MP4: Smooth I

CS 418 – Interactive Computer Graphics
TA: Gong Chen
Fall 2012

Today's Topics

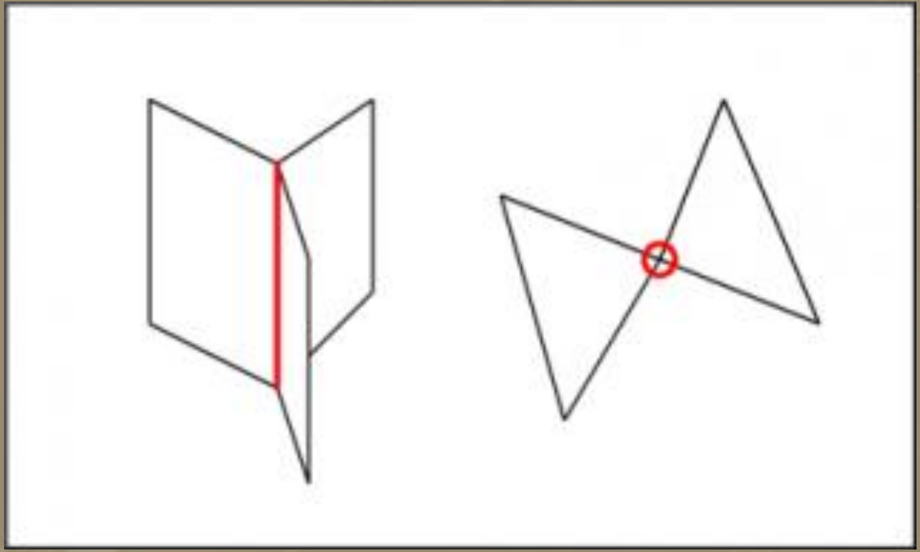
- MP4 Explaination
  - Subdivision 40%
    - Curved camera path 10%
    - Appearance (texture/lighting/color) 10%
  - Compilation 20%
  - Documentation 20%
- Non-manifold mesh
- catmul-clark subdivision coding scheme
- camera transitions
- More about particles

Non-Manifold Mesh

- A "Non-Manifold" mesh is a mesh for which there are edges belonging to *more* than two faces.
- In general a "Non-Manifold" mesh occurs when you have internal faces and the like.

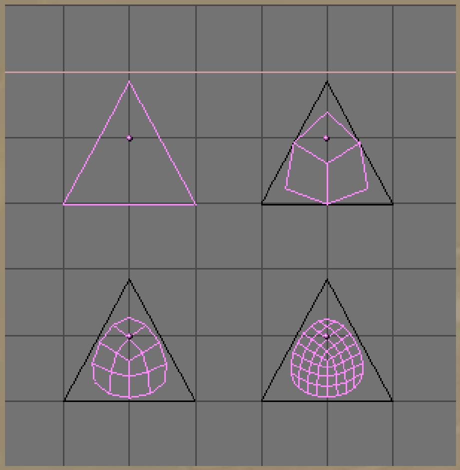
➔ **(make sure you have a manifold mesh)**

- http://www.youtube.com/watch?feature=player_embedded&v=vrqxp89ilM4

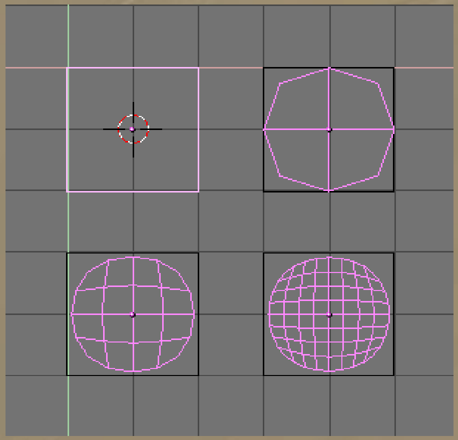


catmul-clark subdivision

- Start with a manifold mesh.
- All the vertices in the mesh are called original points.



