Notes

- Read "Physically Based Modelling" SIGGRAPH course notes by Witkin and Baraff (at least, rigid body sections)
  - An alternative way to derive the equations of motion for rigid bodies

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 $f(\sigma)=0$ (where $f(\sigma)\le 0$ 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 $f(\sigma) = \sqrt{\tfrac{3}{2}}\|dev(\sigma)\|_F - \sigma_Y > 0$?
- If so, need to update plastic strain:
$$\begin{aligned}\varepsilon_p^{new} &= \varepsilon_p + \gamma\frac{\partial f}{\partial \sigma}\\ &= \varepsilon_p + \gamma\sqrt{\tfrac{3}{2}}\frac{dev(\sigma)}{\|dev(\sigma)\|_F}\end{aligned}$$
  - 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.

Rigid Bodies

Rigid Bodies

- Most volumes in the real world are very stiff---not visibly deformable
- Rather than stiff and expensive deformable mechanics, mathematically abstract this into perfectly rigid bodies
  - Constrain motion to rigid body modes
  - Avoid having to model internal “constraint” forces which keep bodies rigid
- More efficient, but rigid abstraction can cause problems…
- Still, best approach especially for real-time simulations of such objects - e.g. games
  - Even large objects which deform may be best decomposed into rigid parts

Object Space vs. World Space

- As before, we have rest/reference/object space configuration (label points with variable p)
- And current/real/world space configuration (position is x(p))
- First note:
  - If it ever gets confusing, replace continuous matter with a finite set of mass points (object space positions p_1, p_2, … and world space positions x_1, x_2, …)
- Rigidity means that x(p,t)=R(t)p+X(t)
  - R is a rotation matrix (orthogonal and det(R)=1)
  - X is a translation vector

Kinematics

- Differentiate map in time: $v = \dot{R}p + V$
- Invert map for p: $p = R^T(x - X)$
- Thus: $v = \dot{R}R^T(x - X) + V$
- 1st term: rotation, 2nd term: translation
  - Let’s simplify the rotation

Skew-Symmetry

- Differentiate $RR^T=\delta$ w.r.t. time:

$$\dot{R}R^T + R\dot{R}^T = 0 \;\Rightarrow\; \dot{R}R^T = -\left(\dot{R}R^T\right)^T$$

- Skew-symmetric! Thus can write as:

$$\dot{R}R^T = \begin{pmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{pmatrix}$$

- Call this matrix ω^* (built from a vector ω)

$$\dot{R}R^T = \omega^* \quad\Rightarrow\quad \dot{R} = \omega^* R$$

The cross-product matrix

- Note that:

$$\omega^* x = \begin{pmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{pmatrix}\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \omega_1 x_2 - \omega_2 x_1 \\ \omega_2 x_0 - \omega_0 x_2 \\ \omega_0 x_1 - \omega_1 x_0 \end{pmatrix} = \omega \times x$$

- So we have:

$$v = \omega \times \left(x - X\right) + V$$

- ω is the angular velocity of the object
- Magnitude gives speed of rotation, direction gives axis of rotation

Velocity Modes

- Think of linear space of all possible velocities, and choose a set of basis vectors for the subspace of allowed motions (rigid body motions)
  - Think back to modal dynamics…
- In this case, velocity is a linear combination of 3 translations and 3 rotations, with coefficients V and ω
- Write this as v=Su, or

$$v(x) = S(x)u$$
$$= \begin{pmatrix}\delta & X^* - x^*\end{pmatrix}\begin{pmatrix}V \\ \omega\end{pmatrix} \qquad \left[= V - (x - X)\times\omega\ldots\right]$$

Virtual Work

- The internal "constraint" forces are going to keep v in the span of S, so v=Su for some coefficients u
- But assume (and this is the key assumption) that they don't mess with these allowed modes
- That is, they are orthogonal:
  - $S^T F_{int}=0$
  - They do no "virtual work"
- For example, internal forces won't cause an object to out of the blue start translating, or rotating…
  - Can derive from, for example, the assumption that down at some level forces between particles are in the direction between particles

