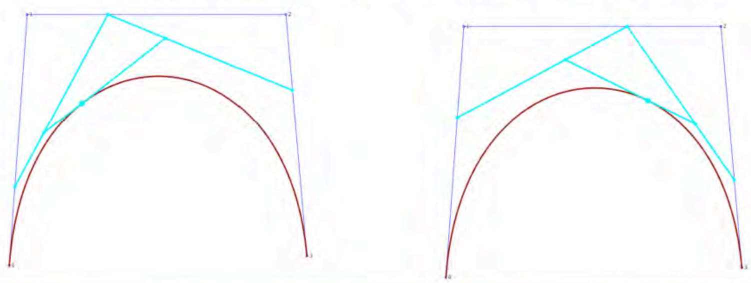
Bezier example

- Interaction
- Animation
- Tracking
- Menus
- Debugging

Point on cubic Bezier curve

```
pt cubicBezier(pt A, pt B, pt C, pt D, float t) {
return( s( s( s(A,t,B) ,t, s(B,t,C) ) ,
                              t,
            s( s(B,t,C) ,t, s(C,t,D) ) ) ); }
pt s(pt A, float s, pt B) {return(new
                pt( A.x+s*(B.x-A.x) , A.y+s*(B.y-A.y) )  ); };
```



Draw cubic Bezier curve

```
void draw() { background(121);
   noFill(); stroke(dred);  strokeWeight(3);
   drawCubicBezier(PP[0],PP[1],PP[2],PP[3]);

void drawCubicBezier(pt A, pt B, pt C, pt D) {
    beginShape();
     for (float t=0; t<=1; t+=0.02) {cubicBezier(A,B,C,D,t).v(); };
     endShape(); }
```

Manual control of time

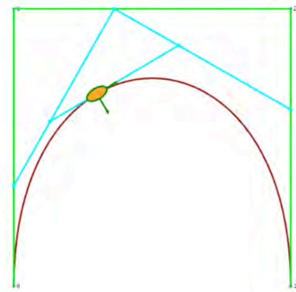
```
float t=0.5;
void draw() {

  if ((mousePressed)&&(keyPressed)&&
      mouseIsInWindow()&&(key=='t')) {
         t+=2.0*(mouseX-pmouseX)/height;
         fill(black); text("t="+t,20,40);
        };

  pt P = cubicBezier(PP[0],PP[1],PP[2],PP[3],t);
  P.show();
```

Automatic control of time

```
float t=0.5, tt=0.5;
void draw() {
  if (animate) {tt+=PI/180; if(tt>=PI*2) tt=0; t=(cos(tt)+1)/2;}
  else {… manual control …};
  pt P = cubicBezier(PP[0],PP[1],PP[2],PP[3],t); P.show();

void mousePressed() {
   k= Toggles.click();   m=-1;
   …
    if(k==++m) {animate=Toggles.v(m);
           if(animate) tt=acos(t*2-1);
                 else t=(cos(tt)+1)/2;  }
```

TRACKING

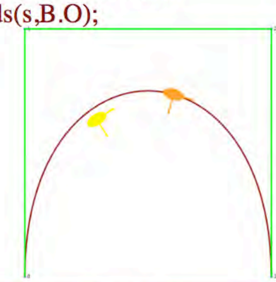
Frame



```
frame T= new frame ();
void draw() {
  pt P = cubicBezier(PP[0],PP[1],PP[2],PP[3],t);
  vec V = cubicBezierTangent(PP[0],PP[1],PP[2],PP[3],t);
  V.normalize();
  vec N=V.left();
  T.setTo(P,V,N);
  fill(dgreen); stroke(orange); T.show();
  pushMatrix(); T.apply(); ellipse(0,0,40,20); popMatrix();

class frame {     // frame [O I J]
   pt O = new pt();  vec I = new vec(1,0);  vec J = new vec(0,1)
  void setTo(pt pO, vec pI, vec pJ) {O.setTo(pO); I.setTo(pI); J.setTo(pJ); }
  void apply() {translate(O.x,O.y); rotate(angle());}
  void show() {float d=height/20; O.show(); I.makeScaledBy(d).showArrowAt(O);
     J.makeScaledBy(d).showArrowAt(O); }
```

Ghost frame

```
frame T= new frame (), F= new frame (), G= new frame (), H= new frame ();
void draw() {…. compute P, V, N as above
   T.setTo(P,V,N);
   H.moveTowards(T,0.1); G.moveTowards(H,0.1); F.moveTowards(G,0.1);
   if(showGhostFrame) {fill(yellow); stroke(yellow); F.show();
                     pushMatrix(); F.apply();  ellipse(0,0,40,20); popMatrix();};

class frame {pt O = new pt();  vec I = new vec(1,0);  vec J = new vec(0,1);
 void moveTowards(frame B, float s) {O.translateTowards(s,B.O);
    rotateBy(s*(B.angle()-angle()));}
 void rotateBy(float a) {I.rotateBy(a); J.rotateBy(a); }

class vec { float x=0,y=0;
 void rotateBy (float a) {float xx=x, yy=y;
      x=xx*cos(a)-yy*sin(a);
      y=xx*sin(a)+yy*cos(a); };
```



MENU

```
void setup() { size(1100, 800);
  loadButtons(); loadToggles();
void draw() {
 if(showMenu) {Buttons.show(); Toggles.show(); …
void loadButtons() {
 Buttons.add(new button("recursions",0,rec,7,1,7));    …
void loadToggles() {
 Toggles.add(new toggle("Animate",animate));   …

void mousePressed() {
 int k= Buttons.click();  int m=-1;
  if(k== ++m) {rec=int(Buttons.v(m));};    …
 k= Toggles.click();   m=-1;
  if(k==++m) {animate=Toggles.v(m); if(animate) tt=acos(t*2-1); …};…
  if(k==++m) {Toggles.B[m].V=false; reset(); };   // one time action, not toggle
```

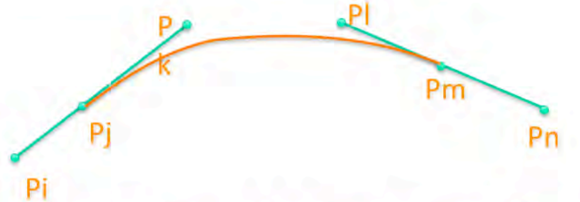
DEBUGGING

```
boolean printIt=false;
void draw() {
 …
 if (printIt)  P.write();
  …
  printIt=false;
 }

void keyPressed() {
   if (key=='?') printIt=true;
```

Fitting a Bezier span

- Hermite problem:
  - You are given 2 points and associated tangent directions
  - Find a “nice”smooth curve that interpolates these end conditions
  - Why do you need a cubic?

Fit Bezier to point and tangent constraints



- We create a joining curve with thickness between Pj and Pm as a cubic Bezier arc.
- d=|PmPj|, a=|PiPj|, b=|PmPn|
- Pk=Pj+dPiPj/3a, Pl=Pm+dPnPm/3b
- rk=rj+d(rj-ri)/3a, rl=rm+d(rm-rn)/2a
- rs=I(I(I(rj,s,rk),s, I(rk,s,rl)),s, I(I(rk,s,rl),s, I(rl,s,rm))), Bezier construction of r
- Ps=I(I(I(Pj,s,Pk),s, I(Pk,s,Pl)),s, I(I(Pk,s,Pl),s, I(Pl,s,Pm))), Bezier construction of P
- I(X,s,Y) = X+s(Y-X), linear interpolation