CS 283
Advanced Computer Graphics

Simulation Basics

James F. O'Brien

Associate Professor
U.C. Berkeley

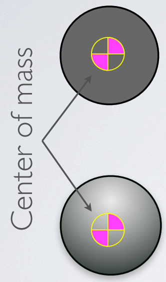
A Rigid Body

- A solid object that does not deform
  - Consists of infinite number of infinitesimal mass points...
  - ...that share a single RB transformation
    - Rotation + Translation (no shear or scale)

$$\boldsymbol{x}^W = \boldsymbol{R} \cdot \boldsymbol{x}^L + \boldsymbol{t}$$

    - Rotation and translation vary over time
  - Limit of deformable object as $k_S \rightarrow \infty$

A Rigid Body



In 2D:

Translation 2 "directions"
Rotation 1 "direction"
3 DOF Total

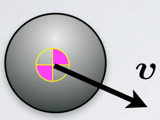
In 3D:

Translation 3 "directions"
Rotation 3 "direction"
6 DOF Total

Translation and rotation are ***decoupled***

2D is boring... we'll stick to 3D from now on...

Translational Motion

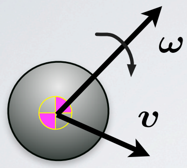


Just like a point mass:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}$$
$$\dot{\boldsymbol{v}} = \boldsymbol{a} = \boldsymbol{f}/m$$

Note: Recall discussion on integration...

Rotational Motion

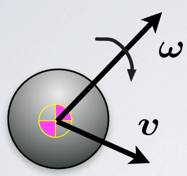


Rotation gets a bit odd, as well see...

Rotational "position" $\boldsymbol{R}$
  Rotation matrix
  Exponential map
  Quaternions

Rotational velocity $\boldsymbol{\omega}$
  Stored as a vector
  (Also called angular velocity...)
  Measured in radians / second

Rotational Motion



Kinetic energy due to rotation:

"Sum energy (from rotation) over all points in the object"

$$E = \int_\Omega \frac{1}{2} \rho \, \dot{\boldsymbol{x}} \cdot \dot{\boldsymbol{x}} \; du$$

$$E = \int_\Omega \frac{1}{2} \rho ([\omega \times] \boldsymbol{x}) \cdot ([\omega \times] \boldsymbol{x}) \; du$$

Rotational Motion

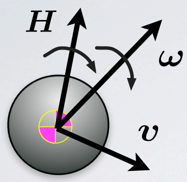


Figure is a lie if this really is a sphere...

Angular momentum
- Similar to linear momentum
- Can be derived from rotational energy

$$\boldsymbol{H} = \partial \boldsymbol{E} / \partial \boldsymbol{\omega}$$

$$\boldsymbol{H} = \int_\Omega \rho\, \boldsymbol{x} \times \dot{\boldsymbol{x}}\, du$$

$$\boldsymbol{H} = \int_\Omega \rho\, \boldsymbol{x} \times (\omega \times \boldsymbol{x})\, du$$

$$\boldsymbol{H} = \left(\int_\Omega \cdots \, du \right) \boldsymbol{\omega}$$

$$\boldsymbol{H} = \boldsymbol{I}\boldsymbol{\omega}$$

"Inertia Tensor" not identity matrix...

Inertia Tensor

$$\mathbf{I} = \int_\Omega \rho \begin{bmatrix} y^2+z^2 & -xy & -xz \\ -xy & z^2+x^2 & -yz \\ -xz & -yz & x^2+y^2 \end{bmatrix} \mathrm{d}u$$

See example for simple shapes at
http://scienceworld.wolfram.com/physics/MomentofInertia.html

Can also be computed from polygon models by transforming volume integral to a surface one.
See paper/code by Brian Mirtich.

Rotational Motion

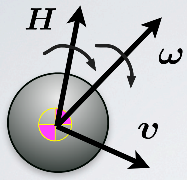


Figure is a lie if this really is a sphere...

Conservation or momentum:

$$\boldsymbol{H}^W = \boldsymbol{I}^W \boldsymbol{\omega}^W$$

$$\boldsymbol{H}^W = \boldsymbol{R}\boldsymbol{I}^L\boldsymbol{R}^\mathsf{T}\boldsymbol{\omega}^W$$

$$\dot{\boldsymbol{H}}^W = \dot{\boldsymbol{R}}\boldsymbol{I}^L\boldsymbol{R}^\mathsf{T}\boldsymbol{\omega}^W + \cancel{\boldsymbol{R}\boldsymbol{I}^L\dot{\boldsymbol{R}}^\mathsf{T}}\boldsymbol{\omega}^W + \boldsymbol{R}\boldsymbol{I}^L\boldsymbol{R}^\mathsf{T}\boldsymbol{\alpha}^W$$

$$\dot{\boldsymbol{H}}^W = 0$$

$$\dot{\boldsymbol{R}} = \boldsymbol{\omega} \times \boldsymbol{R}$$

$$\boldsymbol{\alpha}^W = (\boldsymbol{R}\boldsymbol{I}^L\boldsymbol{R}^\mathsf{T})^{-1}(-\boldsymbol{\omega}^W \times \boldsymbol{H}^W)$$

In other words, things wobble when they rotate.

Rotational Motion

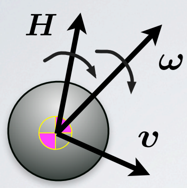


Figure is a lie if this really is a sphere...

$$\dot{\boldsymbol{R}} = [\omega \times]\boldsymbol{R}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha}$$

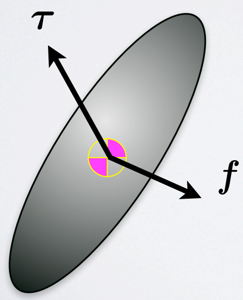
$$\boldsymbol{\alpha}^W = (\boldsymbol{R}\boldsymbol{I}^L\boldsymbol{R}^\mathsf{T})^{-1}\left((-\boldsymbol{\omega}^W \times \boldsymbol{H}^W) + \boldsymbol{\tau}\right)$$

$$\boldsymbol{\tau} = \boldsymbol{f} \times \boldsymbol{x}$$

Take care when integrating rotations, they need to stay rotations.

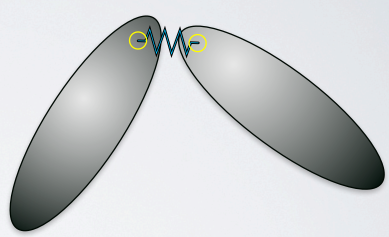
Couples

- A force / torque pair is a couple
  - Also a wrench (I think)
- Many couples are equivalent



Constraints

- Simples method is to use spring attachments
  - Basically a penalty method



  - Spring strength required to get good results may be unreasonably high
    - There are ways to cheat in some contexts...

