PARTICLE SYSTEMS

OUTLINE

- Newtonian Particles
- Meshes
- Efficiency
- Constraints

INTRODUCTION

- Most important of procedural methods
- Used to model
  - Natural phenomena
    - Clouds
    - Terrain
    - Plants
  - Crowd Scenes
  - Real physical processes

NEWTONIAN PARTICLE

- Particle system is a set of particles
- Each particle is an ideal point mass
- Six degrees of freedom
  - Position
  - Velocity
- Each particle obeys Newtons' law

$$f = ma$$

FORCE VECTOR

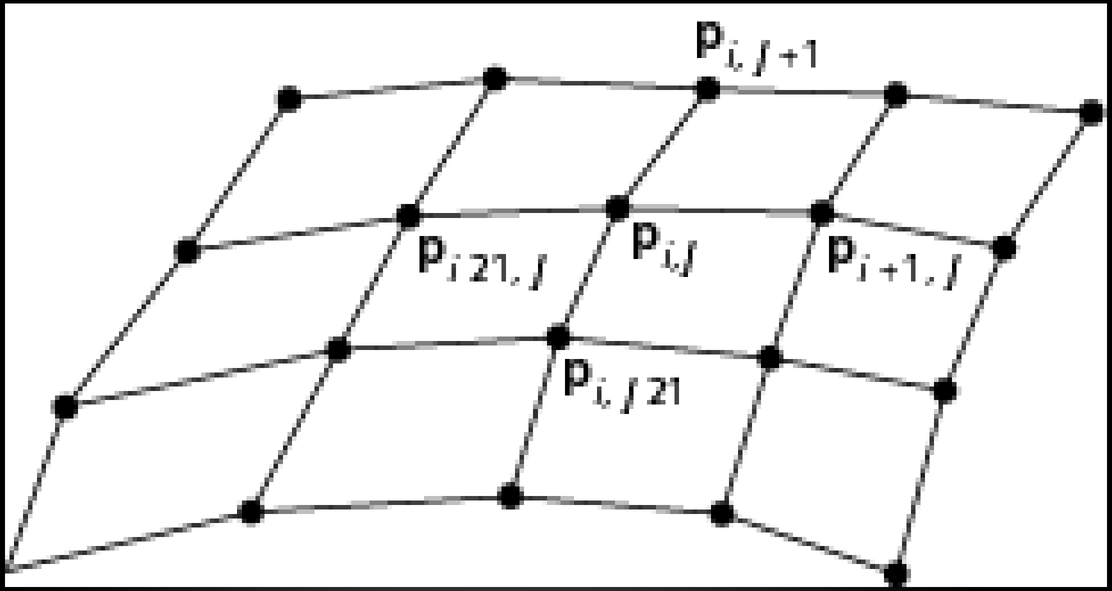
- Independent Particles
  - Gravity
  - Wind forces
  - O(n) calulation
- Coupled Particles O(n)
  - Meshes
  - Spring-Mass Systems
- Coupled Particles O(n^2)
  - Attractive and repulsive forces

SOLUTION OF PARTICLE SYSTEMS

```
float time, delta state[6n], force[3n];
state = initial_state();
for(time = t0; time<final_time, time+=delta) {
   force =  force_function(state, time);
   state = ode(force, state, time, delta);
   render(state, time)
}
```

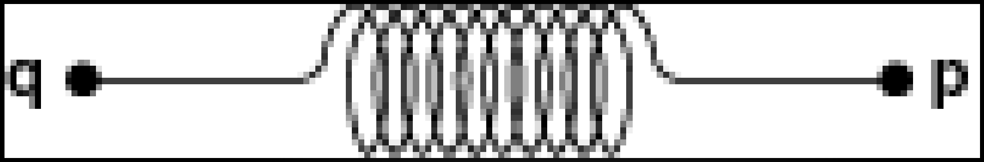
MESHES

- Connect each particle to its closest neighbors
  - O(n) force calculation
- Use spring-mass system