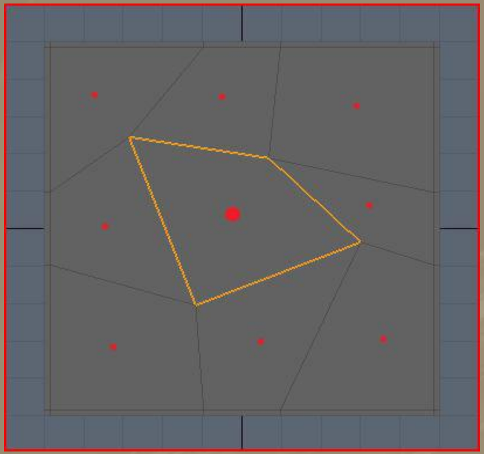
- Loops on:
  1. each face
  2. each edge
  3. each original point *P*



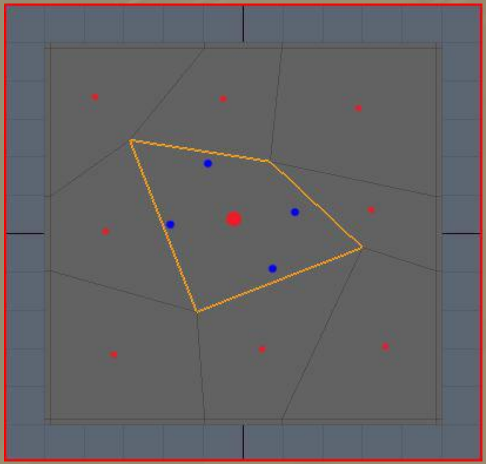
- Connect the new points

catmul-clark subdivision (cont.)

- For each face, add a *face point*
  - Set each face point to be the *centroid of all original points for the respective face*.



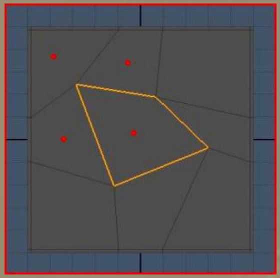
- For each edge, add an *edge point*
  - Set each edge point to be the *average of the two neighbouring face points and its two original endpoints*.



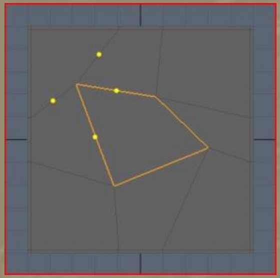
Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

catmul-clark subdivision (cont.)

- For each original point *P:*
  - F = average *F* of all *n* face points for faces touching *P*

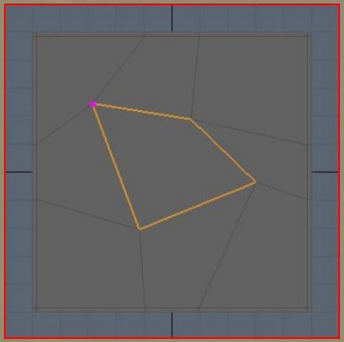


  - R = average *R* of all *n* edge midpoints for edges touching *P*
    - each edge midpoint is the average of its two endpoint vertices.



  - "***Move***" *each original point* to the point (n=4):

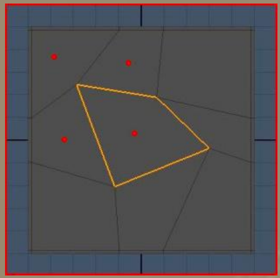
$$\frac{F + 2R + (n-3)P}{n}$$



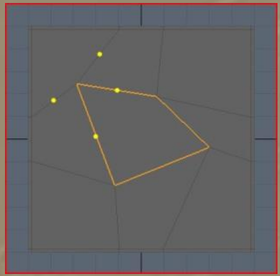
Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

catmul-clark subdivision (cont.)

- For each original point *P:*
  - F = average *F* of all *n* face points for faces touching *P*

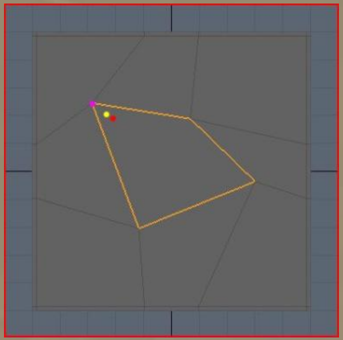


  - R = average *R* of all *n* edge midpoints for edges touching *P*
    - each edge midpoint is the average of its two endpoint vertices.



