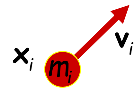
Particle Dynamics

Set of particles modeled as point masses in motion



- m_i : mass of particle i
- $\mathbf{x}_i$: position of particle i
- $\mathbf{v}_i$: velocity of particle i

Can write Newton's second law as differential equation

$$\mathbf{f}_i(t) = m_i \mathbf{a}_i(t)$$

so

$$\ddot{\mathbf{x}}_i(t) = \frac{\mathbf{f}_i(t)}{m_i}$$

velocity $\mathbf{v}_i(t) = \dfrac{d\mathbf{x}_i(t)}{dt} = \dot{\mathbf{x}}_i(t)$

acceleration $\mathbf{a}_i(t) = \dfrac{d\mathbf{v}_i(t)}{dt} = \dfrac{d^2\mathbf{x}_i(t)}{dt^2} = \ddot{\mathbf{x}}_i(t)$

$\mathbf{f}_i$: sum of all forces acting on particle

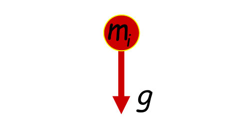
Gravity

Select a "down" direction

- Here, we'll assume that the y-axis points up

Force due to gravity is simply

$$\mathbf{f}_i = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}$$



- g : gravitational constant
  - ≈ 9.78 m/sec^2 on Earth

Deformable Models

Continuum mechanics

- Deformable solid models
  - *Cloth*
  - *Rubber*
  - *Soft tissues (muscle, skin, hair, …)*
- Fluid models
  - *Water (oceans, puddles, rain, …)*
- Gas-like models
  - *Steam, smoke, fire, …*

Physical Principles

Deformation

- Strain

Force

- Stress

Constitutive law

- Hooke's Law: Stress = Elasticity x Strain

Newton's law of motion

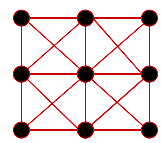
- Acceleration = Mass^{-1} x Stress

Deformable Solids: Mass-Spring-Damper Systems

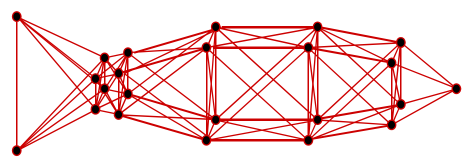
Useful for building deformable models

1-dimensional:

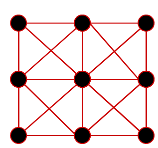
2-dimensional:

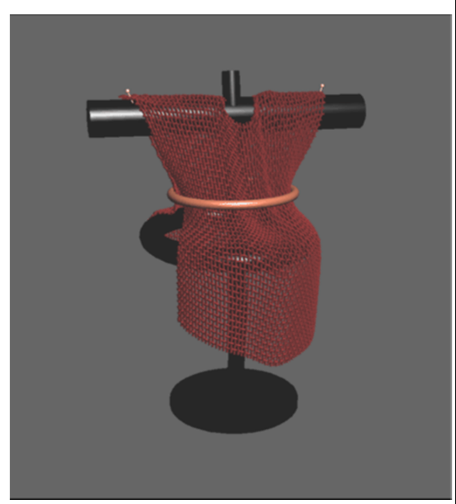


3-dimensional:



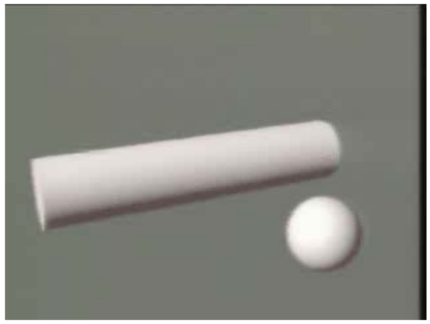
Physics-Based Cloth Models





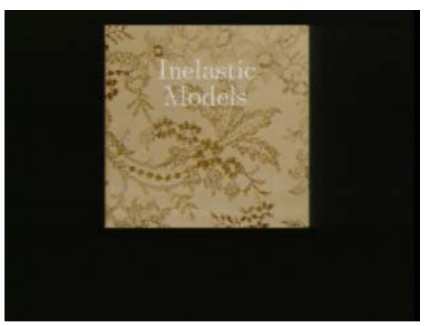
Flying Carpet

Gravity and collision forces (1987)



