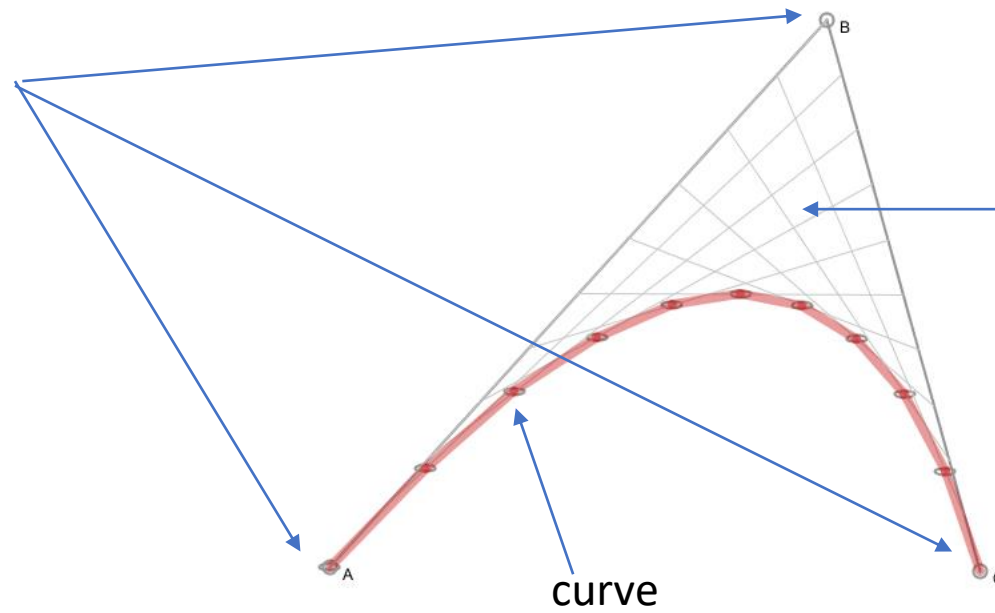


Inputs – 3 points



Linear Interpolation

curve

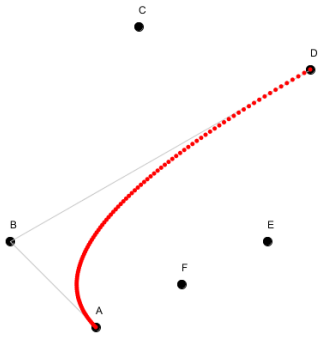
De Cateljau algorithm – parametric Bezier curve

Extremely high smoothness & performance

decasteljau\_linear\_interpolation | Processing 3.3

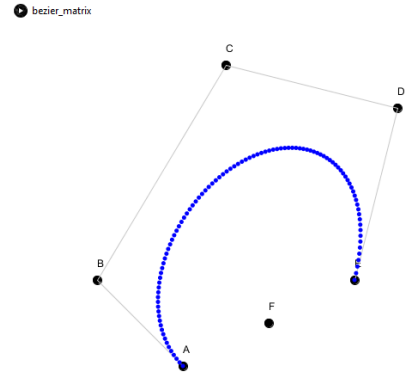
File Edit Sketch Debug Tools Help

```
decasteljau_linear_interpolation  Lerp  Pt  ▼
26  text("press g to show linear interpolation lines", 10,40);
27  text("press h to hide linear interpolation lines", 10,55);
28
29  ArrayList<Pt> ptsA=Lerp(A,B,t);ArrayList<Pt> ptsB=Lerp(B,C,t);ArrayList<Pt> ptsC=new
30  for(int i=0; i<t; i++){
31    Pt a=ptsA.get(i);Pt b=ptsB.get(i);
32    ptsC.add(new Pt(a.x+(b.x-a.x)*i/t,a.y+(b.y-a.y)*i/t));
33  }
34  ptsC.add(C); // linear interpolation operations
35
36
37  if(showLerp==true){for(int i=0; i<t; i++){ stroke(200);strokeWeight(1); line(ptsA.get
38  for(int i=0; i<ptsC.size()-1; i++){strokeWeight(7); stroke(200,0,0,100); line(ptsC.g
39  stroke(0);strokeWeight(1);
```



