1

Notes

- Reference
  - Witkin and Baraff, “Physically Based Modelling” course, SIGGRAPH 2001
  - Link on the course website

2

True Collisions

- Turn attention from repulsions for a while
- Model collision as a discrete event - a bounce
  - Input: incoming velocity, object normal
  - Output: outgoing velocity
- Need some idea of how “elastic” the collision
  - Fully elastic - reflection
  - Fully inelastic - sticks (or slides)
- Let’s ignore friction for now
- Let’s also ignore how to incorporate it into algorithm for moving particles for now

3

Newtonian Collisions

- Say object is stationary, normal at point of impact is n
- Incoming particle velocity is v
- Split v into normal and tangential components:

$$v_N = v \cdot n$$

$$v_T = v - v_N n$$

- Newtonian model for outgoing velocity
  - Unchanged tangential component v_T
  - New normal component is $v_N^{new} = -\varepsilon v_N^{old}$
  - The “coefficient of restitution” is ε, ranging from 0 (inelastic) to 1 (perfectly elastic)
- The final outgoing velocity is

$$v^{new} = v_T^{old} - \varepsilon v_N^{old} n$$

4

Relative velocity in collisions

- What if particle hits a moving object?
- Now process collision in terms of relative velocity
  - $v_{rel} = v_{particle} - v_{object}$
  - Take normal and tangential components of relative velocity
  - Reflect normal part appropriately to get new v_{rel}
  - Then new $v_{particle} = v_{object} + (\text{new } v_{rel})$

5

Movable Objects

- Before assumed $m_{object} >> m_{particle}$
  - Then effect on object is negligible
  - If not, still calculate new v_{rel} as above
  - But change v_{object} and $v_{particle}$ with "impulses"
- Unknown impulse I (force * time) applied to particle and opposite -I to object
- New velocities: $v^{new}_{particle} = v_{particle} + I/m_{particle}$

$$v^{new}_{object} = v_{object} - I/m_{object}$$

- New relative velocity in terms of I gives equation to solve for I:

$$v^{new}_{rel} = v_{rel} + \left(1/m_{particle} + 1/m_{object} \right) I$$

6

Friction

- Friction slows down the relative tangential velocity
- Causes a tangential force F_T that opposes sliding, according to
  - Magnitude of normal force F_N pressing on particle
  - And friction coefficient μ
- Basic Coulomb law:
  - If kinetic friction ($v_{Trel} \neq 0$) then $|F_T| = \mu |F_N|$ and is in a direction most opposing sliding
  - If static friction ($v_{Trel} = 0$) then $|F_T| \leq \mu |F_N|$

7

Implementing Friction

- Gets really messy to directly use friction forces (really hard to get true static friction!)
- Instead integrate into relative velocity update
- Integrating normal and friction force over the collision time and dividing by mass gives Coulomb friction in terms of velocity changes:
  - Static friction: $|\Delta v_T| \leq \mu |\Delta v_N|$ (and then $v_T = 0$)
  - Kinetic friction: $\Delta v_T = -\mu |\Delta v_N| \, v_T / |v_T|$
    - Assuming direction of friction force is always opposing the initial tangential velocity
  - Combine into one formula for new relative tangential velocity:

$$v_T^{after} = \max\left(0, 1 - \frac{\mu |\Delta v_N|}{|v_T^{before}|}\right) v_T^{before}$$

8

Collisions so far

- We now have a black box collision processing routine
  - Input:
    - particle velocity before
    - (maybe object velocity and masses)
    - object normal
    - parameters ε and μ
  - Intermediate:
    - Relative velocity, split into normal and tangential components
  - Output:
    - new particle velocity
    - (maybe new object velocity)
- How do we use this in time integration?

9

Simple collision algorithm

- After each time step, check if particles collided with objects
  - If so, change velocities according to routine
- Fails catastrophically for more interesting cases
  - New velocity may or may not get particle out next time step - is that another collision?
  - Is it ok to have particles inside (or on the wrong side of) objects any time?

10

Backing up time

- Can avoid some problems by processing collision when it happens, not after the fact
- Figure out when collision happens (or at least get close to time of collision, but not later than)
- Apply velocity update then
- Potential problems:
  - Hard to figure out time
  - Could involve a lot of work per time step (unpredictable)

11

Simultaneous collision resolution

- Ignore exact timing and order of collisions during a time step
- Begin with old position x^{old}
- New candidate position x^{new}
- If collision occurred, process with $v^{avg}=(x^{new}-x)/\Delta t$ to get new post-collision velocity v^{after}
- Then change x^{new} to $x^{after}=x^{old}+\Delta t\ v^{after}$
- Iterate until no collisions remain

12

Notes on collision resolution

- This works really well for inelastic collisions
- Can use a large Δt: separate collision processing from particle physics
  - Can take many small steps to from x^{old} to x^{new} if stability demands it
- Problems arise with elastic collisions
  - May not converge
  - Bouncing block problem: a block won't come to rest on the floor

Elastic collision resolution

- Start with x^{old} and v^{old}
- Advance to x^{new}
- If collision, apply elastic collision law to v^{old} to get v^2
- Take $x^2=x^{new}+\Delta t\,(v^2-v^{old})$ or reintegrate from x^{old}, v^2 if you can afford it
  - Repeat elastic step a few times if you want, and there are still collisions with x^2
- If still collision, apply INELASTIC collision law to $v^{avg}=(x^2-x^{old})/\Delta t$ to get v^{after}
- Change x^2 to $x^{after}=x^{old}+\Delta t\, v^{after}$
- Repeat as needed

One last problem

- Due to round-off error, or pathological geometry, may still go into a long loop resolving collisions
- So cut loop off after a small number of iterations
- Failsafe: take $v^{after}=0$, $x^{new}=x^{old}$
  - May look weird, still could have issues for moving objects especially
- Last resort: accept the penetration, apply a repulsion force to eventually move the particle out from the object(s)