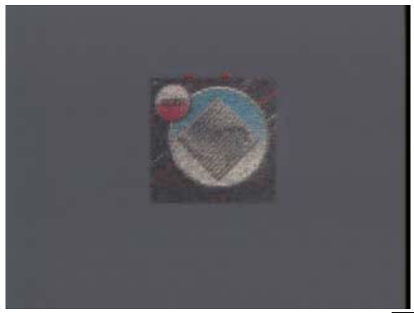
Curtain

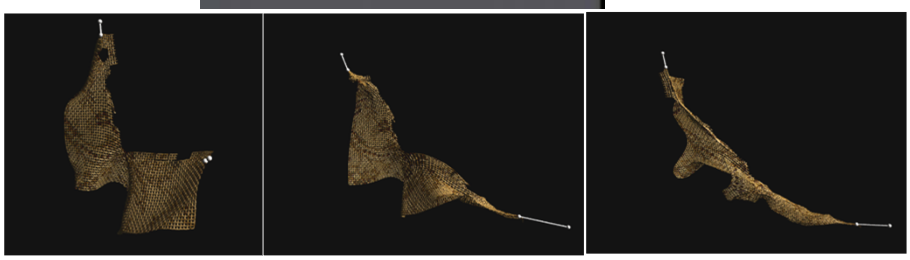
(1987)



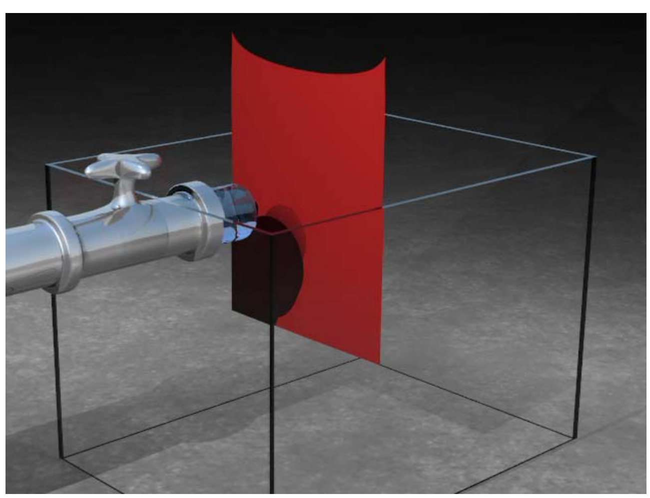
Ripping Cloth

(1988)

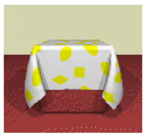


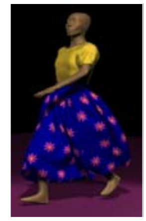


Cloth-Fluid Interaction



Other Uses of Mass-Spring Systems: Cloth Simulation





Physics-Based Facial Simulation with Mass-Spring-Damper Systems

Data Primitives

Node

- A lumped mass 
  - *Mass:* m
  - *Damping:* γ
  - *Position:* $\boldsymbol{x}(t) = [x(t), y(t), z(t)]^T$
  - *Velocity:* $\boldsymbol{v}(t) = d\boldsymbol{x}(t) / dt$
  - *Acceleration:* $\boldsymbol{a}(t) = d^2\boldsymbol{x}(t) / dt^2$
  - *Nodal force:* $\boldsymbol{f}(t)$

Spring

- Connects a pair of nodes 
  - *Rest length:* l
  - *Stiffness:* c

Equations of Motion

Newton's law of motion

- Mass x **Acceleration** = **Net Force**
- Mathematically: for each node $i = 1, 2, \ldots, N$

$$m_i \mathbf{a}_i = \mathbf{f}_i \quad \text{or} \quad m_i \frac{d^2 \mathbf{x}_i}{dt^2} = \mathbf{f}_i$$

- This is a system of second-order ordinary differential equations in time
- The net nodal force is: $\mathbf{f}_i = \mathbf{s}_i - \gamma_i \mathbf{v}_i + \boldsymbol{g}_i$
  - *Gravity:* $\boldsymbol{g}_i$
  - *Damping force:* $-\gamma_i \mathbf{v}_i$ *(nodal drag)*
  - *Spring force:* $\boldsymbol{s}_i$