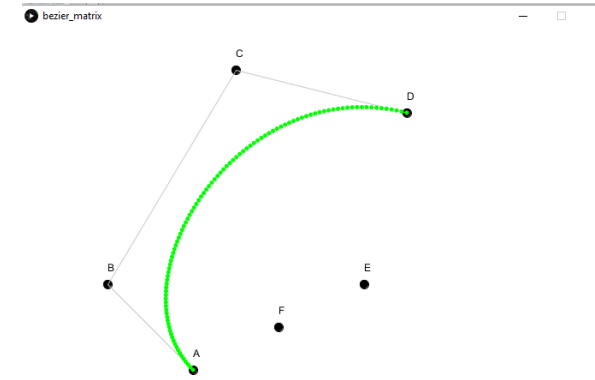
Input points : 3  
Quadratic Bezier curve

```
float x=(A.x*pow((1-t),2)) + (B.x*2*(1-t)*t) + D.x*pow(t,2);
float y=(A.y*pow((1-t),2)) + (B.y*2*(1-t)*t) + D.y*pow(t,2);
```



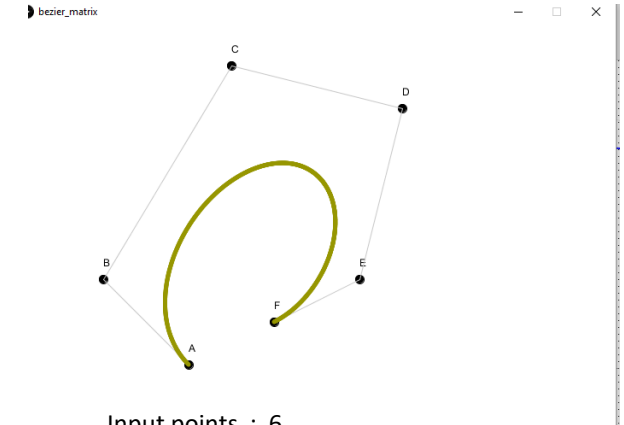
Input points : 5  
Quartic Bezier curve

```
float x=(A.x*pow((1-t),4)) + (B.x*4*pow((1-t),3)*t) + (C.x*6*sq(1-t)*sq(t)) + (D.x*4*(1-t)*pow(t,3)) + E.x*pow(t,4);
float y=(A.y*pow((1-t),4)) + (B.y*4*pow((1-t),3)*t) + (C.y*6*sq(1-t)*sq(t)) + (D.y*4*(1-t)*pow(t,3)) + E.y*pow(t,4);
```



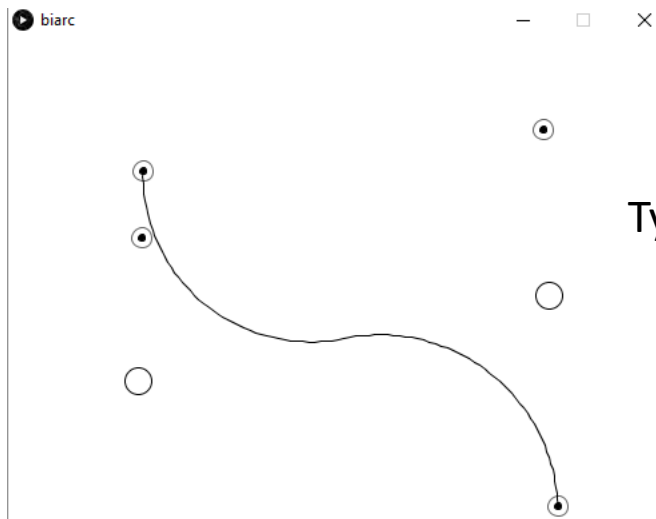
Input points : 4  
Cubic Bezier curve

```
float x=(A.x*pow((1-t),3)) + (B.x*3*pow((1-t),2)*t) + (C.x*3*(1-t)*sq(t)) + D.x*pow(t,3);
float y=(A.y*pow((1-t),3)) + (B.y*3*pow((1-t),2)*t) + (C.y*3*(1-t)*sq(t)) + D.y*pow(t,3);
```



