Notes

- Today 4pm, Dempster 310
  Demetri Terzopoulos is talking
- Please read Pentland and Williams, “Good vibrations”, SIGGRAPH’89

Simplifications of Elasticity

Rotated Linear Elements

- Green strain is quadratic - not so nice
- Cauchy strain can’t handle big rotations
- So instead, for each element factor deformation gradient A into a rotation Q times a deformation F: A=QF
  - Polar Decomposition
- Strain is now just F-I, compute stress, rotate forces back with Q^T
- See Mueller et al, “Interactive Virtual Materials”, GI’04
- Quick and dirty version: use QR, F=symmetric part of R

Inverted Elements

- Too much external force will crush a mesh, cause elements to invert
- Usual definitions of strain can’t handle this
- Instead can take SVD of A, flip smallest singular value if we have reflection
  - Strain is just diagonal now
- See Irving et al., “Invertible FEM”, SCA’04

Embedded Geometry

- Common technique: simulation geometry isn't as detailed as rendered geometry
  - E.g. simulate cloth with a coarse mesh, but render smooth splines from it
- Can take this further: embedded geometry
  - Simulate deformable object dynamics with simple coarse mesh
  - Embed more detailed geometry inside the mesh for collision processing
  - Fast, looks good, avoids the need for complex (and finnicky) mesh generation
  - See e.g. "Skeletal Animation of Deformable Characters," Popovic et al., SIGGRAPH'02

Quasi-Static Motion

- Assume inertia is unimportant---given any applied force, deformable object almost instantly comes to rest
- Then we are quasi-static: solve for current position where $F_{internal}+F_{external}=0$
- For linear elasticity, this is just a linear system
  - Potentially very fast, no need for time stepping etc.
  - Schur complement technique: assume external forces never applied to interior nodes, then can eliminate them from the equation…
    Just left with a small system of equations for surface nodes (i.e. just the ones we actually can see)

Boundary Element Method

- For quasi-static linear elasticity and a homogeneous material, can set up PDE to eliminate interior unknowns---before discretization
  - Very accurate and efficient!
  - Essentially the limit of the Schur complement approach…
- See James & Pai, "ArtDefo…", SIGGRAPH'99
  - For interactive rates, can actually do more: preinvert BEM stiffness matrix
  - Need to be smart about updating inverse when boundary conditions change…

Modal Dynamics

- See Pentland and Williams, "Good Vibrations", SIGGRAPH'89
- Again assume linear elasticity
- Equation of motion is $Ma+Dv+Kx=F_{external}$
- M, K, and D are constant matrices
  - M is the mass matrix (often diagonal)
  - K is the stiffness matrix
  - D is the damping matrix: assume a multiple of K
- This a large system of coupled ODE's now
- We can solve eigen problem to diagonalize and decouple into scalar ODE's
  - M and K are symmetric, so no problems here - complete orthogonal basis of real eigenvectors

Eigenstuff

- Say $U=(u_1 \mid u_2 \mid \dots \mid u_{3n})$ is a matrix with the columns the eigenvectors of $M^{-1}K$ (also $M^{-1}D$)
  - $M^{-1}Ku_i=\lambda_i u_i$ and $M^{-1}Du_i=\mu_i u_i$
  - Assume λ_i are increasing
  - We know $\lambda_1=\dots=\lambda_6=0$ and $\mu_1=\dots=\mu_6=0$ (with $u_1, \dots, u_6$ the rigid body modes)
  - The rest are the deformation modes: the larger that λ_i is, the smaller scale the mode is
- Change equation of motion to this basis…

Decoupling into modes

- Take $y=U^Tx$ (so $x=Uy$) - decompose positions (and velocities, accelerations) into a sum of modes

$$U\ddot{y} = -M^{-1}KUy - M^{-1}DU\dot{y} + M^{-1}F_{ext}$$

- Multiply by U^T to decompose equations into modal components:

$$U^T U\ddot{y} = -U^T M^{-1}KUy - U^T M^{-1}DU\dot{y} + U^T M^{-1}F_{ext}$$

$$\ddot{y} = -diag(\lambda_i)y - diag(\mu_i)\dot{y} + U^T M^{-1}F_{ext}$$

- So now we have 3n independent ODE's
  - If F_{ext} is constant over the time step, can even write down exact formula for each

Examining modes

- Mode i:

$$\ddot{y}_i = -\lambda_i y_i - \mu_i \dot{y}_i + u_i \cdot M^{-1}F_{ext}$$

- Rigid body modes have zero eigenvalues, so just depend on force
  - Roughly speaking, rigid translations will take average of force, rigid rotations will take cross-product of force with positions (torque)
  - Better to handle these as rigid body…
- The large eigenvalues (large i) have small length scale, oscillate (or damp) very fast
  - Visually irrelevant
- Left with small eigenvalues being important

Throw out high frequencies

- Only track a few low-frequency modes (5-10)
- Time integration is blazingly fast!
- Essentially reduced the degrees of freedom from thousands or millions down to 10 or so
  - But keeping full geometry, just like embedded element approach
- Collision impulses need to be decomposed into modes just like external forces

Simplifying eigenproblem

- Low frequency modes not affected much by high frequency geometry
  - And visually, difficult for observers to quantify if a mode is actually accurate
- So we can use a very coarse mesh to get the modes, or even analytic solutions for a block of comparable mass distribution
- Or use a Rayleigh-Ritz approximation to the eigensystem (eigen-version of Galerkin FEM)
  - E.g. assume low frequency modes are made up of affine and quadratic deformations
  - [Do FEM, get eigenvectors to combine them]

More savings

- External forces (other than gravity, which is in the rigid body modes) rarely applied to interior, and we rarely see the interior deformation
- So just compute and store the boundary particles
  - E.g. see James and Pai, “DyRT…”, SIGGRAPH’02 -- did this in graphics hardware!