Spring Force

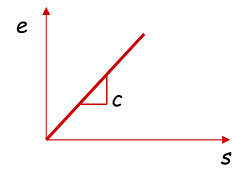
Net spring force at node i is the sum of forces due to springs connecting node i to neighboring nodes j

- Denoting the neighbors of node i as N_i

$$\mathbf{s}_i(t) = \sum_{j \in N_i} \mathbf{s}_{ij}$$

Spring force

$$\mathbf{s}_{ij} = c_{ij} e_{ij} \frac{\mathbf{r}_{ij}}{\left\| \mathbf{r}_{ij} \right\|}$$

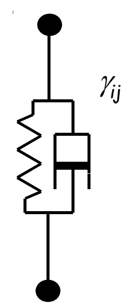


- $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ is the separation of the two nodes
- $\|\mathbf{r}_{ij}\|$ is the actual length of the spring
- $e_{ij} = \|\mathbf{r}_{ij}\| - l_{ij}$ is the deformation of the spring
- Force varies linearly with deformation (but not with node positions)

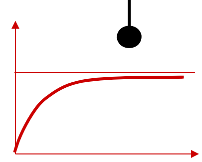
A Damped Spring

Parallel combination of spring and damper

- Known as Voigt model
- Damping coefficient γ_{ij}



$$\mathbf{s}_{ij} = (c_{ij} e_{ij} - \gamma_{ij} \frac{de_{ij}}{dt}) \frac{\mathbf{r}_{ij}}{\left\| \mathbf{r}_{ij} \right\|}$$



Note: $\frac{de_{ij}}{dt} = \mathbf{v}_{ij} \cdot \frac{\mathbf{r}_{ij}}{\left\| \mathbf{r}_{ij} \right\|}$ $\quad \mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$

Finite Differences

Discretization of time

- $t_i = i\,\Delta t \;\; = \; 0, \Delta t, 2\Delta t, \dots$

***First finite differences of a function* f**

- Let $f^i = f(t_i)$, for $i = 0, 1, \dots$
- Forward difference: $$\frac{df(t)}{dt} \approx \frac{f^{t+1} - f^t}{\Delta t}$$
- Backward difference: $$\frac{df(t)}{dt} \approx \frac{f^t - f^{t-1}}{\Delta t}$$
- Central difference: $$\frac{df(t)}{dt} \approx \frac{f^{t+1} - f^{t-1}}{2\Delta t}$$

Disretization of Nodal Motion

***Finite difference approximation of motion of node* i**

- Velocity $$\mathbf{v}_i(t) = \frac{d\mathbf{x}_i(t)}{dt} \approx \frac{\mathbf{x}_i^{t+1} - \mathbf{x}_i^t}{\Delta t}$$
- Acceleration $$\mathbf{a}_i(t) = \frac{d\mathbf{v}_i(t)}{dt} \approx \frac{\mathbf{v}_i^{t+1} - \mathbf{v}_i^t}{\Delta t}$$
  - *Or,*

$$\mathbf{a}_i(t) = \underbrace{\frac{\mathbf{v}_i^t - \mathbf{v}_i^{t-1}}{\Delta t}}_{\text{Backward Difference}} = \underbrace{\frac{\mathbf{x}_i^{t+1} - 2\mathbf{x}_i^t + \mathbf{x}_i^{t-1}}{(\Delta t)^2}}_{\text{Central 2}^{\text{nd}}\text{ Difference}}$$

Integrating the Equations of Motion Through Time

The explicit Euler time-integration method

- For each node *i* do:
  - *Step 1:* $\mathbf{a}_i^t = \dfrac{\mathbf{f}_i^t}{m_i}$
  - *Step 2:* $\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \Delta t\, \mathbf{a}_i^t$
  - *Step 3:* $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta t\, \mathbf{v}_i^{t+1}$

Computing the Spring Forces

What is the best way?