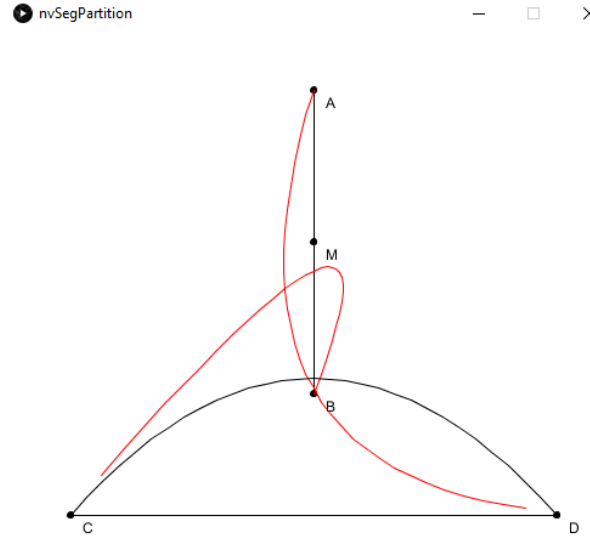
Input points : 6  
Quintic Bezier curve

```
float x=(A.x*pow((1-t),5)) + (B.x*5*pow((1-t),4)*t) + (C.x*10*pow((1-t),3)*sq(t)) + (D.x*10*sq(1-t)*pow(t,3)) + E.x*5*(1-t)*pow(t,4) + F.x*pow(t,5);
float y=(A.y*pow((1-t),5)) + (B.y*5*pow((1-t),4)*t) + (C.y*10*pow((1-t),3)*sq(t)) + (D.y*10*sq(1-t)*pow(t,3)) + E.y*5*(1-t)*pow(t,4) + F.y*pow(t,5);
```



Types of curves – Biarc

biarc



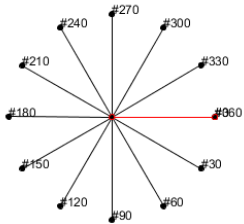
Types of curves – Bezier, lerp, Neville

```
nvSegPartition | Processing 3.3
File Edit Sketch Debug Tools Help

nvSegPartition Pt Seg bezierFunctions lerp nevilleFunctions
1 void nevilleFunction(float[] ar, float[] bs){
2   int numSeg;
3   float[][] pts01=new float[num+1][2];
4   float[][] pts23=new float[num+1][2];
```