Constraints

- Articulation constraints
  - Spring trick is an example of a full coordinate method
    - Better constraint methods exist
  - Reduced coordinate methods use DOFs in kinematic skeleton for simulation
    - Much more complex to explain
- Collisions
  - Penalty methods can also be used for collisions
  - Again, better constraint methods exist

A Simple Spring

- Ideal ***zero***-length spring

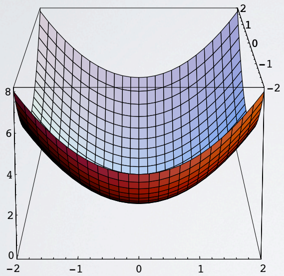
$$\boldsymbol{f}_{a\to b} = k_s(\boldsymbol{b} - \boldsymbol{a})$$

$$\boldsymbol{f}_{b\to a} = -\boldsymbol{f}_{a\to b}$$

- Force pulls points together
- Strength proportional to distance

A Simple Spring

- Energy potential



$$E = 1/2\; k_s(\boldsymbol{b} - \boldsymbol{a}) \cdot (\boldsymbol{b} - \boldsymbol{a})$$

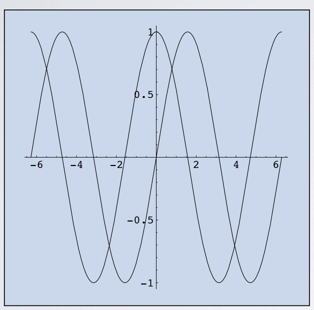
$$\boldsymbol{f}_{a \to b} = k_s(\boldsymbol{b} - \boldsymbol{a})$$

$$\boldsymbol{f}_{b \to a} = -\boldsymbol{f}_{a \to b}$$

$$\boldsymbol{f}_a = -\nabla_a E = -\left[\frac{\partial E}{\partial a_x}, \frac{\partial E}{\partial a_y}, \frac{\partial E}{\partial a_z}\right]$$

A Simple Spring

- Energy potential: kinetic **vs** elastic



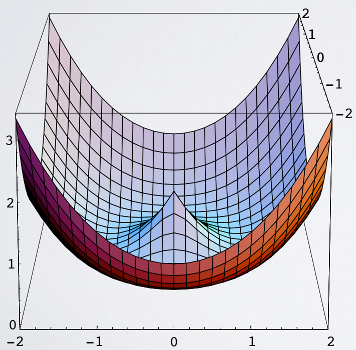
$$E = 1/2\; k_s(\boldsymbol{b} - \boldsymbol{a}) \cdot (\boldsymbol{b} - \boldsymbol{a})$$

$$E = 1/2\; m(\dot{\boldsymbol{b}} - \dot{\boldsymbol{a}}) \cdot (\dot{\boldsymbol{b}} - \dot{\boldsymbol{a}})$$

Non-Zero Length Springs

$$\boldsymbol{f}_{a \to b} = k_s \frac{\boldsymbol{b} - \boldsymbol{a}}{||\boldsymbol{b} - \boldsymbol{a}||} \left(||\boldsymbol{b} - \boldsymbol{a}|| - l \right)$$

Rest length



$$E = k_s \left(||\boldsymbol{b} - \boldsymbol{a}|| - l \right)^2$$

Comments on Springs

- Springs with zero rest length are linear
- Springs with non-zero rest length are nonliner
  - Force ***magnitude*** linear w/ discplacement (from rest length)
  - Force direction is non-linear
  - Singularity at $||\boldsymbol{b} - \boldsymbol{a}|| = 0$

Damping

- "Mass proportional" damping

$$\boldsymbol{f} = -k_d \dot{\boldsymbol{a}}$$

  - Behaves like viscous drag on all motion
  - Consider a pair of masses connected by a spring
    - How to model rusty *vs* oiled spring
    - Should internal damping slow group motion of the pair?
  - Can help stability... up to a point

Damping

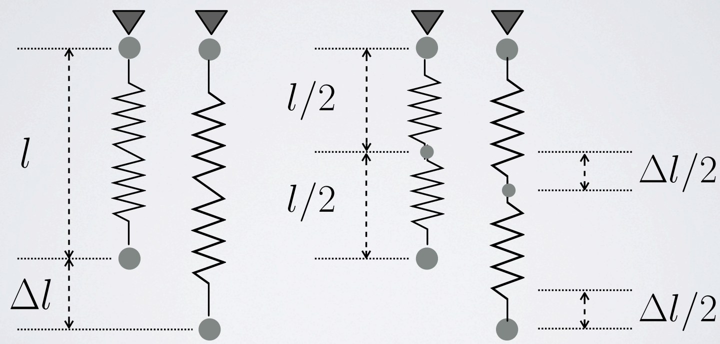
- "Stiffness proportional" damping

$$\boldsymbol{f}_a = -k_d \frac{\boldsymbol{b}-\boldsymbol{a}}{||\boldsymbol{b}-\boldsymbol{a}||^2} (\boldsymbol{b}-\boldsymbol{a}) \cdot (\dot{\boldsymbol{b}}-\dot{\boldsymbol{a}})$$

  - Behaves viscous drag on change in spring length
  - Consider a pair of masses connected by a spring
    - How to model rusty *vs* oiled spring
    - Should internal damping slow group motion of the pair?

Spring Constants

- Two ways to model a single spring



Spring Constants

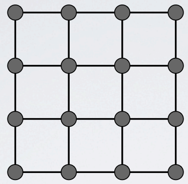
- Constant k_s gives inconsistent results with different discretizations
- Change in length is not what we want to measure
- Strain: change in length as fraction of original length

$$\epsilon = \frac{\Delta l}{l_0}$$

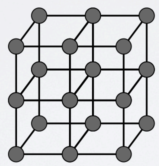
Nice and simple for 1D...

Structures from Springs

- Sheets



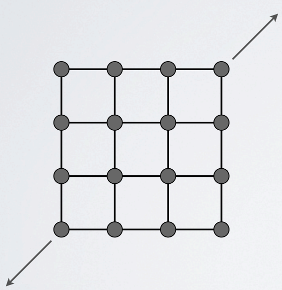
- Blocks



- Others

Structures from Springs

- They behave like what they are (obviously!)

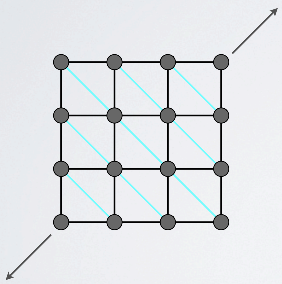


This structure will not resist shearing

This structure will not resist out-of-plane bending either...

Structures from Springs

- They behave like what they are (obviously!)

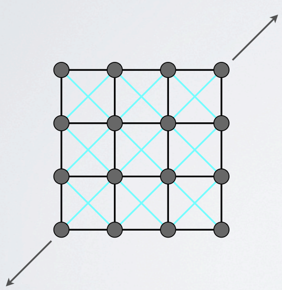


This structure will resist shearing but has anisotopic bias

This structure still will not resist out-of-plane bending

Structures from Springs

- They behave like what they are (obviously!)

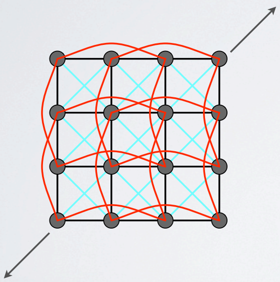


This structure will resist shearing
Less bias
Interference between spring sets

This structure still will not resist out-of-plane bending

Structures from Springs

- They behave like what they are (obviously!)

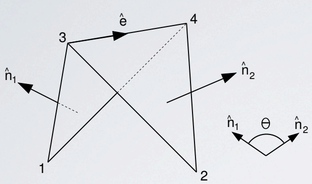


This structure will resist shearing
Less bias
Interference between spring sets

This structure will resist out-of-plane bending
Interference between spring sets
Odd behavior

How do we set spring constants?