- Access each spring *ij* in sequence
- Compute spring force

$$\mathbf{s}_{ij}^t = \left(c_{ij} e_{ij}^t - \frac{\gamma_{ij}}{\Delta t}(e_{ij}^t - e_{ij}^{t-1}) \right) \frac{\mathbf{r}_{ij}^t}{\left\| \mathbf{r}_{ij}^t \right\|}$$

- Accumulate force on nodes *i* and *j*

$$\mathbf{f}_i^t = \mathbf{f}_i^{t-1} + \mathbf{s}_{ij}^t$$

$$\mathbf{f}_j^t = \mathbf{f}_j^{t-1} - \mathbf{s}_{ij}^t$$

Other Time-Integration Methods

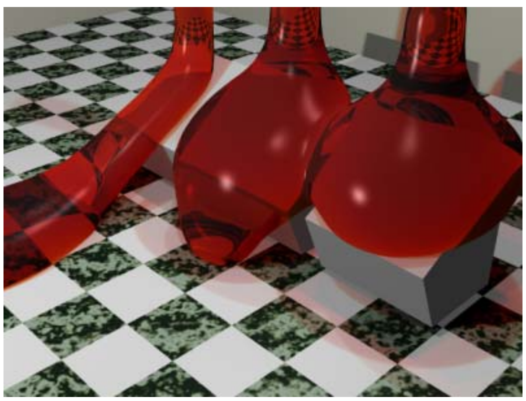
There are more stable and/or accurate explicit methods than the Euler method

- E.g., the Runge-Kutta method

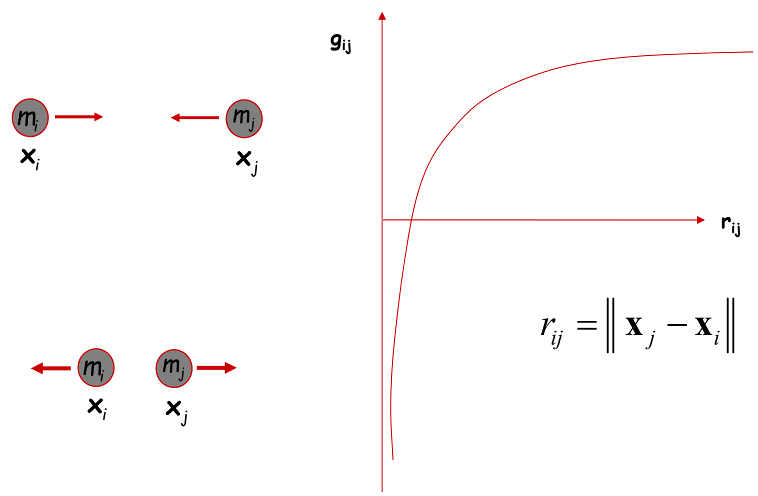
Implicit methods are stable

- The implicit Euler method is obtained using backward finite differences
- Implicit methods require the solution of systems of linear equations at each time step
- They are too complicated for us to cover in this introductory graphics course

Fluid Flow Simulation



Lenard-Jones Force Profile



$$r_{ij} = \left\| \mathbf{x}_j - \mathbf{x}_i \right\|$$

Discrete Fluid Model

The total force on a particle i due to all other particles:

$$\mathbf{g}_i(t) = \sum_{j \neq i} \mathbf{g}_{ij}(t)$$

$$\mathbf{g}_{ij}(t) = m_i m_j (\mathbf{x}_i - \mathbf{x}_j)\left(-\frac{\alpha}{(r_{ij} + \varepsilon)^a} + \frac{\beta}{r_{ij}^{\ b}} \right) \qquad r_{ij} = \left\| \mathbf{x}_j - \mathbf{x}_i \right\|$$

α and β determine the strength of the attraction and repulsion forces

Exponents a = 2, b = 4

ε is minimum required separation of particles

Rigid-Body Dynamics

To create a nearly rigid object using a mass-spring-damper system, make the springs really stiff

- This works in principle, but leads to numerical instability in practice

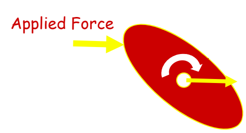
Better to use rigid-body dynamics

- There are no such things as perfectly rigid bodies in the real world, so this is an approximation

When a force is applied to extended bodies, the movement induced can consist of both translation and rotation

- Rotation is modeled explicitly in rigid-body dynamics
- A force applied other than at the ***center of mass*** (COM) of the extended body produces a ***torque***

Rigid Body Dynamics



Kinematics of 3D body in space

- Three translational degrees of freedom: $\mathbf{x}$
- Three rotational degrees of freedom: θ