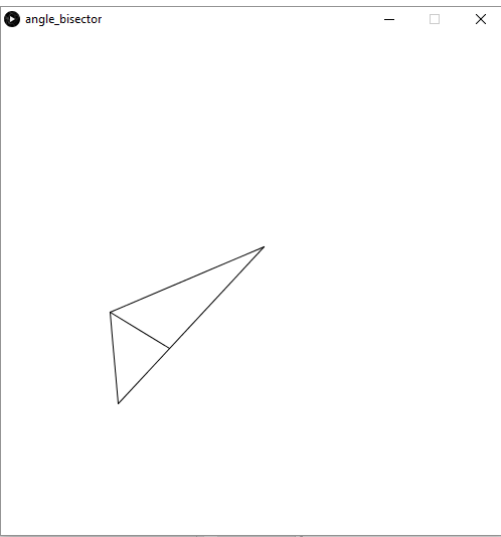
Functions Separated by tab

angles



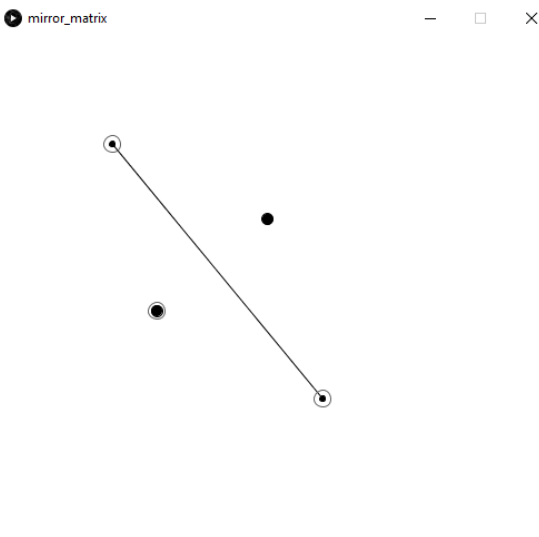
Angles

angle\_bisector



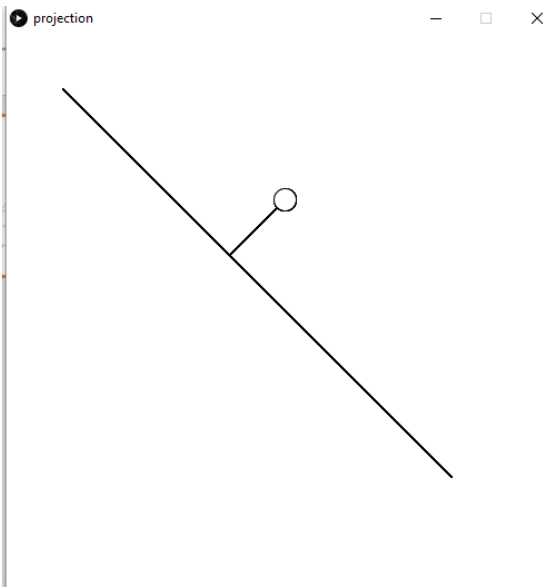
Angle\_bisector

mirror\_matrix

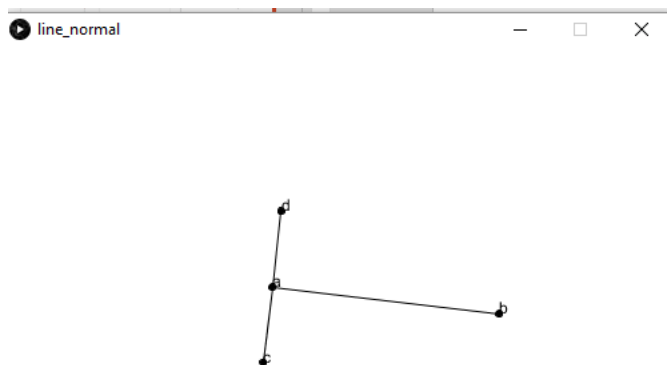


Mirror

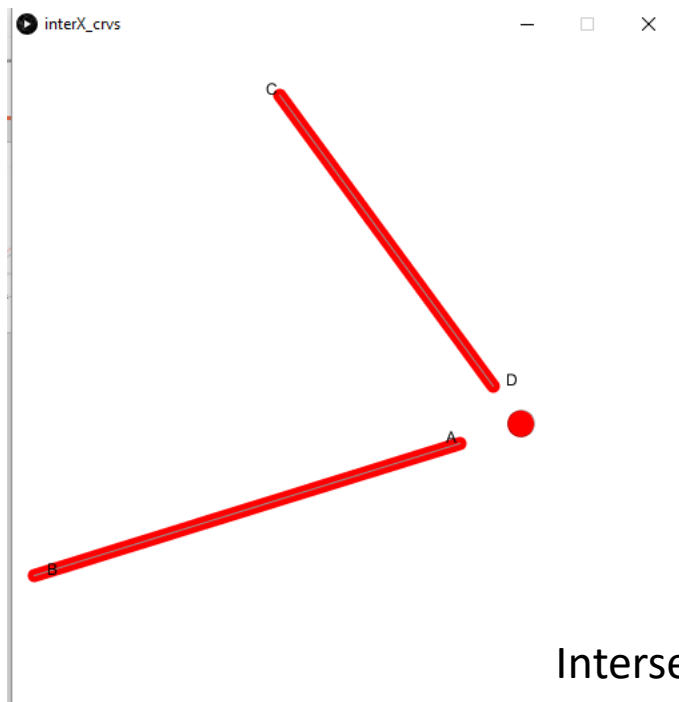
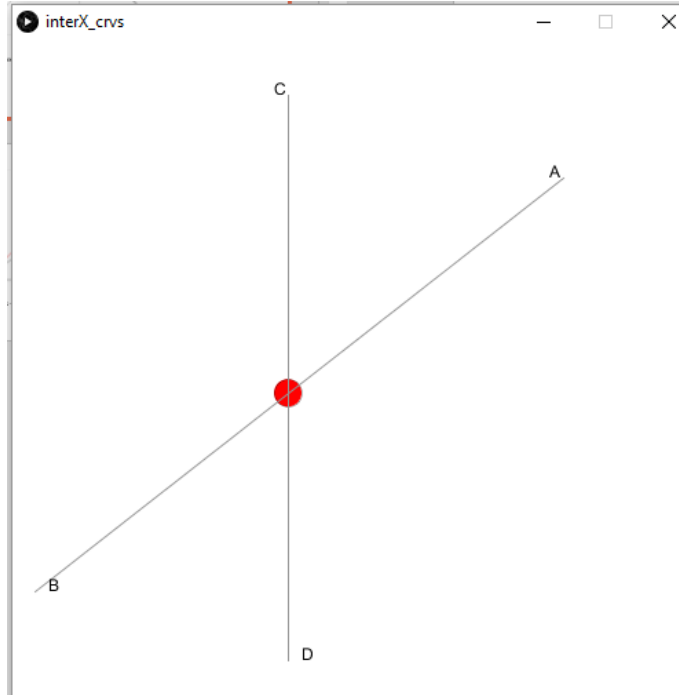
projection



