CYRUS-BECK (LIANG-BARSKY) CLIPPING

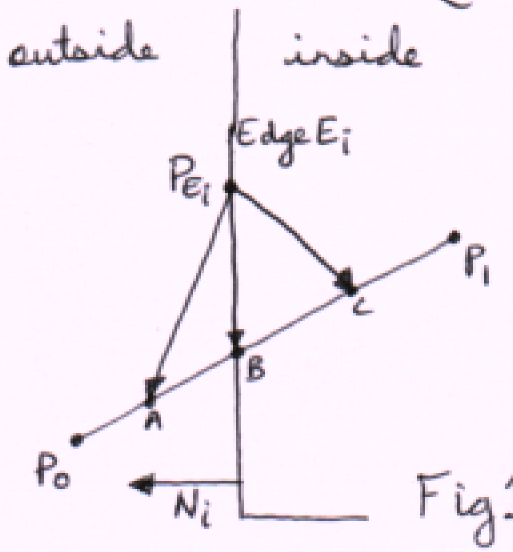

Fig1.

Let E_i be the edge against which we are trying to clip the line segment

$$P(t) = P_0 + (P_1 - P_0)t$$

Let N_i be the outward normal to the edge.

Then at point A, B, C on the line: (See Fig1.)

At A : $N_i \cdot [P(t) - P_{E_i}] > 0$

At B : $N_i \cdot [P(t) - P_{E_i}] = 0$

At C : $N_i \cdot [P(t) - P_{E_i}] < 0$

⇒ To calculate the 't' at which the line intersects the edge we solve.

$$N_i \cdot [P(t) - P_{E_i}] = 0$$

$$\Rightarrow N_i \cdot [P_0 + (P_1 - P_0)t - P_{E_i}] = 0$$

$$\Rightarrow t = \frac{N_i \cdot [P_0 - P_{E_i}]}{-N_i \cdot D} \quad (\text{with } D = P_1 - P_0) \quad \text{— (1)}$$

We get a valid value of 't' only is $N_i \neq 0$, $D \neq 0$ and $N_i \cdot D \neq 0$ (i.e., line is not parallel to the clip edge.)

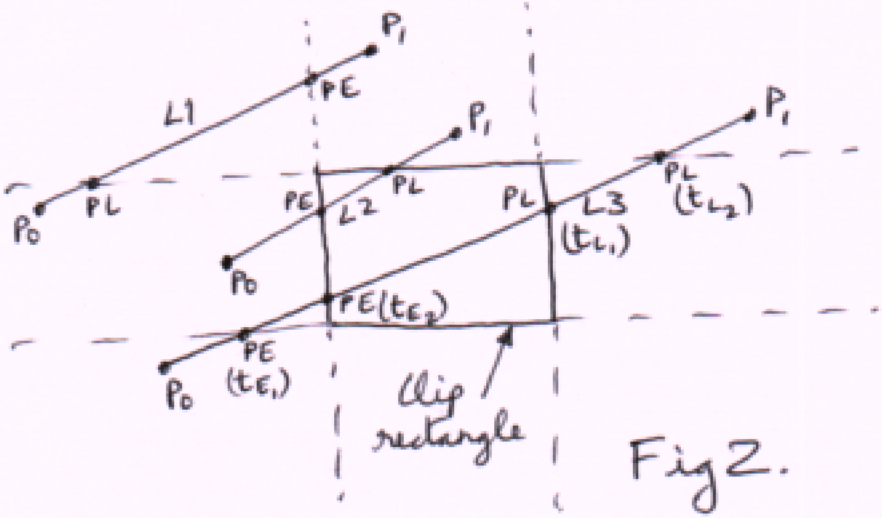

Fig2.

If we consider infinite extents of the clip edges, a line segment can intersect each of the four edges once, given a rectangular clipping window.

So we compute these 't' values for each intersection using equation ① and corresponding N_i and P_{E_i} for each edge. Retain only those t values that are between 0 and 1.

NOW,

- We say a 't' is a Potentially Entering (PE) intersection if at that 't', $N_i \cdot D < 0$
- We say a 't' is a Potentially Leaving (PL) intersection if at that 't', $N_i \cdot D > 0$

Now if the t value corresponding to PE, i.e., t_E is greater than the t value corresponding to PL, i.e., t_L - then the intersections lie outside the clip rectangle (as is the case with L1 in Fig2.)

If however, $t_E < t_L$, then the maximum t_E and minimum t_L values will be the intersections with the clip rectangle.

Note, that there can be multiple t_E and t_L values (as for L3 in Fig2). That is why the maximum t_E and minimum t_L are considered.

Pseudocode for CYRUS-BECK CLIPPING

Given Ni and PEi for each edge.

```
for (each line segment to be clipped) do
  {
   if (P1 == P0)
     then line is degenerate so clip as a point;
     else {
         tE = 0 ; tL = 1;
         for (each candidate intersection with a clip edge) do
           {
            if (Ni . D) != 0)
            then {
                 calculate t using (1);
                 Use sign of Ni . D to categorize at PE or PL;
                 if (PE) then tE = max (tE, t);
                 if (PL) then tL = min (tL, t);
                 }
           }
         if (tE > tL) then return no intersection;
         else
           return P(tE) and P(tL) as intersections;
         }
  }
```