Edge Springs



$$u_1 = |E|\frac{N_1}{|N_1|^2} \qquad u_2 = |E|\frac{N_2}{|N_2|^2}$$

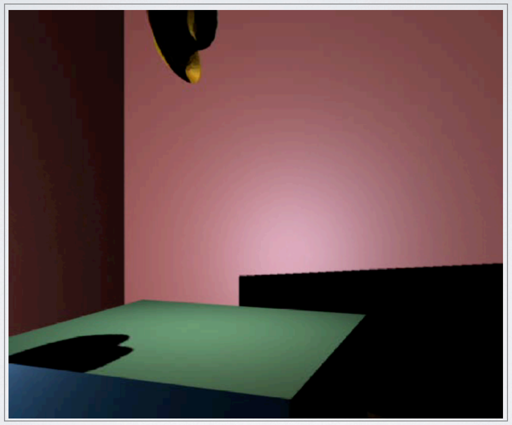
$$u_3 = \frac{(x_1 - x_4)\cdot E}{|E|}\frac{N_1}{|N_1|^2} + \frac{(x_2 - x_4)\cdot E}{|E|}\frac{N_2}{|N_2|^2}$$

$$u_4 = -\frac{(x_1 - x_3)\cdot E}{|E|}\frac{N_1}{|N_1|^2} - \frac{(x_2 - x_3)\cdot E}{|E|}\frac{N_2}{|N_2|^2}$$

$$F_i^e = k^e \frac{|E|^2}{|N_1| + |N_2|}\sin(\theta/2)\, u_i$$

From Bridson ***et al.,*** 2003, also see Grinspun ***et al.,*** 2003

Example: Thin Material



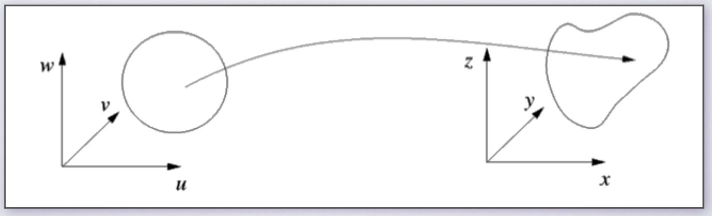
FEM Problem Setup

- Lagrangian Formulation
  - Where in space did this material mode to?
  - Commonly used for solid materials
- Eulerian Formulation
  - What material is at this location in space?
  - Commonly used for fluids

Problem Setup

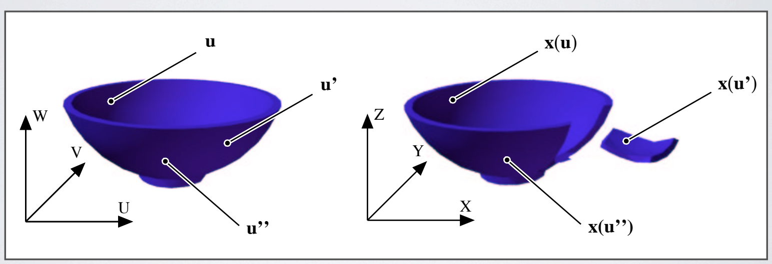
- Lagrangian Formulation
  - Where in space did this material mode to?
  - Commonly used for solid materials

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{u})$$

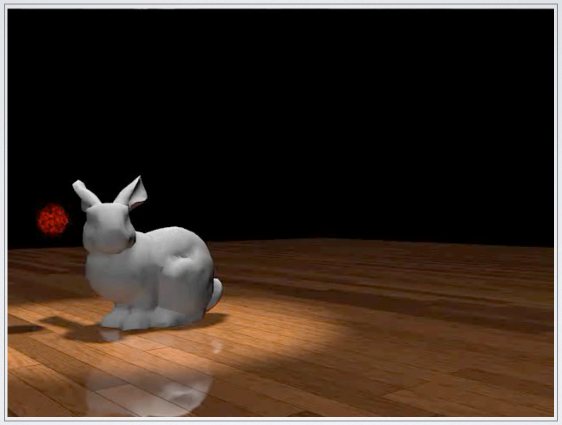


Lagrangian Formulation

- Deformation described by mapping from material (local) to word coordinates



Example

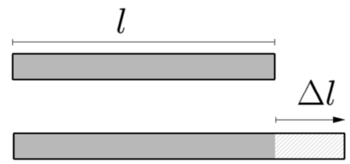


Another Example

Video footage © LucasArts, used with permission.

Strain

- Strain measures deformation
- Purely geometric
- Example: simple strain in a bar



$$\epsilon = \frac{\Delta l}{l}$$

Strain

- Green's strain tensor

$$\epsilon_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i} \cdot \frac{\partial \boldsymbol{x}}{\partial u_j} \right) - \delta_{ij}$$

- Vanishes when not deformed
- Only measures deformation
- Does not depend on the coordinate system

Strain

- Green's strain tensor

$$\epsilon_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i} \cdot \frac{\partial \boldsymbol{x}}{\partial u_j} \right) - \delta_{ij}$$

$$l_x^2 - l_u^2 = \boldsymbol{d} \cdot \boldsymbol{\epsilon} \cdot \boldsymbol{d}$$

Strain

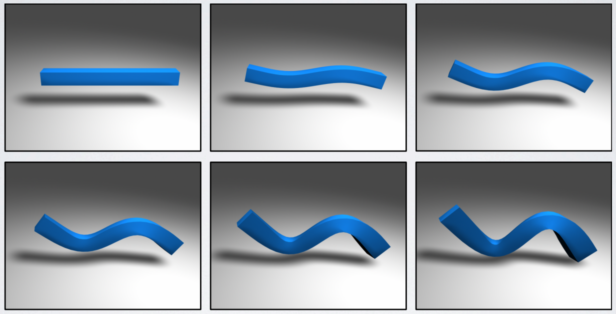
- Cauchy's strain tensor

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial x_i}{\partial u_j} + \frac{\partial x_j}{\partial u_i} \right) - \delta_{ij}$$

- Linearization of Green's strain tensor
- Vanishes when not deformed
- *Not invariant w.r.t rotations*

$$l_x - l_u \approx \boldsymbol{d} \cdot \boldsymbol{\epsilon} \cdot \boldsymbol{d}$$

Linearization Errors



We'll fix this problem later...