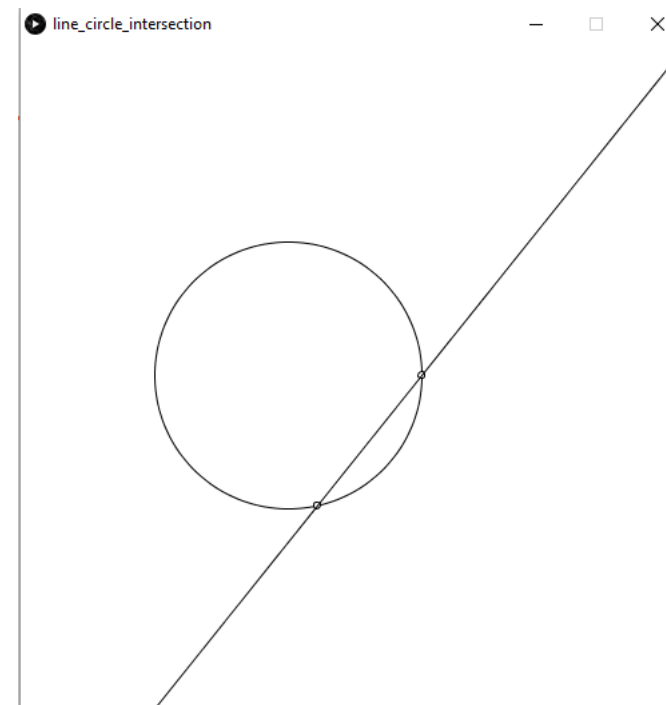
Projection



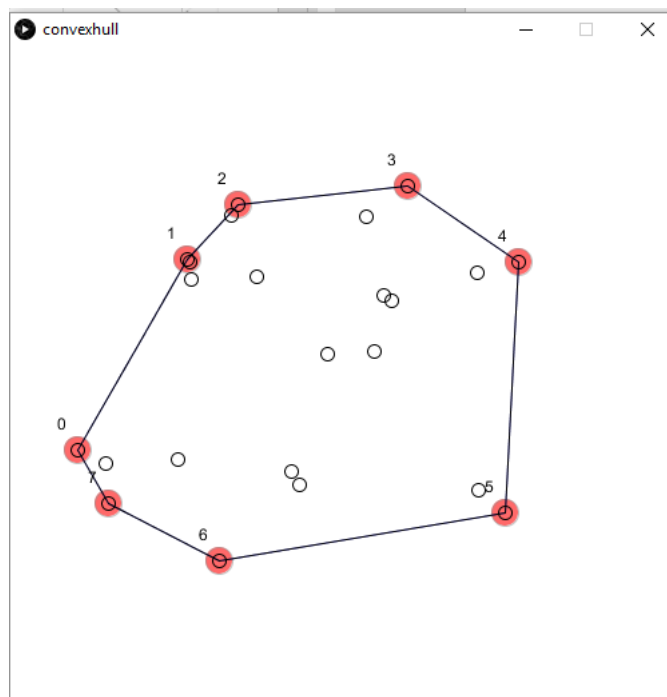
normals



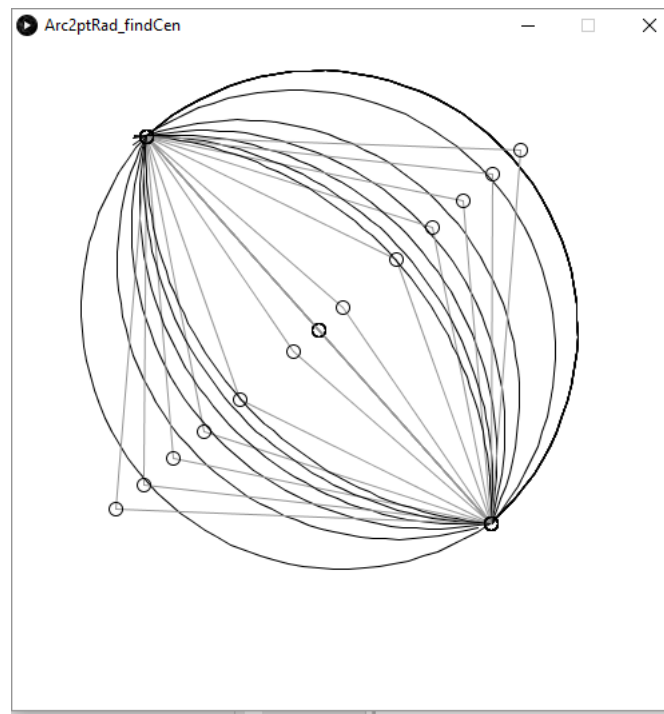
Intersection, detect\_apparent\_intersection