Inertia tensor

- Specifies how mass is distributed about the COM

Equations of motion

$$m\mathbf{a} = \mathbf{f}$$

$$\frac{d}{dt}\mathbf{I}\underbrace{\boldsymbol{\omega}}_{\text{Angular Velocity } d\theta/dt} = \overbrace{\boldsymbol{\tau}}^{\text{Torque}}$$

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

where

$$I_{xx} = \int (y^2 + z^2)\,dm \qquad I_{xy} = \int xy\,dm$$

$$I_{yy} = \int (x^2 + z^2)\,dm \qquad I_{xz} = \int xz\,dm$$

$$I_{zz} = \int (x^2 + y^2)\,dm \qquad I_{yz} = \int yz\,dm$$

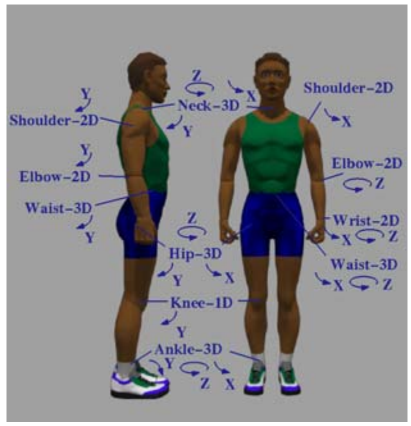
Articulated Dynamics

Rigid bodies with joints

- A.k.a. constrained multibody systems

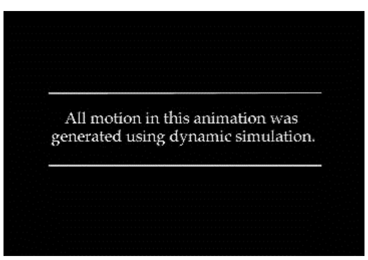
Dynamic human model

- J. Hodgins, et al. GATech
- 15-17 rigid body parts
- 22-32 controlled dofs
- Body part densities from anthropometric data
- Masses & moments calculated from polygonal model



"Atlanta in Motion"

J. Hodgins, et al., Georgia Tech



Falling Backward, Rolling Over, Rising, and Balancing in Gravity

Help, I've fallen! *and I can get up!!*

Rising From a Supine Position

The Virtual Stuntman Does a Kip Stunt

The Virtual Stuntman: A Suicidal Dive Down Stairs

Behavioral Animation

Closely related to procedural animation

- Procedures based on ethological principles
  - *Artificial Life*

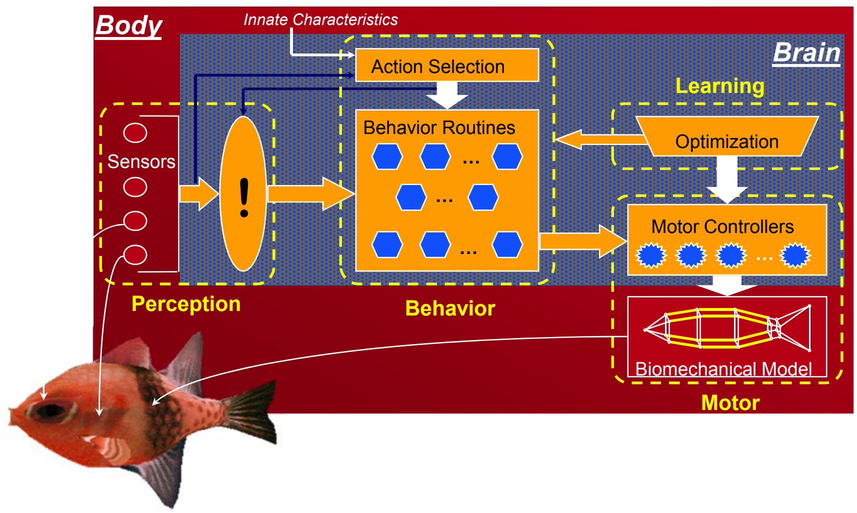
A common example of this approach is flocking (or schooling, herding, crowds)

- Motion of an ***agent*** is determined by others nearby
- Simple rules lead to interesting ***emergent behaviors***
- Very helpful for choreographing large-scale action
- Wildebeests in “The Lion King”
- Army of mounted soldiers in “Mulan”
- Flying bats in “Batman”

Behavioral Animation

An army of orcs from the “Lord of the Rings” trilogy

An Artificial Fish Model



Go Fish !

(Produced for the SIGGRAPH Electronic Theater)