/ Advanced Graphics 2000 PAPER 9/Q4/PAGE 2

NAD

(a) ray is: $\left.\begin{array}{l} x = x_0 + t x_d \\ y = y_0 + t y_d \\ z = z_0 + t z_d \end{array}\right\} 0 \le t$

infinite cylinder is: $y^2 + z^2 = 1$

find solutions to:

$$(y_0 + t y_d)^2 + (z_0 + t z_d)^2 = 1$$

this will give zero, one, or two real values of t

if zero → no intersections

if one or two call them t_1 and t_2 ($t_1 = t_2$ in case of one value of t)

finite open cylinder is infinite cylinder with additional constraint that $x_{min} \le x \le x_{max}$

check where $x_1 = x_0 + t_1 x_d$ and $x_2 = x_0 + t_2 x_d$

1) if $x_1, x_2 < x_{min}$ or $x_1, x_2 > x_{max}$ then no intersection points

2) if $x_{min} \le x_1, x_2 \le x_{max}$ then intersection point is defined by ~~$x = x_0 + t x_d$ where~~ $t = \min(t_1, t_2)$

3) if only one of x_1, x_2 lies between x_{min} and x_{max} then the corresponding ~~that~~ value of t defines the intersection point

given the intersection point (x, y, z), the normal is going to be $(0, y, z)$

<u>unless</u> the intersection is on the inner face of the cylinder, in which case the normal is $(0, -y, -z)$ this can only happen in case 3 where the closer intersection point ~~on~~ the infinite cylinder is <u>not</u> on the finite cylinder

(b)
$$N_{1,1}(t) = \begin{cases} 1, & 1 \le t < 2 \\ 0, & o/w \end{cases}$$

$$N_{2,1}(t) = \begin{cases} 1, & 2 \le t < 4 \\ 0 & o/w \end{cases}$$

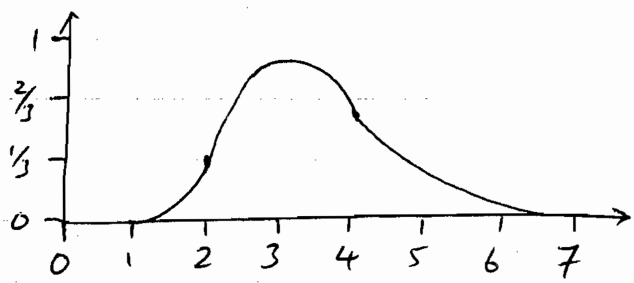
$$N_{3,1}(t) = \begin{cases} 1, & 4 \le t < 7 \\ 0 & o/w \end{cases}$$

$$N_{1,2}(t) = \begin{cases} t-1, & 1 \le t < 2 \\ \tfrac{1}{2}(4-t), & 2 \le t < 4 \\ 0 & o/w \end{cases}$$

$$N_{2,2}(t) = \begin{cases} \tfrac{1}{2}(t-2), & 2 \le t < 4 \\ \tfrac{1}{3}(7-t), & 4 \le t < 7 \\ 0, & o/w \end{cases}$$

$$N_{1,3}(t) = \begin{cases} \tfrac{1}{3}(t-1)^2, & 1 \le t < 2 \\ \tfrac{1}{6}(4-t)(t-1) + \tfrac{1}{10}(t-2)(7-t), & 2 \le t < 4 \\ \tfrac{1}{15}(7-t)^2, & 4 \le t < 7 \end{cases}$$

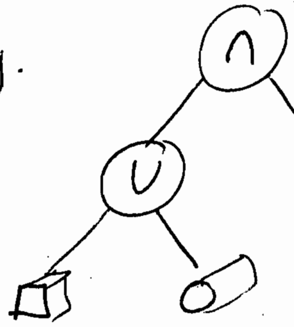
@ $t=1$ $N=0$
@ $t=2$ $N=\frac{1}{3}$ $N=\frac{1}{3}$
@ $t=4$ $N=\frac{3}{5}$ $N=\frac{3}{5}$
@ $t=7$ $N=0$



You would need four control points with this knot vector.

(c) An object built using CSG can be represented as a binary tree by putting the primitives at the leaves and the intersection/union/difference operators in the inner nodes

e.g.



Each leaf or node has a list of pairs of in/out intersections between the ray & that node.

These lists are combined by the inner nodes, and a new list passed up the ~~tree~~ tree. At the ~~tree~~ root of the tree the first intersection point in the list is the one that you want.

To combine two lists, sort them into ascending order & then keep track of whether you are:

in neither object (NONE)
in the left object only (LEFT)
" " right " " (RIGHT)
" both objects (BOTH)

pass up only transitions which take you into or out of the ~~valid set~~ defining state of your operator:

UNION:	NONE	(i.e. LEFT or RIGHT or BOTH)
INTERSECTION:	BOTH	
DIFFERENCE:	LEFT	(assuming left − right)