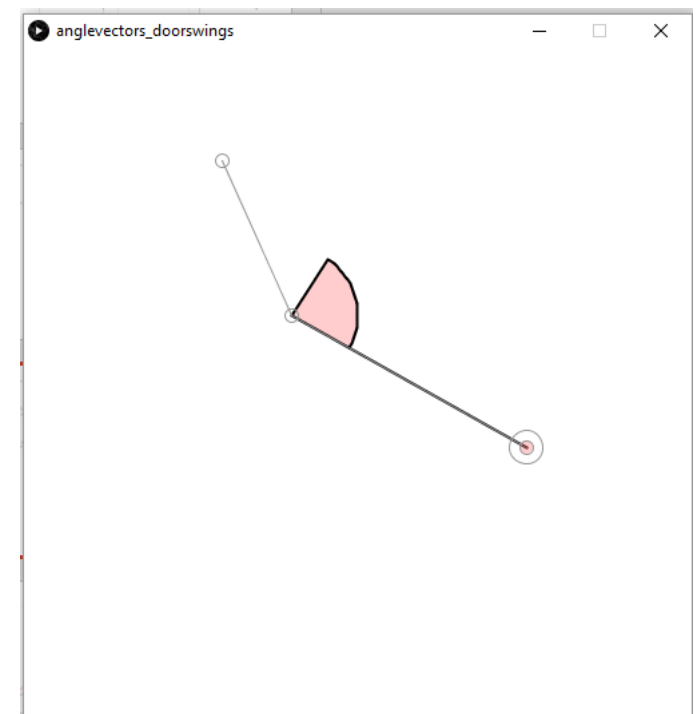
Line\_circle\_intersection



Convex hull

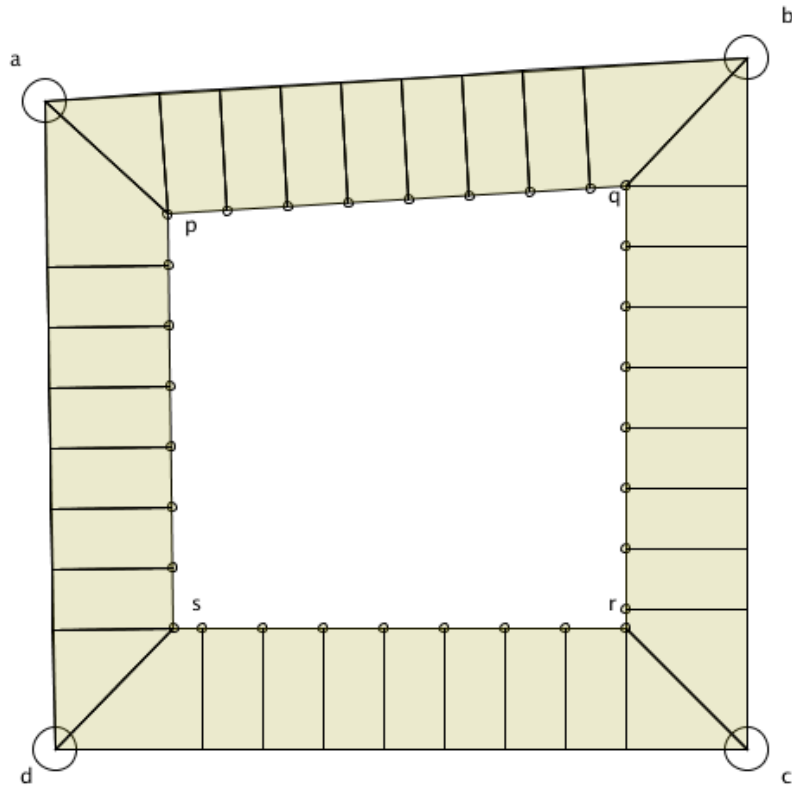


Arcs points normals



Intersection, arc, normal

Press 'r' or 'R' to reset



angle\_bisector\_subdivisions