Strain Rate

- Time derivative of Green's strain tensor
- Measures rate of deformation
- Used for internal damping

$$\dot{\epsilon}_{ij} = \left(\frac{\partial \boldsymbol{x}}{\partial u_i} \cdot \frac{\partial \dot{\boldsymbol{x}}}{\partial u_j} \right) + \left(\frac{\partial \dot{\boldsymbol{x}}}{\partial u_i} \cdot \frac{\partial \boldsymbol{x}}{\partial u_j} \right)$$

Strain Rate

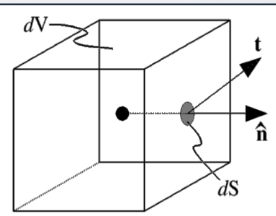
- Time derivative of Cauchy's strain tensor
- Measures rate of deformation
- Used for internal damping

$$\dot{\epsilon}_{ij} = \frac{1}{2}\left(\frac{\partial \dot{x}_i}{\partial u_j} + \frac{\partial \dot{x}_j}{\partial u_i}\right)$$

Stress

- Stress determines internal forces
- Measures how much material "wants" to return to original shape

$$\boldsymbol{t} = \boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}}$$



Stress due to Strain

$$\sigma_{ij}^{(\epsilon)} = \sum_{k=1}^{3} \underline{\lambda \epsilon_{kk} \delta_{ij}} + \underline{2\mu \epsilon_{ij}}$$

Elastic (Lamé) Constants

(in)compressibility

rigidity

Generalization of

$f = kd$

Stress due to Rate

$$\sigma_{ij}^{(\nu)} = \sum_{k=1}^{3} \underline{\psi \dot{\epsilon}_{kk} \delta_{ij}} + \underline{2\psi \dot{\epsilon}_{ij}}$$

Damping Constants

ϕ

ψ

Generalization of

$f = cv$

Energy Potentials

Elastic Energy Density

$$\eta = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}^{(\epsilon)}\epsilon_{ij}$$

Generalization of

$$E = \frac{1}{2}kd^2$$

Kinetic Energy Density

$$\kappa = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sigma_{ij}^{(\nu)}\dot{\epsilon}_{ij}$$

Generalization of

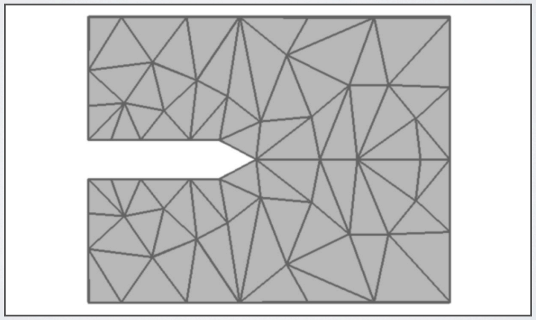
$$E = \frac{1}{2}mv^2$$

Discretization

- Transition from continuous model to something we can compute with...

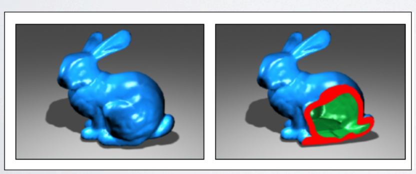
Finite Element Method

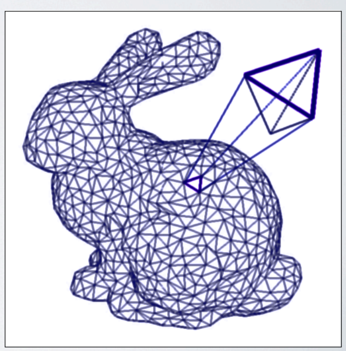
- Disjoint elements tile material domain
- Derivatives from shape functions
- Nodes shared by adjacent elements



Finite Element Method

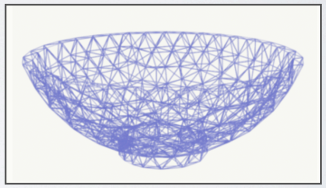
- Disjoint elements tile material domain
- Derivatives from shape functions
- Nodes shared by adjacent elements





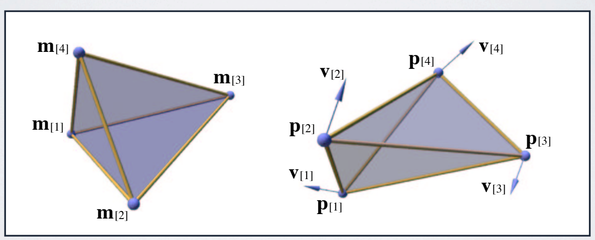
FEM Discretization

- Solid volumes
- Tetrahedral elements
- Linear shape functions



FEM Discretization

- Each element defined by four nodes
- m - location in material (local) coordinates
- p - position in world coordinates
- v - velocity in world coordinates



Element Shape Functions

Barycentric coordinates

$$\begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{m}_{[1]} & \boldsymbol{m}_{[2]} & \boldsymbol{m}_{[3]} & \boldsymbol{m}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix} \boldsymbol{b}$$

Invert to obtain basis matrix

$$\boldsymbol{b} = \boldsymbol{\beta} \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{m}_{[1]} & \boldsymbol{m}_{[2]} & \boldsymbol{m}_{[3]} & \boldsymbol{m}_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1}$$

Material Derivatives

World pos. as function of material coordinates

$$\boldsymbol{x}(\boldsymbol{u}) = \boldsymbol{P}\,\boldsymbol{\beta} \begin{bmatrix} \boldsymbol{u} \\ 1 \end{bmatrix}$$

where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_{[1]} & \boldsymbol{p}_{[2]} & \boldsymbol{p}_{[3]} & \boldsymbol{p}_{[4]} \end{bmatrix}$$

Derivative w.r.t. material coordinates

$$\frac{\partial \boldsymbol{x}}{\partial u_i} = \boldsymbol{P}\,\boldsymbol{\beta}_{\mathrm{col}_i}$$

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial x_i}{\partial u_j} + \frac{\partial x_j}{\partial u_i}\right) - \delta_{ij}$$

Recall

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial x_i}{\partial u_j} + \frac{\partial x_j}{\partial u_i}\right) - \delta_{ij}$$

$$\sigma_{ij}^{(\epsilon)} = \sum_{k=1}^{3} \lambda \epsilon_{kk} \delta_{ij} + 2\mu\epsilon_{ij}$$

$$\eta = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} \sigma_{ij}^{(\epsilon)} \epsilon_{ij}$$

Node Forces

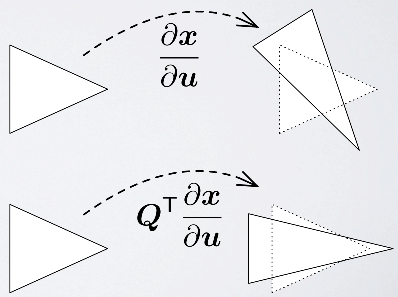
- Combine derivative formula w/ equations for elastic energy
- Integrate over volume of element
- Take derivative w.r.t. node positions

$$\boldsymbol{f}_{[i]}^{(\epsilon)} = -\frac{\text{vol}}{2}\sum_{j=1}^{4} \boldsymbol{p}_{[j]} \sum_{k=1}^{3}\sum_{l=1}^{3} \beta_{jl}\beta_{ik}\sigma_{kl}^{(\epsilon)}$$

Corotational Method

- Factor out rotation using polar decomposition
  - Cauchy strain without errors due to rotations

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} \rightarrow \boldsymbol{Q}\boldsymbol{F}$$