  - "***Move***" *each original point* to the point (n=4):

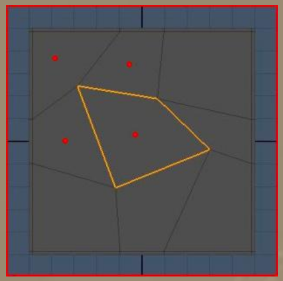
$$\frac{F + 2R + (n-3)P}{n}$$



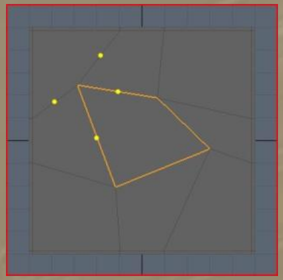
Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

catmul-clark subdivision (cont.)

- For each original point *P:*
  - F = average *F* of all *n* face points for faces touching *P*

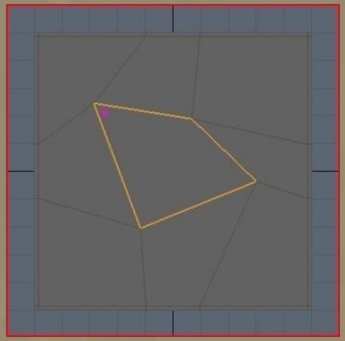


  - R = average *R* of all *n* edge midpoints for edges touching *P*
    - each edge midpoint is the average of its two endpoint vertices.



  - "***Move***" *each original point* to the point (n=4):

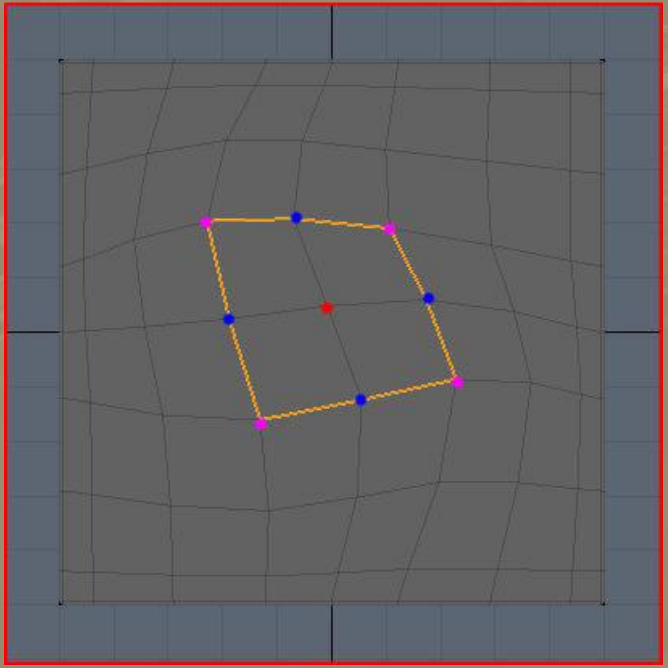
$$\frac{F + 2R + (n-3)P}{n}$$



Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

catmul-clark subdivision (cont.)

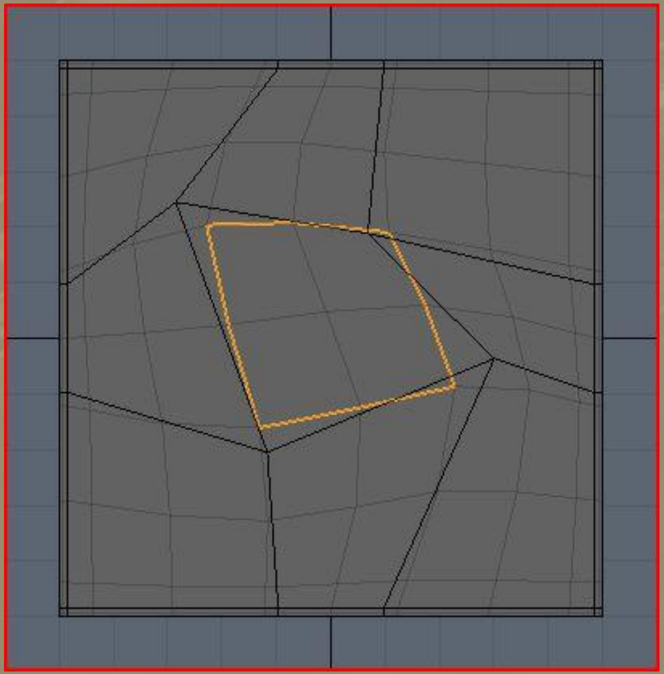
- Connect all points:
  - Blue = (new) edge points
  - Red = (new) face point
  - Pink = (modified) vertex



Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

catmul-clark subdivision (cont.)

