p348
NAD

Solution notes for Computer Graphics & Image Processing 2002

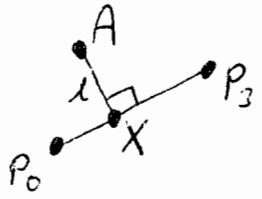
(a) A Bezier cubic curve can be specified by four points: P_0, P_1, P_2, P_3

The algorithm is:

```
drawBez (P0, P1, P2, P3) {
    if closeEnough (P0, P1, P2, P3)
    then drawLine (P0, P3)
    else {
        drawBez (P0, 1/2 P0 + 1/2 P1, 1/4 P0 + 1/2 P1 + 1/4 P2, 1/8 P0 + 3/8 P1 + 3/8 P2 + 1/8 P3)
        drawBez (1/8 P0 + 3/8 P1 + 3/8 P2 + 1/8 P3, 1/4 P + 1/2 P2 + 1/4 P3, 1/2 P2 + 1/2 P3, P3)
    }
}
```

closeEnough (P_0, P_1, P_2, P_3) checks whether the Bezier curve P_0, P_1, P_2, P_3 is approximated well enough (e.g. to within half a pixel width) by the line segment P_0P_3

closeEnough can be implemented as follows:



A is either P_1 or P_2

$l = |AX|$

need $l <$ tolerance for both P_1 and P_2 in order to be close enough AND for X to lie between P_0 and P_3 in both cases

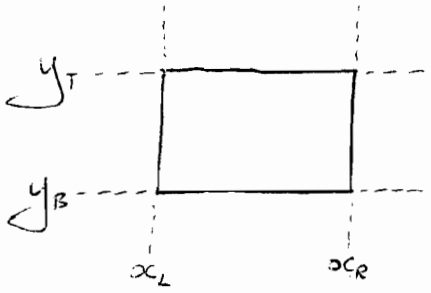
let $X = (1-t)P_0 + tP_3$

let $\overline{AX} \cdot \overline{P_0P_3} = 0$
solve to find t

if $0 \le t \le 1$ and $|\overline{AX}| <$ tolerance for both $A = P_1$ and $A = P_2$ then we are close enough.

(b) Check each endpoint against all edges of the rectangle to get a four bit code:

$a = (x < x_L)$ $\quad c = (y < y_B)$
$b = (x > x_R)$ $\quad d = (y > y_T)$



You get two codes $Z_1 = abcd$ for P_1
$Z_2 = abcd$ for P_2

if $Z_1 \wedge Z_2 \neq 0$ then reject (nothing to draw)

if $Z_1 \vee Z_2 = 0$ then accept (draw P_1P_2)

~~Otherwise clip the line by selecting an end point for which $Z_i \neq 0$~~

otherwise you need to clip the line
select one of Z_1 and Z_2 which is non-zero, Z_i
the non-zero bits of Z_i tell you which lines you could usefully clip against select one of these
now calculate the new end point

for example, to clip against $x = x_L$
$P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$
$Z_1 = 10$??
find a new location for P_1, $P_1' = (x_1', y_1')$

$$x_1' = x_L, \quad y_1' = \frac{x_L - x_1}{x_2 - x_1}(y_2 - y_1) + y_1$$

Similarly for the other three edges.

Once you have the new endpoint you need to put the new line through the whole algorithm again.

(c) clipBez (P_0, P_1, P_2, P_3)

find the bounding box of the Bezier by finding $x_{min}, x_{max}, y_{min}, y_{max}$ for P_0, P_1, P_2, P_3

compare this with the rectangle
- if entirely outside then draw nothing and stop
- if entirely inside then use algorithm (a) [i.e. no clipping required]
- otherwise: check tolerance, as in (a).
  - if within tolerance use algorithm in (b)
  - if outside tolerance subdivide (as in (a)) and recurse on clipBez.

Marking Scheme for Computer Graphics & Image Processing

(a) and (b) test knowledge of core material from the "2D computer graphics" part of the course

(c) stretches the students a little, but is not difficult

Allocation of marks – preliminary

(a) Bezier cubic specified by four points, P_0, P_1, P_2, P_3 1
can be approximated by straight line $\overline{P_0P_3}$ 1
need to check if within tolerance 1
need to check both P_1 and P_2 against $\overline{P_0P_3}$ 1
method for doing this 2
new point locations for subdivided curves as functions of P_0, P_1, P_2, P_3 1

(b) need to check <u>both</u> endpoints 1
against all four edges 1
trivial reject case 1
trivial accept case 1
need to clip one end point against one edge 1
how to chose that edge 1
how to calculate the new point 1
need to recurse 1

(c) check bounding box of Bezier against rectangle 1
how to calculate bounding box 1
trivial accept and what to then do 1
if not trivial reject then need to check tolerance 1
what to do if within or without tolerance 1