See paper by
Müller & Gross, 2004

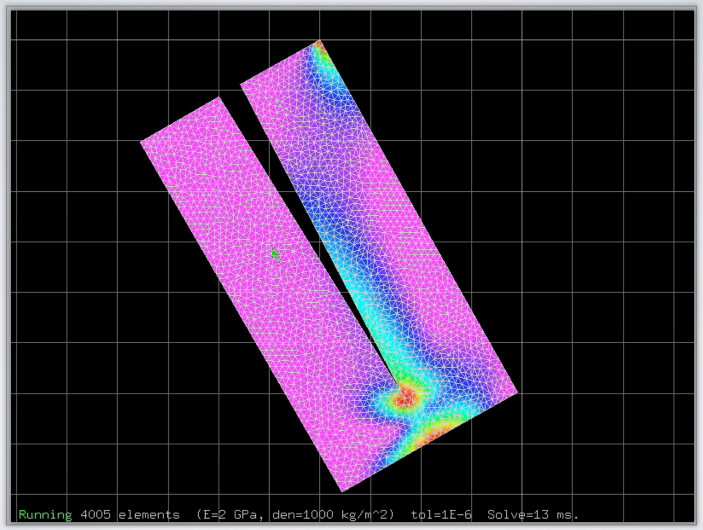
Node Forces and Jacobian

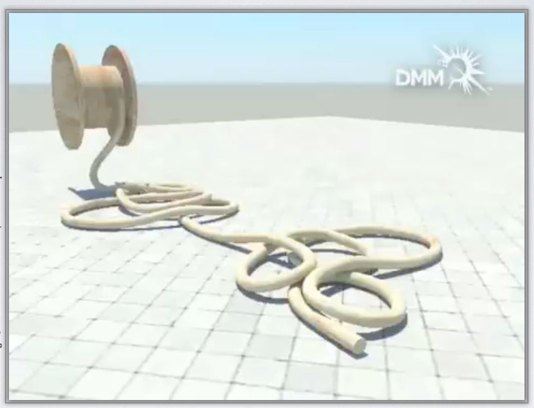
- Combine derivative formula w/ equations for elastic energy
- Integrate over volume of element
- Take derivative w.r.t. node positions

- Jacobian core is constant
  - 12 x 12 made from little 3 x 3 blocks $\boldsymbol{J}_{[i][j]}$

$$\boldsymbol{f}_{[i]} = \boldsymbol{Q}\,\boldsymbol{\sigma}\,\boldsymbol{n}_{[i]}$$

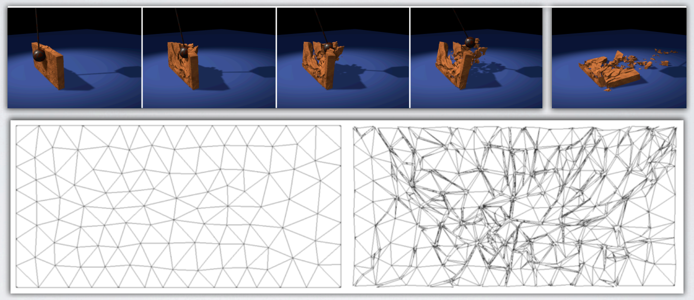
$$\boldsymbol{J}_{[i][j]} = -\boldsymbol{Q}(\lambda \boldsymbol{n}_{[i]}\boldsymbol{n}_{[j]}^{\mathsf{T}} + \mu(\boldsymbol{n}_{[i]} \cdot \boldsymbol{n}_{[j]})\boldsymbol{I} + \mu \boldsymbol{n}_{[j]}\boldsymbol{n}_{[i]}^{\mathsf{T}})\boldsymbol{Q}^{\mathsf{T}}$$


Running 4005 elements (E=2 GPa, den=1000 kg/m^2) tol=1E-6 Solve=13 ms.

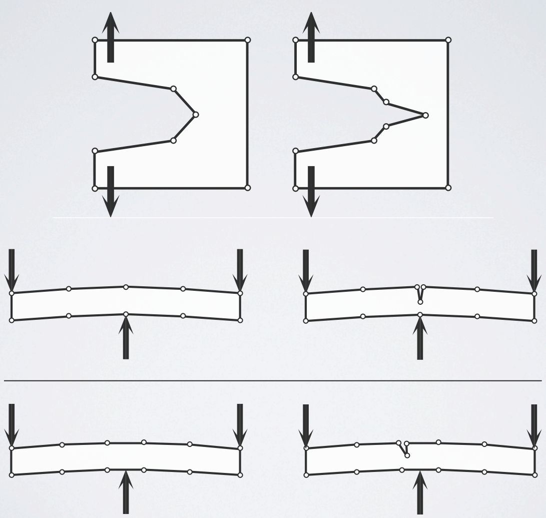

DMM

Fracture

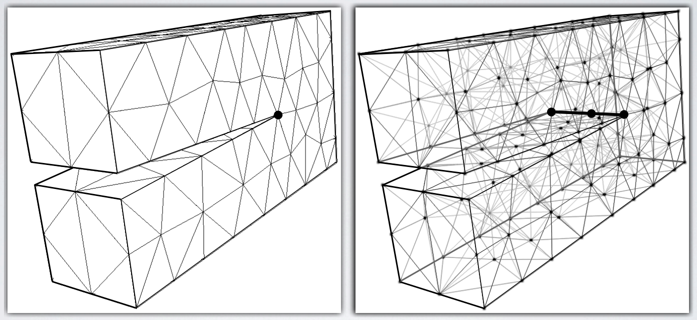
- Fracture changes the mesh



Fracture



Fracture



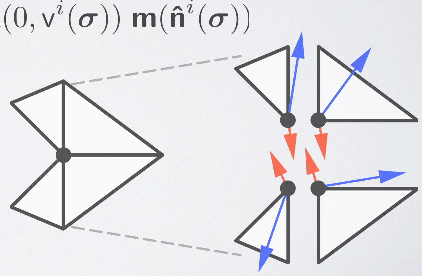
Force Decomposition

- Separate tensile / compressive forces

$$\boldsymbol{\sigma}^+ = \sum_{i=1}^{3} \max(0, \mathrm{v}^i(\boldsymbol{\sigma}))\ \mathbf{m}(\hat{\mathbf{n}}^i(\boldsymbol{\sigma}))$$

$$\boldsymbol{\sigma}^- = \sum_{i=1}^{3} \min(0, \mathrm{v}^i(\boldsymbol{\sigma}))\ \mathbf{m}(\hat{\mathbf{n}}^i(\boldsymbol{\sigma}))$$

$$\mathbf{m}(\boldsymbol{a}) = \begin{cases} \boldsymbol{a}\,\boldsymbol{a}^{\mathsf{T}}/|\boldsymbol{a}| & : \ \boldsymbol{a} \neq \mathbf{0} \\ \mathbf{0} & : \ \boldsymbol{a} = \mathbf{0} \end{cases}$$



Separation

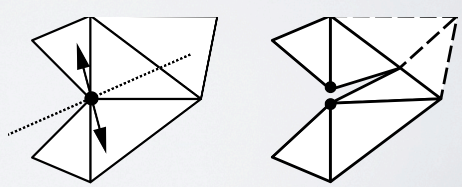
- Build psuedo-stress at each vertex

$$\varsigma = \frac{1}{2}\left(-\mathbf{m}(\boldsymbol{f}^{+}) + \sum_{\boldsymbol{f}\in\{\boldsymbol{f}^{+}\}}\mathbf{m}(\boldsymbol{f}) + \mathbf{m}(\boldsymbol{f}^{-}) - \sum_{\boldsymbol{f}\in\{\boldsymbol{f}^{-}\}}\mathbf{m}(\boldsymbol{f})\right) .$$