- Connect all points:
  - Blue = (new) edge points
  - Red = (new) face point
  - Pink = (modified) vertex



Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

Sharp Edges

1. Tag Edges as "sharp" or "not-sharp"
   - sh = 0 "not sharp"
   - sh > 0 sharp
2. During Subdivision,
   - if an edge is "sharp", use sharp subdivision rules. Newly created edges, are assigned a sharpness of sh-1.
   - If an edge is "not-sharp", use normal smooth subdivision rules.

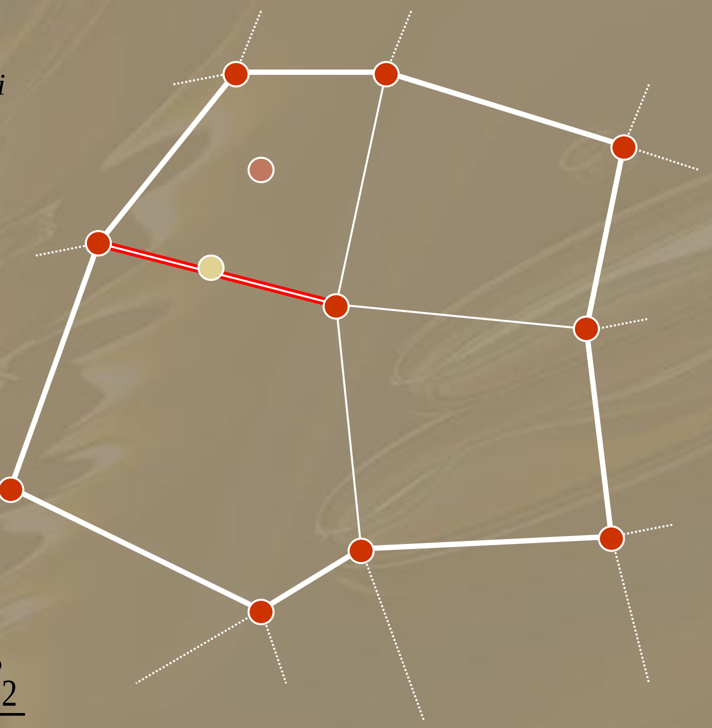
Ref: "Catmull-Clark Subdivision: The Basics", http://www.rorydriscoll.com/

Sharp Rules

- FACE (unchanged) $f = \frac{1}{n}\sum_{1}^{n} v_i$

- EDGE $e = \frac{v_1 + v_2}{2}$

- VERTEX



Dart >2
(One sharp incident edge)

$$v_{i+1} = v_i$$

Crease 2
(Two sharp edges)

$$v_{i+1} = \frac{e_1 + 6v_i + e_2}{8}$$

Corner 0,1
(Three or more sharp edges)

$$v_{i+1} = \frac{n-2}{n} v_i + \frac{1}{n^2}\sum_j e_j + \frac{1}{n^2}\sum_j f_j$$

Ref: “Subdivision Surfaces”, Geri’s Game (1989) : Pixar Animation Studios

Camera Position Update

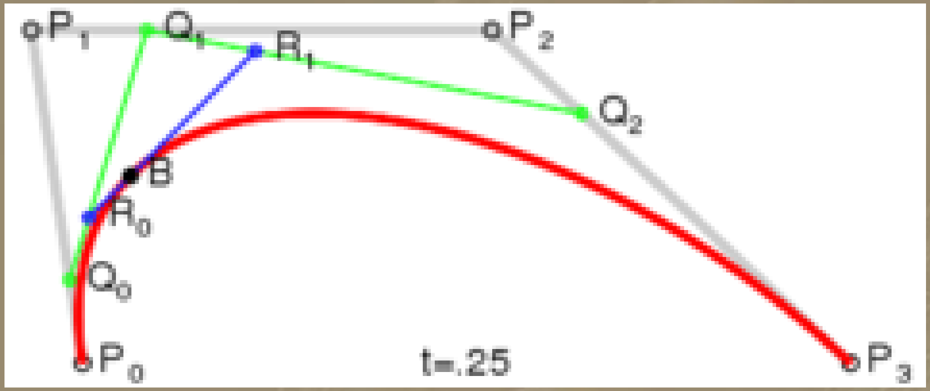
- Generate Random Key points:
  - Make sure the points don't go inside the "I"
- Interpolate Between the key points using:
  - B-spline