Constrained Dynamics

- We have F=Ma, i.e. $M\dot{v} = F_{\text{int}} + F_{\text{ext}}$
- Now plug in form for constrained velocity

$$MS\dot{u} + M\dot{S}u = F_{\text{int}} + F_{\text{ext}}$$

- And eliminate the internal forces:

$$S^T MS\dot{u} + S^T M\dot{S}u = \underbrace{S^T F_{\text{int}}}_{=0} + S^T F_{\text{ext}}$$

$$\left(S^T MS\right)\dot{u} + \left(S^T M\dot{S}\right)u = S^T F_{\text{ext}}$$

Working it out

- What is the matrix multiplying du/dt?

$$S^T MS = \sum_i \begin{pmatrix}\delta \\ X^{*T} - x_i^{*T}\end{pmatrix} M_i \begin{pmatrix}\delta & X^* - x_i^*\end{pmatrix}$$
$$= \sum_i m_i \begin{pmatrix}\delta & X^* - x_i^* \\ X^{*T} - x_i^{*T} & \left(X^* - x_i^*\right)^T\left(X^* - x_i^*\right)\end{pmatrix}$$

- Using total mass M (not the matrix!) and centre of mass X_{CM} this simplifies to:

$$\begin{pmatrix}M\delta & MX^* - MX_{CM}^* \\ MX^{*T} - MX_{CM}^{*T} & \sum_i m_i\left(X - x_i\right)^{*T}\left(X - x_i\right)^*\end{pmatrix}$$

Can we do better?

- We have some freedom in defining S
- Don't have to rotate about the origin: can rotate around the centre of mass instead
  - This will let us zero out the off-diagonal blocks, make it simpler to invert the 6x6 matrix
- That is, $X=X_{CM}$ so that $S^T MS$ becomes

$$\begin{pmatrix}M\delta & 0 \\ 0 & \sum_i m_i\left(X - x_i\right)^{*T}\left(X - x_i\right)^*\end{pmatrix} = \begin{pmatrix}M\delta & 0 \\ 0 & I(t)\end{pmatrix}$$

- We call I(t) the "inertia tensor"

Change of S

- There was also the dS/dt term
- Note that identity matrix is constant: disappears
- We get

$$S^T M\dot{S} = \sum_i m_i \begin{pmatrix}\delta \\ X^{*T} - x_i^{*T}\end{pmatrix}\begin{pmatrix}0 & V^* - v_i^*\end{pmatrix}$$
$$= \sum_i m_i \begin{pmatrix}0 & V^* - v_i^* \\ 0 & \left(X - x_i\right)^{*T}\left(V - v_i\right)^*\end{pmatrix}$$
$$= \begin{pmatrix}0 & 0 \\ 0 & \sum_i m_i\left(X - x_i\right)^{*T}\left(V - v_i\right)^*\end{pmatrix}$$

- Note simplification from $m_i v_i$ summing to MV
- Left with dI(t)/dt in lower right corner

What about the external forces?

- What is $S^T F_{ext}$?

$$S^T F_{\text{ext}} = \sum_i \begin{pmatrix} \delta \\ X^{*T} - x_i^{*T} \end{pmatrix} f_i = \begin{pmatrix} \sum_i f_i \\ \sum_i (x_i - X) \times f_i \end{pmatrix} = \begin{pmatrix} F \\ \tau \end{pmatrix}$$

- We call F the net force, and τ the net torque

Equations of Motion

- Plugging it all in (and assuming centre of mass is at the origin in object space)

$$M\dot{V} = F$$

$$\frac{d}{dt}\big(I(t)\omega\big) = \boldsymbol{\tau}$$

- Call L=**I**(t)ω the angular momentum
  - The component of momentum in the rotational mode
- Also add in

$$\dot{X} = V$$
$$\dot{R} = \omega^* R$$

To Do

- Figure out I(t) efficiently
- Numerically integrate the ODE's
  - Turns out R is not a good representation for the current rotation
- Look at the net force and torque of some external forces
  - Gravity
  - Collisions