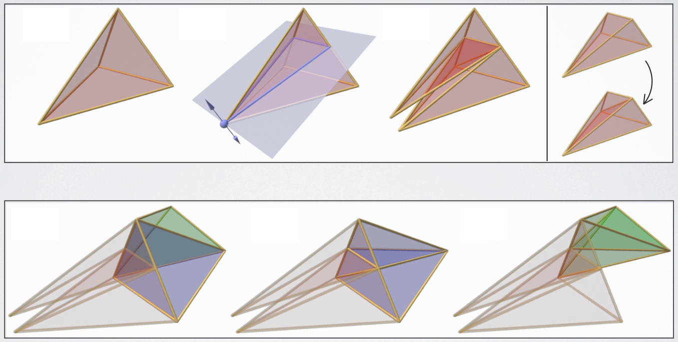
- Eigen decomposition describes how material is being "pulled apart" at each vertex.
- If positive eigenvector over threshold → fracture

Remeshing

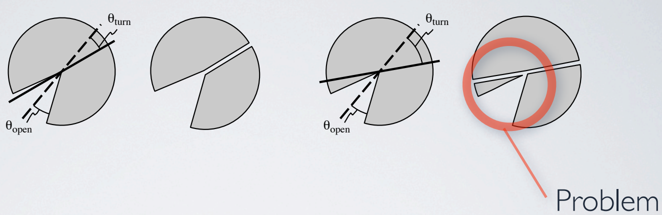
- Remeshing:
  - Fracture plane is normal to max eigenvector
  - Duplicate vertex
  - Split surrounding tetrahedra
    (Easily implemented as edge-splits)



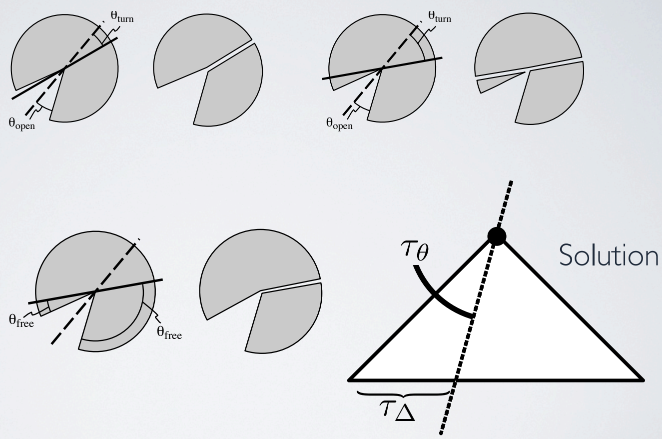
Remeshing



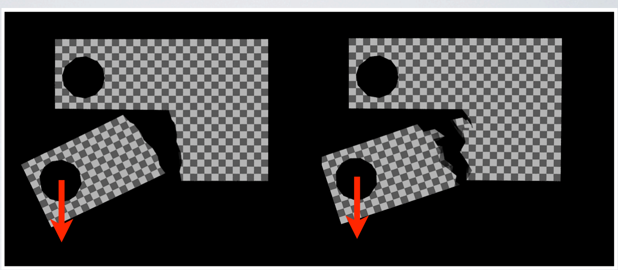
Some Tricks: Back-Cracks



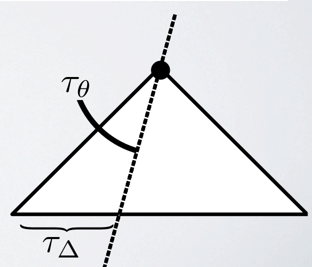
Some Tricks: Back-Cracks



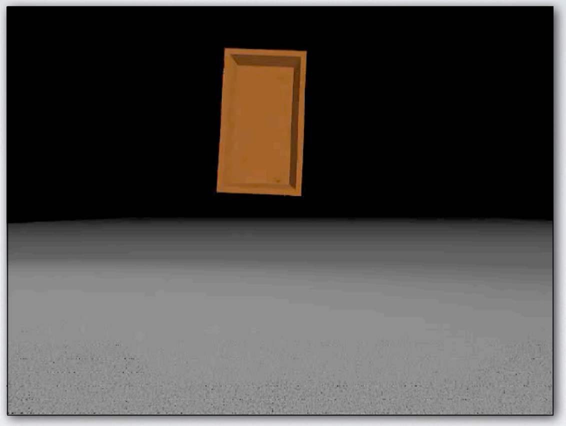
Without Splitting



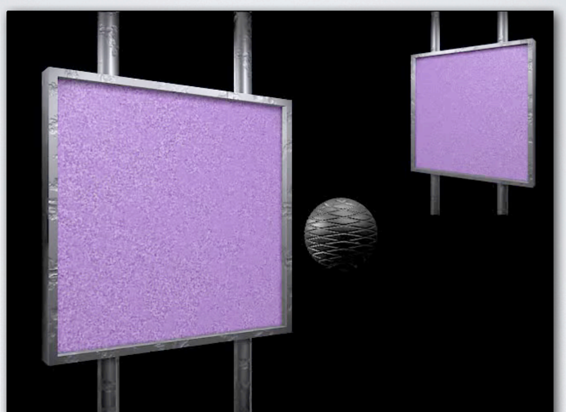
Set thresholds infinite to suppress splitting



Example



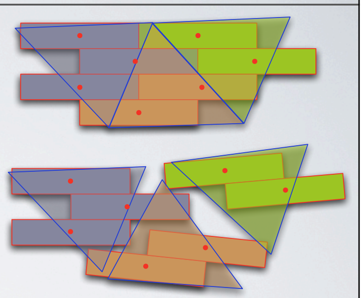
Example



Remeshing

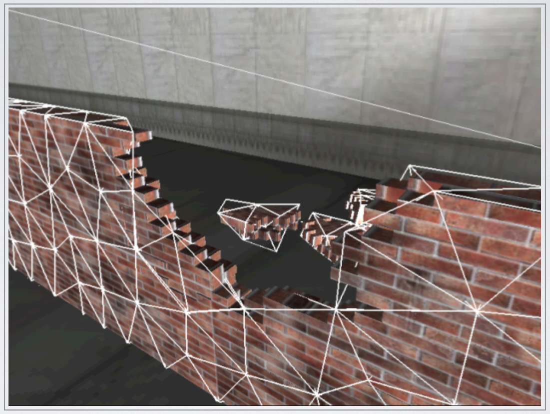
- Remeshing:
  - Fracture plane is normal to max eigenvector
  - Duplicate vertex
  - ~~Split surrounding tetrahedra~~
    ~~(Easily implemented as edge splits)~~

Splinters



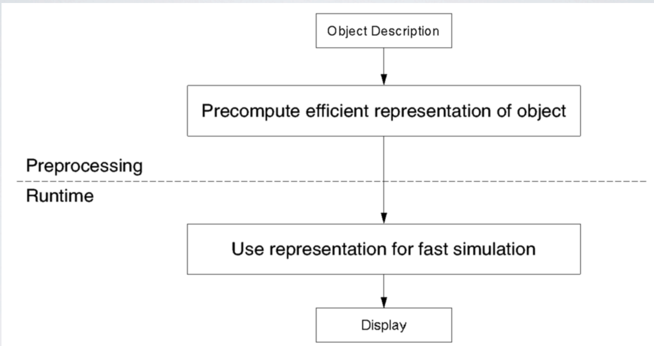
- Splinters are small pieces of geometry attached to a parent element
- The splinter may stick outside the element
- Splinters that cross a face are turned on when the face fractures
- Edge masking, not pre-scoring
- Artistic control

Splinters

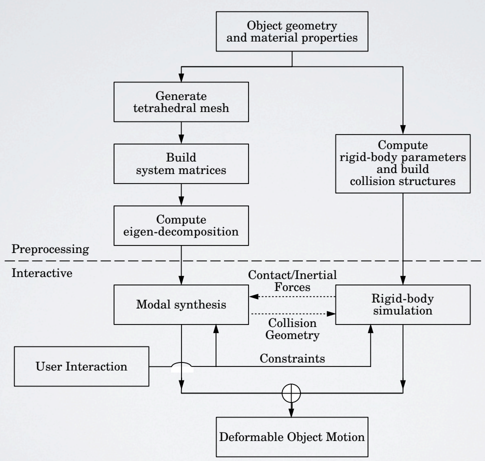


Video footage © LucasArts and Pixelux Entertainment, used with permission.