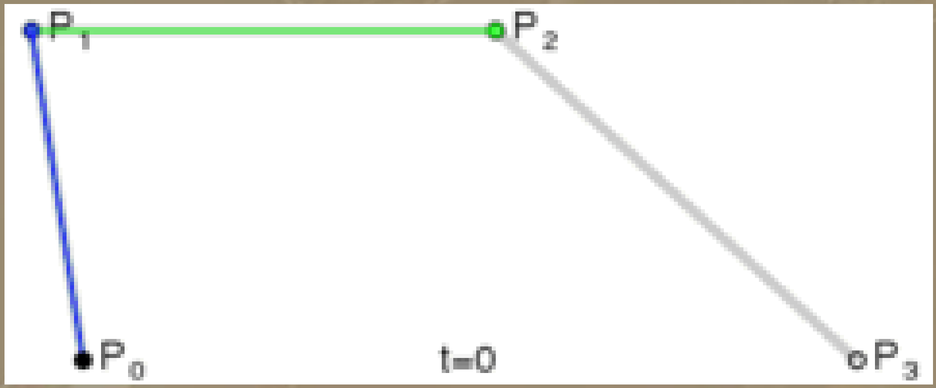
  or

  - Bezier

Cubic Bezier Curve

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , \ t \in [0,1]$$





Ref: “Subdivision Surfaces”, Geri’s Game (1989) : Pixar Animation Studios

Continuity

C^0 continuity

$$p_3 = q_0$$

C^1 continuity

$$p_3 = q_0$$

$$p_3 - p_2 = q_1 - q_0$$

G^1 continuity

$$p_3 = q_0$$

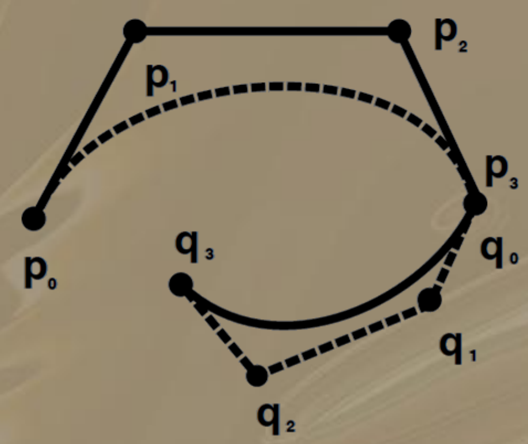
$$p_3 - p_2 = \alpha(q_1 - q_0)$$

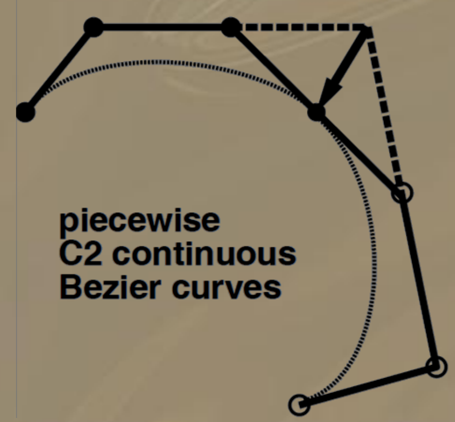
C^2 continuity

$$p_3 = q_0$$

$$p_3 - p_2 = q_1 - q_0$$

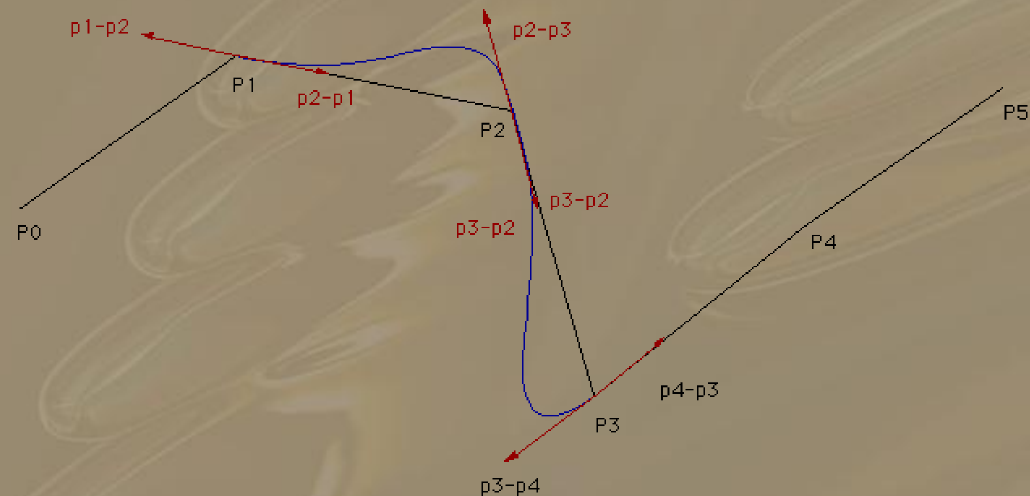
$$p_3 - 2p_2 + p_1 = q_2 - 2q_1 + q_0$$





Achieving C^2 Continuity

- Find tangent vectors: differences between subsequent key-frame points
  - for example: for the segment between p_1 and p_2 the four points use for the Bézier would be p_1, p_2, $2p_2$-p_3, p_3



- You can also use the de-Casteljau Algorithm

Cubic B-Spline

$$\mathbf{S}_i(t) = \sum_{k=0}^{3} \mathbf{P}_{i-3+k} b_{i-3+k,3}(t)\ ;\quad t \in [0,1]$$

- S_i is the i^{th} B-spline segment
- P is the set of control points
- segment i and k is the local control point index

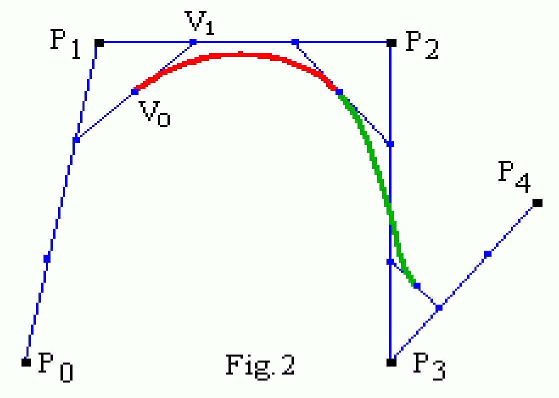


Fig. 2

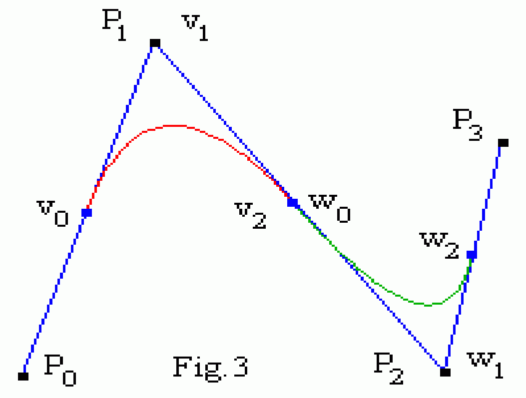


Fig. 3