Inelasticity: Plasticity & Fracture

Plasticity & Fracture

- If material deforms too much, becomes permanently deformed: plasticity
  - Yield condition: when permanent deformation starts happening (“if stress is large enough”)
  - Elastic strain: deformation that can disappear in the absence of applied force
  - Plastic strain: permanent deformation accumulated since initial state
  - Total strain: total deformation since initial state
  - Plastic flow: when yield condition is met, how elastic strain is converted into plastic strain
- Fracture: if material deforms too much, breaks
  - Fracture condition: “if stress is large enough”

For springs (1D)

- Go back to Terzopoulos and Fleischer
- Plasticity: change the rest length if the stress (tension) is too high
  - Maybe different yielding for compression and tension
  - Work hardening: make the yield condition more stringent as material plastically flows
  - Creep: let rest length settle towards current length at a given rate
- Fracture: break the spring if the stress is too high
  - Without plasticity: "brittle"
  - With plasticity first: "ductile"

Fracturing meshes (1D)

- Breaking springs leads to volume loss: material disappears
- Solutions:
  - Break at the nodes instead (look at average tension around a node instead of on a spring)
    - Note: recompute mass of copied node
  - Cut the spring in half, insert new nodes
    - Note: could cause CFL problems…
  - Virtual node algorithm
    - Embed fractured geometry, copy the spring (see Molino et al. "A Virtual Node Algorithm…" SIGGRAPH'04)

Multi-Dimensional Plasticity

- Simplest model: total strain is sum of elastic and plastic parts: $\varepsilon=\varepsilon_e+\varepsilon_p$
- Stress only depends on elastic part (so rest state includes plastic strain): $\sigma=\sigma(\varepsilon_e)$
- If σ is too big, we yield, and transfer some of ε_e into ε_p so that σ is acceptably small

Multi-Dimensional Yield criteria

- Lots of complicated stuff happens when materials yield
  - Metals: dislocations moving around
  - Polymers: molecules sliding against each other
  - Etc.
- Difficult to characterize exactly when plasticity (yielding) starts
  - Work hardening etc. mean it changes all the time too
- Approximations needed
  - Big two: Tresca and Von Mises

Yielding

- First note that shear stress is the important quantity
  - Materials (almost) never can permanently change their volume
  - Plasticity should ignore volume-changing stress
- So make sure that if we add kI to σ it doesn't change yield condition

Tresca yield criterion

- This is the simplest description:
  - Change basis to diagonalize σ
  - Look at normal stresses (i.e. the eigenvalues of σ)
  - No yield if $\sigma_{max}-\sigma_{min} \le \sigma_Y$
- Tends to be conservative (rarely predicts yielding when it shouldn't happen)
- But, not so accurate for some stress states
  - Doesn't depend on middle normal stress at all
- Big problem (mathematically): not smooth

Von Mises yield criterion

- If the stress has been diagonalized:

$$\frac{1}{\sqrt{2}}\sqrt{(\sigma_1-\sigma_2)^2+(\sigma_2-\sigma_3)^2+(\sigma_3-\sigma_1)^2} \le \sigma_Y$$

- More generally: $\sqrt{\tfrac{3}{2}}\sqrt{\|\sigma\|_F^2-\tfrac{1}{3}Tr(\sigma)^2} \le \sigma_Y$
- This is the same thing as the Frobenius norm of the deviatoric part of stress
  - i.e. after subtracting off volume-changing part:

$$\sqrt{\tfrac{3}{2}}\left\|\sigma-\tfrac{1}{3}Tr(\sigma)I\right\|_F \le \sigma_Y$$

Linear elasticity shortcut

- For linear (and isotropic) elasticity, apart from the volume-changing part which we cancel off, stress is just a scalar multiple of strain
  - (ignoring damping)
- So can evaluate von Mises with elastic strain tensor too (and an appropriately scaled yield strain)

Perfect plastic flow

- Once yield condition says so, need to start changing plastic strain
- The magnitude of the change of plastic strain should be such that we stay on the yield surface
  - I.e. maintain f(σ)=0 (where f(σ)≤0 is, say, the von Mises condition)
- The direction that plastic strain changes isn't as straightforward
- "Associative" plasticity: $$\dot{\varepsilon}_p = \gamma \frac{\partial f}{\partial \sigma}$$

Algorithm

- After a time step, check von Mises criterion: is $f(\sigma) = \sqrt{\tfrac{3}{2}}\left\|dev(\sigma)\right\|_F - \sigma_Y > 0$?
- If so, need to update plastic strain:

$$\varepsilon_p^{new} = \varepsilon_p + \gamma \frac{\partial f}{\partial \sigma}$$
$$= \varepsilon_p + \gamma \sqrt{\tfrac{3}{2}} \frac{dev(\sigma)}{\left\|dev(\sigma)\right\|_F}$$

  - with γ chosen so that $f(\sigma^{new})=0$ (easy for linear elasticity)

Multi-Dimensional Fracture

- Smooth stress to avoid artifacts (average with neighbouring elements)
- Look at largest eigenvalue of stress in each element
- If larger than threshhold, introduce crack perpendicular to eigenvector
- Big question: what to do with the mesh?
  - Simplest: just separate along closest mesh face
  - Or split elements up: O'Brien and Hodgins
  - Or model crack path with embedded geometry: Molino et al.