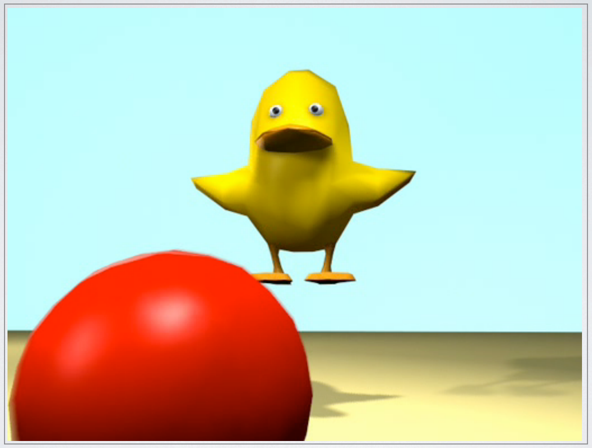
Modal Analysis



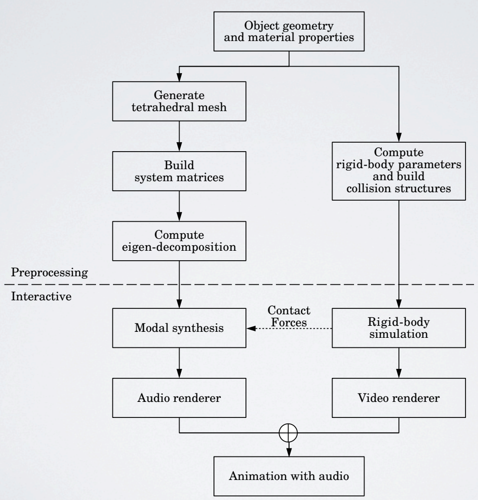
Deformation



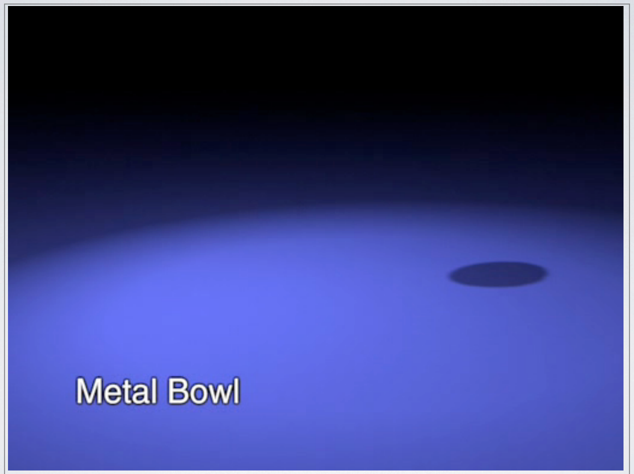
Example



Sound



Sound Example



Modal Decomposition

- Linearize non-linear system

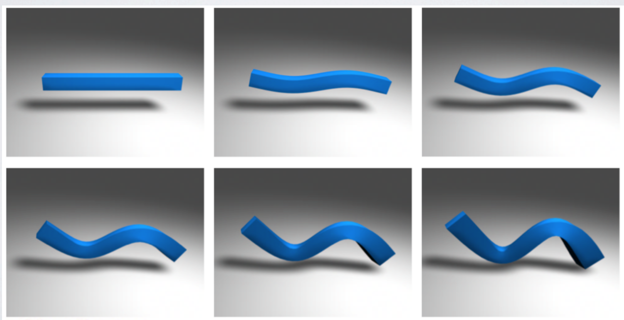
$$\boldsymbol{\mathcal{K}}(\boldsymbol{d}) + \boldsymbol{\mathcal{C}}(\boldsymbol{d}, \dot{\boldsymbol{d}}) + \boldsymbol{\mathcal{M}}(\ddot{\boldsymbol{d}}) = \boldsymbol{f}$$

$$\Downarrow$$

$$\boldsymbol{K}\boldsymbol{d} + \boldsymbol{C}\dot{\boldsymbol{d}} + \boldsymbol{M}\ddot{\boldsymbol{d}} = \boldsymbol{f}$$

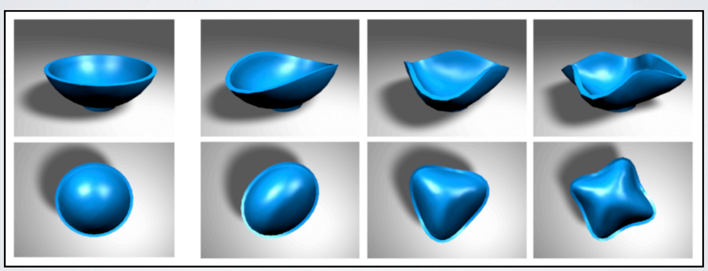
Modal Decomposition

- Consequences of linearization
  - No local rotations



Modal Decomposition

$$\boldsymbol{\Lambda} \boldsymbol{z} + (\alpha_1 \boldsymbol{\Lambda} + \alpha_2 \boldsymbol{I}) \dot{\boldsymbol{z}} + \ddot{\boldsymbol{z}} = \boldsymbol{g}$$



Linearization

$$\mathcal{K}(\boldsymbol{d}) + \mathcal{C}(\dot{\boldsymbol{d}}) + \mathcal{M}(\ddot{\boldsymbol{d}}) = \boldsymbol{f}$$

$$\boldsymbol{K}\boldsymbol{d} + \boldsymbol{C}\dot{\boldsymbol{d}} + \boldsymbol{M}\ddot{\boldsymbol{d}} = \boldsymbol{f}$$

$$\boldsymbol{K}(\boldsymbol{d} + \alpha_1\dot{\boldsymbol{d}}) + \boldsymbol{M}(\alpha_2\dot{\boldsymbol{d}} + \ddot{\boldsymbol{d}}) = \boldsymbol{f}$$

$$\boldsymbol{C} = \alpha_1\boldsymbol{K} + \alpha_2\boldsymbol{M}$$

Normalize for Mass

- Normalize for mass by change of coordinates
  - Cholesky decomposition $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}$
  - Change coordinates $\boldsymbol{y} = \boldsymbol{L}^{\mathsf{T}}\boldsymbol{d}$

$$\boldsymbol{K}(\boldsymbol{d} + \alpha_1\dot{\boldsymbol{d}}) + \boldsymbol{M}(\alpha_2\dot{\boldsymbol{d}} + \ddot{\boldsymbol{d}}) = \boldsymbol{f}$$

$$\Downarrow$$

$$\boldsymbol{L}^{-1}\boldsymbol{K}\boldsymbol{L}^{-\mathsf{T}}(\boldsymbol{y} + \alpha_1\dot{\boldsymbol{y}}) + (\alpha_2\dot{\boldsymbol{y}} + \ddot{\boldsymbol{y}}) = \boldsymbol{L}^{-1}\boldsymbol{f}$$