- Again for continuity you can use de-Boor's alg.

Hooks Spring Law:

- Two ways:
  - Edges are considered as springs
  - If you don't want to worry about edges you can consider it's neighbor with all vertices…

$$f = -\left[k_s\left(\left\| x_a - x_b \right\| - r \right) + k_d (v_a - v_b) \frac{x_a - x_b}{\left\| x_a - x_b \right\|} \right] \frac{x_a - x_b}{\left\| x_a - x_b \right\|}$$

k_s = spring constant

k_d = damping constant

r = rest length

Ref: "Particle System Example", Paul Bourke, 1998

Gravitational Attraction

- **Two ways:**
  - **neighbors the have an edge with it**
  - **all the particles: you need to calculate the average position and add all the mass and consider that as one neighboring particle**

$$f = \frac{G\, m_a\, m_b}{\|x_a - x_b\|^2} \; \frac{x_a - x_b}{\|x_a - x_b\|}$$

G = universal gravitational constant = 6.672 x 10-11N m2 kg-2

Ref: “Particle System Example”, Paul Bourke, 1998

Repelling Based on Charge

- particles could have a charge:

Repel : if the charges are the same sign
Attract : if they are the opposite sign

$$f = \frac{k|q_a||q_b|}{\|x_a - x_b\|^2} \frac{x_a - x_b}{\|x_a - x_b\|}$$

k = Coulombs constant = 8.9875 x 109 N m2 C-2

Ref: “Particle System Example”, Paul Bourke, 1998