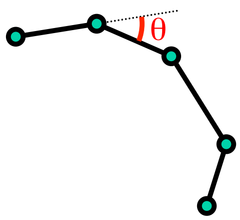

## How would you simulate a string?

- Springs link the particles
- Springs try to keep their rest lengths and preserve the length of the string
- Problems?
  - Stretch, actual length will be greater than rest length
  - Numerical oscillation



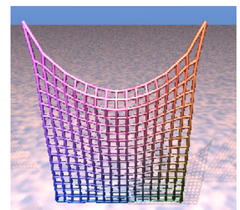
## How would you simulate hair?

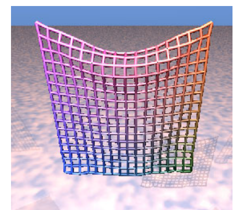
- Similar to string…
- Also… add deformation forces proportional to the angle between segments (hair wants to stay straight or curly)




## Reading for Today

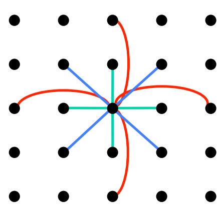
- "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior", Provot, 1995.


Simple mass-spring system


Improved solution

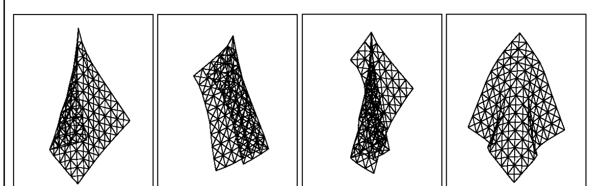
## Cloth Modeled with Mass-Spring

- Network of masses and springs
- Structural springs:
  - link (i, j) & (i+1, j) and (i, j) & (i, j+1)
- Shear springs
  - link (i, j) & (i+1, j+1) and (i+1, j) & (i, j+1)
- Flexion (Bend) springs
  - link (i, j) & (i+2, j) and (i, j) & (i, j+2)



- Be careful not to index out of bounds on the cloth edges!

## Questions?


*Interactive Animation of Structured Deformable Objects*
Desbrun, Schröder, & Barr 1999

## Today

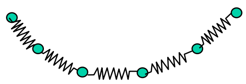
- Particle Systems
  - Equations of Motion (Physics)
  - Numerical Integration (Euler, Midpoint, etc.)
  - Forces: Gravity, Spatial, Damping
- Mass Spring System Examples
  - String, Hair, Cloth
- Stiffness
- Discretization

## The Stiffness Issue

- What relative stiffness do we want for the different springs in the network?
- Cloth is barely elastic, shouldn't stretch so much!
- Inverse relationship between stiffness & $\Delta t$
- We really want constraints (not springs)
- Many numerical solutions
  - reduce $\Delta t$
  - use constraints
  - implicit integration
  - …

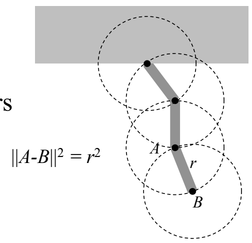
## How would you simulate a string?

- Springs link the particles. Problems?
  - Stretch, actual length will be greater than rest length
  - Numerical oscillation



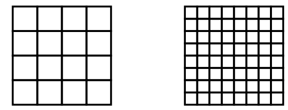
- Rigid, fixed-length bars link the particles
  - Dynamics &
  - Constraints

(must be solved simultaneously non-trivial, even for tiny systems)




## The Discretization Problem

- What happens if we discretize our cloth more finely, or with a different mesh structure?



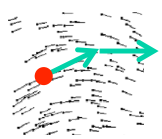
- Do we get the same behavior?
  - Usually not! It takes a lot of effort to design a scheme that does not depend on the discretization.
- Using (explicit) Euler, how many timesteps before a force propagates across the mesh?

## Explicit vs. Implicit Integration

- With an explicit/forward integration scheme:

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\mathbf{g}(\mathrm{y}_k)$$

we must use a very small timestep to simulate *stable, stiff* cloth.



- Alternatively we can use an implicit/backwards scheme:

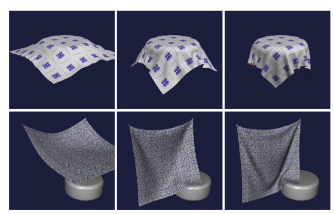
$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\mathbf{g}(\mathrm{y}_{k+1})$$
$$\mathbf{y}_k = \mathbf{y}_{k+1} - h\,\mathbf{g}(\mathrm{y}_{k+1})$$

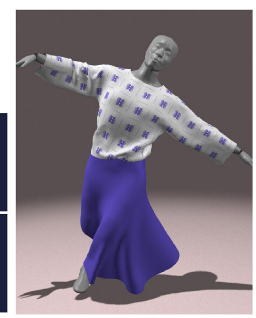
Solving one step is much more expensive (Newton's Method, Conjugate Gradients, …) but overall faster than the thousands of explicit timesteps required for very stiff springs.

## Questions?

David Baraff & Andrew Witkin
*Large Steps in Cloth Simulation*
SIGGRAPH 1998

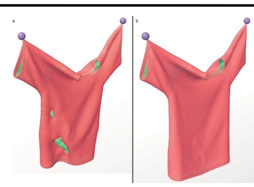
- Dynamic motion driven by animation

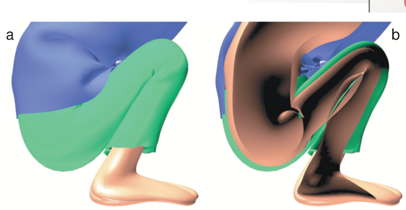




## Optional Reading for Today:

- Baraff, Witkin & Kass
  *Untangling Cloth*
  SIGGRAPH 2003

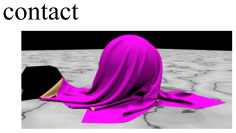


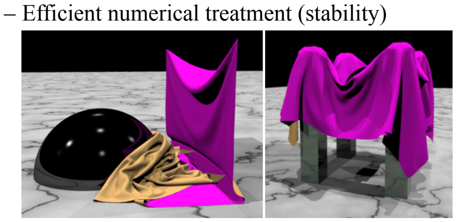



## Cloth Collision

Robert Bridson, Ronald Fedkiw & John Anderson
*Robust Treatment of Collisions, Contact and Friction for Cloth Animation*
SIGGRAPH 2002

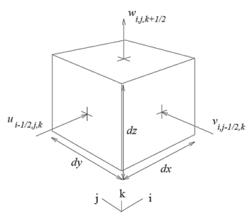
- A cloth has many points of contact
- Stays in contact
- Requires
  - Efficient collision detection
  - Efficient numerical treatment (stability)

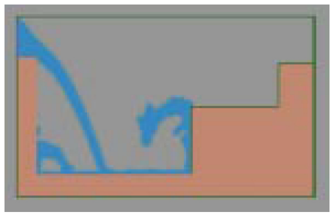




## Reading for Tuesday:

- "Realistic Animation of Liquids", Foster & Metaxas, 1996



- Post a comment or question on the LMS discussion by 10pm Monday

## HW2: Cloth & Fluid Simulation