Diagonalize

- Diagonalize with second change of coordinates
  - Eigen decomposition $\quad \boldsymbol{L}^{-1}\boldsymbol{K}\boldsymbol{L}^{-\mathsf{T}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathsf{T}}$
  - Change coordinates $\quad \boldsymbol{z} = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{y}$

$$\boldsymbol{L}^{-1}\boldsymbol{K}\boldsymbol{L}^{-\mathsf{T}}(\boldsymbol{y} + \alpha_1\dot{\boldsymbol{y}}) + (\alpha_2\dot{\boldsymbol{y}} + \ddot{\boldsymbol{y}}) = \boldsymbol{L}^{-1}\boldsymbol{f}$$

$$\Downarrow$$

$$\boldsymbol{\Lambda}(\boldsymbol{z} + \alpha_1\dot{\boldsymbol{z}}) + (\alpha_2\dot{\boldsymbol{z}} + \ddot{\boldsymbol{z}}) = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{L}^{-1}\boldsymbol{f}$$

$$\Downarrow$$

$$\boldsymbol{\Lambda}\boldsymbol{z} + (\alpha_1\boldsymbol{\Lambda} + \alpha_2\boldsymbol{I})\dot{\boldsymbol{z}} + \ddot{\boldsymbol{z}} = \boldsymbol{g}$$

Diagonalize

$$\boldsymbol{K}(\boldsymbol{d} + \alpha_1\dot{\boldsymbol{d}}) + \boldsymbol{M}(\alpha_2\dot{\boldsymbol{d}} + \ddot{\boldsymbol{d}}) = \boldsymbol{f}$$

Generalized eigenproblem:

$$\boldsymbol{K}\cdot\boldsymbol{w} = \lambda\boldsymbol{M}\cdot\boldsymbol{w}$$

$$\boldsymbol{W} = \boldsymbol{L}^{-\mathsf{T}}\boldsymbol{V}$$

$$\boldsymbol{z} = \boldsymbol{W}^{-1}\cdot\boldsymbol{d} \qquad \boldsymbol{g} = \boldsymbol{W}^{\mathsf{T}}\cdot\boldsymbol{f}$$

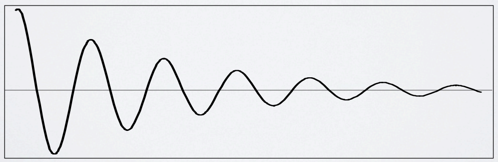
$$\boldsymbol{\Lambda}(\boldsymbol{z} + \alpha_1\dot{\boldsymbol{z}}) + (\alpha_2\dot{\boldsymbol{z}} + \ddot{\boldsymbol{z}}) = \boldsymbol{g}$$

Individual Modes

$$\lambda_i z_i + (\alpha_1 \lambda_i + \alpha_2)\dot{z}_i + \ddot{z}_i = g_i$$

$$z_i = c_1 e^{t\omega_i^+} + c_2 e^{t\omega_i^-}$$

$$\omega_i^\pm = \frac{-(\alpha_1\lambda_i + \alpha_2) \pm \sqrt{(\alpha_1\lambda_i + \alpha_2)^2 - 4\lambda_i}}{2}$$



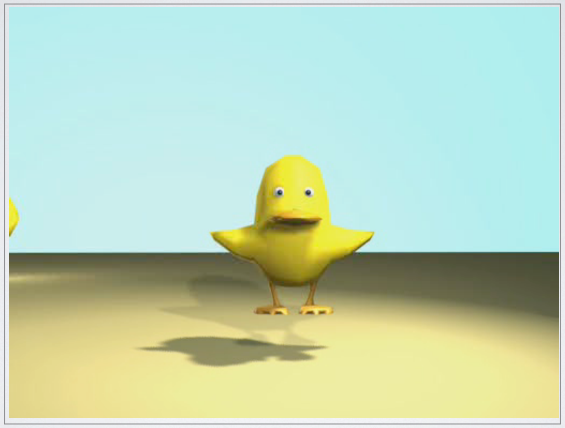
Fast Computation

- Only a pair of complex multiplies per time step

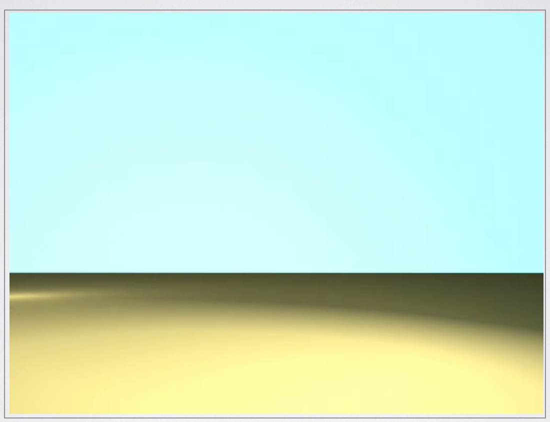
$$e^{\omega(t+\Delta t)} = e^{\omega(t)} e^{\omega(\Delta t)}$$

- No stability limit on step size
- Jump to arbitrary point in time
- Only keep **useful** modes

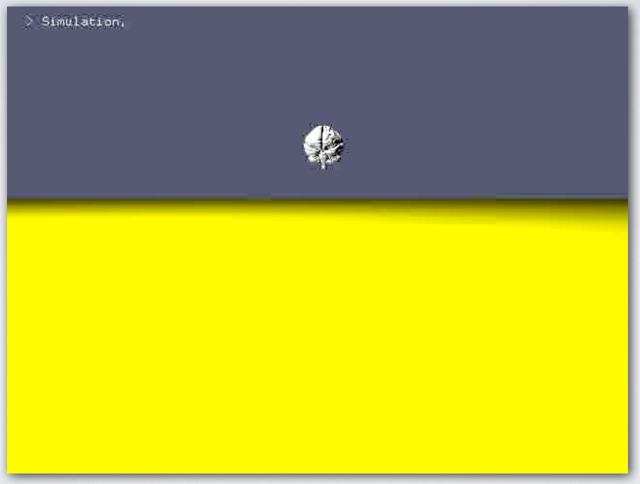
Examples



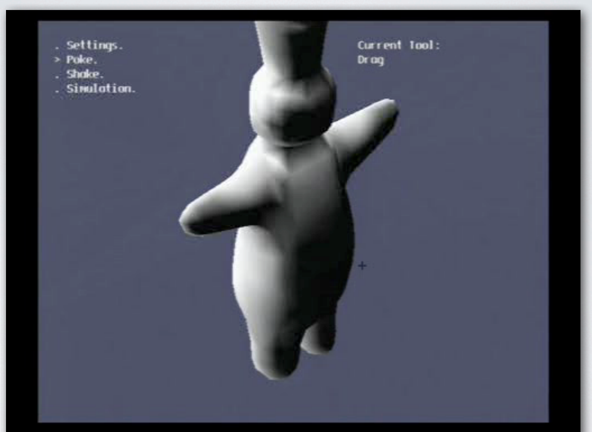
Examples



PS2 Example



PS2 Example



Sound Examples