To Draw a curve:

For $u = u_{beg}$; $u < u_{end}$; $u += \Delta u$
  drawline ($x(u)$, $x(\min(u_{end}, u + \Delta u))$);

Simple but not so great:

let $\Delta x = x(u + \Delta u) - x(u)$

if $|\Delta x|$ too small → wasted effort
if $|\Delta x|$ too large → curve looks bad

Desired value for $|\Delta x|$ depends on application

eg rendering - measure error in screen space

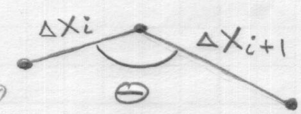
recall inv under prj. xF. ?

manufacturing - given error tolerance

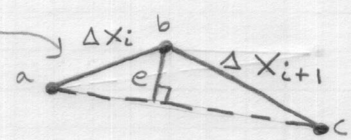
Adaptive: Pick Δu so that Δx "is good"

eg $|\Delta x| \sim 1$ pixel

or $\theta \sim 180$ deg



or $|e| \leq 1$ pixel



or $e = b - (a+c)/2$

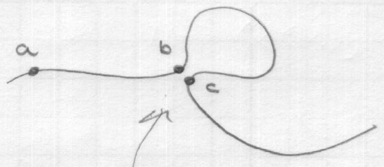
Good metric depends on application

& Cost of Metric

⊛ Δu varies over curve based on metric

What do these approximate?

Could still have problems :



Could miss entire loop

Bounding boxes -- small or flat → use a line

To draw a surface

For $u = u_{beg}$; $u < u_{end}$; $u += \Delta u$
For $v = v_{beg}$; $v < v_{end}$; $v += \Delta v$

drawQuad ($x(u,v)$,

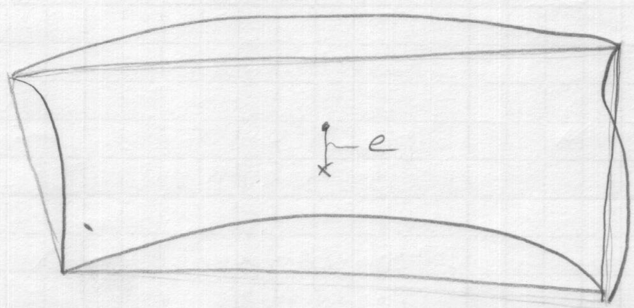
$x(\min(u_{end}, u+\Delta u), v)$,
$x(u, \min(v_{end}, v+\Delta v))$,

$x(\min(u_{end}, u+\Delta u), \min(v_{end}, v+\Delta v))$);

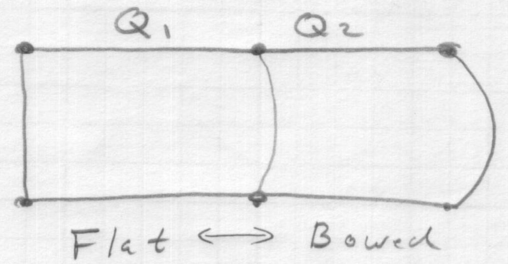
* Quads may not be flat

Again, simple but not so good...

→ Need to adjust Δu & Δv to get good approx to surface.



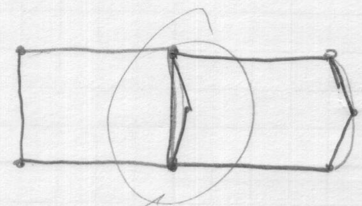
Problem: "Cracking"



Consider:

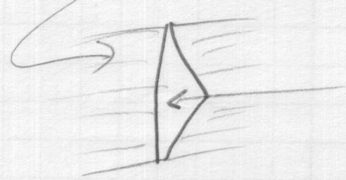
Q_1 Passes

Q_2 Fails



(?) Why is this a problem?

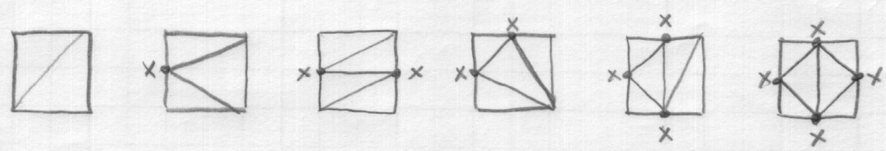
Is it a problem?



crack

gap in surf.

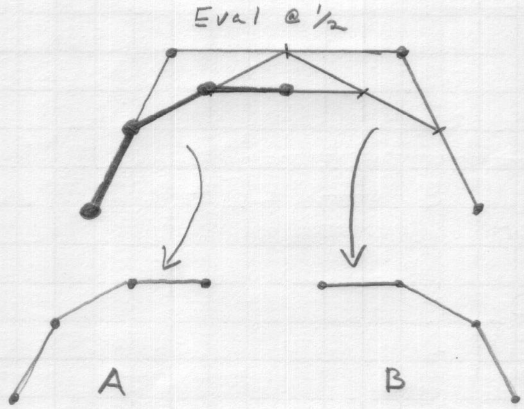
One way to avoid cracks -- split edges (& triangulate)



* Note, you may also care about the aspect ratio of the triangles.

Bézier Sub Division

- Based on De Casteljau Eval.



Split curve into two new Bézier curves

At some point control poly is nearly Flat or small
↳ then just draw control polygon

Express as a Matrix:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} 1/8 & 3/8 & 3/8 & 1/8 \\ 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$