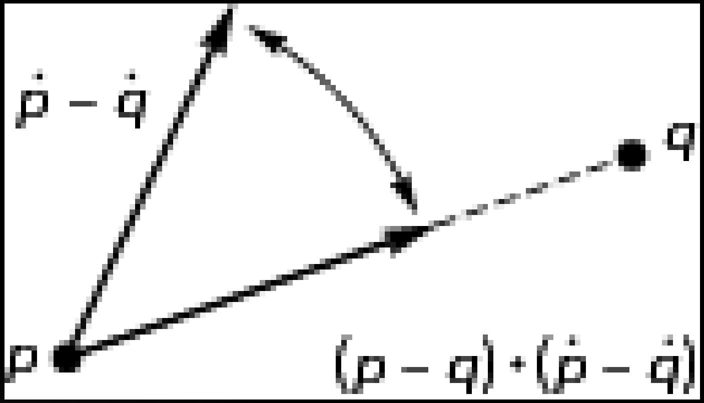
SPRING FORCES

- Assume each particle has unit mass and is connected to its neighbor(s) by a spring
- Hooke's law: force proportional to distance between the points



SPRING DAMPING

- A pure spring-mass will oscillate forever
- Must add a damping term
- Must project velocity
-

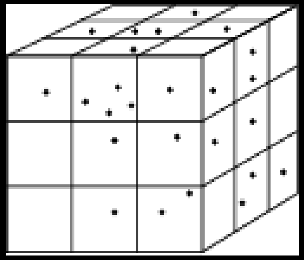


ATTRACTION AND REPULSION

- General case requires $O(n^2)$ calculation
- In most problems, the drop off is such that not many particles contribute to the forces on any given particle
- Sorting problem: is it $O(n \log n)$?

BOXES

- Spatial subdivision technique
- Divide space into boxes
- Particle can only interact with particles in its box or the neighboring boxes
- Must update which box a particle belongs to after each time step



LINKED LISTS

- Each particle maintains a linked list of its neighbors
- Update data structure at each time step
- Must amortize cost of building the data structures initially

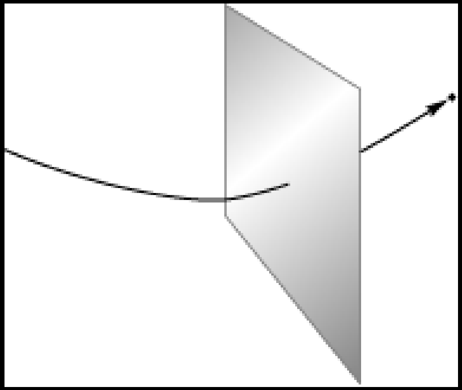
PARTICLE FIELD CALCULATIONS

- Consider simple gravity
- We don’t compute forces due to sun, moon, and other large bodies
- Rather we use the gravitational field
- Usually we can group particles into equivalent point masses

CONSTRAINTS

- Easy in computer graphics to ignore physical reality
- Surfaces are virtual
- Must detect collisions separately if we want exact solution
- Can approximate with

repulsive forces



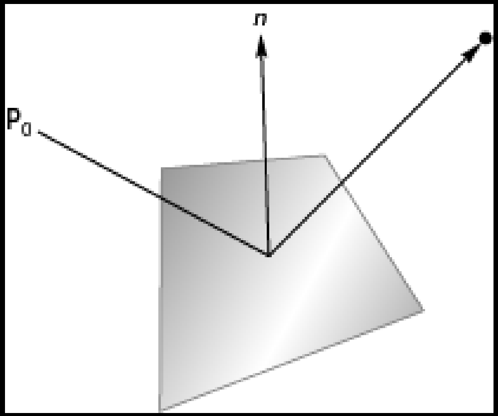
COLLISIONS

Once we detect a collision, we can calculate new path

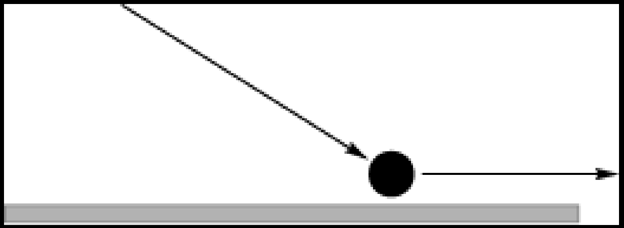
Use coefficient of resititution

Reflect vertical component

May have to use partial time step



CONTACT FORCES



SUMMARY

- Newtonian Particles
- Meshes
- Efficiency
- Constraints