Line Clipping

Lines

- Cyrus-Beck algorithm-2D/3D (1978)
  Parametric representation
- Liang-Barsky algorithm (1984)
  Early detection of rejectable segments
- Other Variations of Line Clipping

Write an algorithm to implement Cyrus-Beck Algorithm
Write an algorithm to implement Liang-Barsky Algorithm

Parametric Line clipping algorithm

- Cohn-Sutherland 2D (1968)
  Eliminates part of a sgment at a time.
- Cyrus – Beck Algorithm 2D/3D (1978)
  - Different and efficient
  - Still uses upright rectangles, can be applied to arbitrary rectangles.
- Liang – Barsky algorithm (1984)
  - More efficient for trivial rejection test

The Cyrus_Beck algorithm

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

We want that part of the line that corresponds to the values of t satisfying

$x_m \le x_0 + t(x_1 - x_0) \le x_M$

$y_m \le y_0 + t(y_1 - y_0) \le y_M$

$(x_1 - x_0)t \le x_M - x_0$

$(y_1 - y_0)t \le y_M - y_0$

$(x_1 - x_0)t \ge x_m - x_0$

$(y_1 - y_0)t \ge y_m - y_0$

For an oblique line not parallel ot x-axis and y-axis, the t values for intersections with clip boundary x_m, y_m, x_M, y_M are

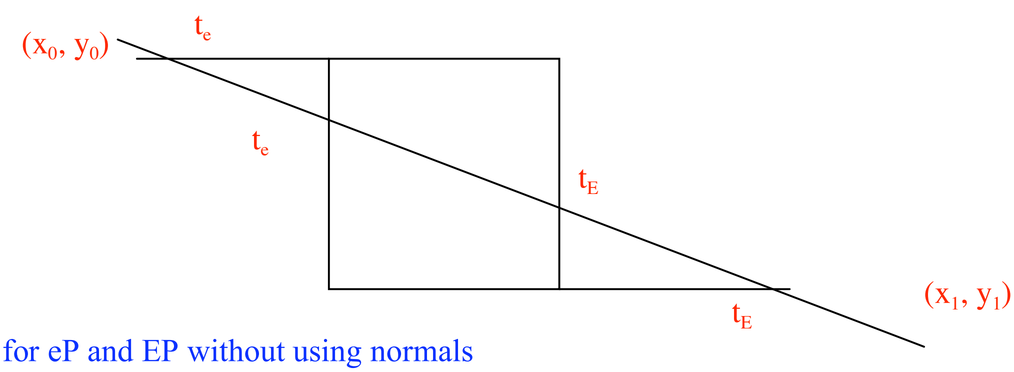
$$t = \frac{x_m - x_0}{x_1 - x_0}, \quad \frac{x_M - x_0}{x_1 - x_0}, \quad \frac{y_m - y_0}{y_1 - y_0}, \quad \frac{y_M - y_0}{y_1 - y_0}$$

Mark these values of t as t_e for entry or t_E for Exit.

Labels depend on normals to edges $x = x_m$, x_M; $y = y_m$, y_M and the direction D of line.

How do we determine t_e or t_E?

Example



Simple test for eP and EP without using normals

$$x_m, x_M \text{ intersects } \begin{cases} x_0 < x_1 & \\ t_{xm} = t_e & t_{xM} = t_E \\ x_0 > x_1 & \\ t_{xm} = t_E & t_{xM} = t_e \end{cases}$$

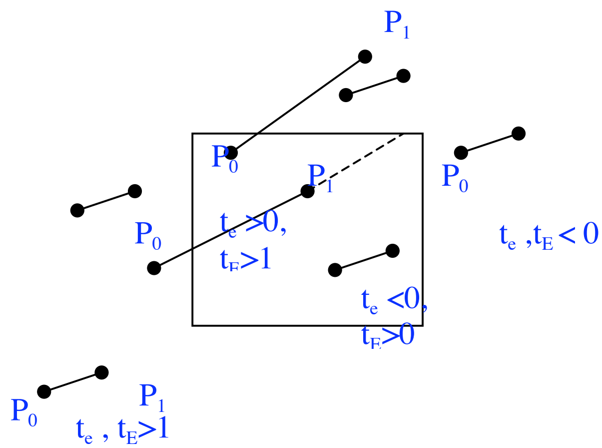
y_m, y_M intersects $\begin{cases} \quad y_0 < y_1 \\ t_{ym} = t_e \qquad t_{yM} = t_E \\ \quad y_0 > y_1 \\ t_{ym} = t_E \qquad t_{yM} = t_e \end{cases}$

Calculate max $\{t_e\} = t$
min $\{t_E\} = T$

If $t < T$, we have a line segment in the clip polygon.

Note: This criteria will drop a single point if it is on the clip rectangle boundary or otherwise.
Note: To retain a single point we use $t \leq T$ instead of $t < T$.
Note: Horizontal and vertical lines are handled separately.

Example



The Cyrus Beck Algorithm Implementation

For lines not parallel to the clip rectangle edges,
determine t values for intersection with lines x_m , x_M , y_m, y_M
for

x_m,x_M; t is found from $x = x_0 + t(x_1-x_0)$
y_m,y_M; t is found from $y = y_0 + t(y_1-y_0)$

Characterize the values of t as t_e or t_E

Select the largest of t_e and smallest of t_E
Find $t = \max(t_e)$
$T = \min(t_E)$
Let $t = \max(0, t_e)$, $T = \min(1,T_E)$

If $t < T$, then intersection segment is given by parameter interval (t, T)
else the line is rejected.

Note: This criteria will drop a single point if it is on the clip rectangle boundary or even inside clip window.
Note: To retain a single point we use $t \le T$ instead of $t < T$.

More Complete Implementation of Cyrus–Beck Algorithm (Liang-Barsky)

Set $t_e = 0$, $t_E = 1$

{ If line is || an Edge e_k **$(((x_0-x_1)*(y_0-y_1)=0))$** }

if $x_0=x_1$, {if $x_m < x_0 < x_M$

find t values at the intersection of $y= y_m$ and $y = y_M$

label t-value as entry or Exit point

If label on t-value is entry and $t>t_e$ then $t_e=t$

If $t_e>t_E$, return no intersection, reject the line

If label on t-value exit and $t<t_E$ then $t_E=t$

If $t_e>t_E$, return no intersection, reject the line

if $t_e > t_E$, then there is no intersection else Accept segment $[t_e , t_E]$ }

else

if $y_0=y_1$, { if $y_m < y_0 < y_M$

find t values at the intersection of $x= x_m$ and $x = x_M$

label t-value as entry or Exit point

If label on t-value is entry and $t>t_e$ then $t_e=t$

If $t_e>t_E$, return no intersection, reject the line

If label on t-value exit and $t<t_E$ then $t_E=t$

If $t_e>t_E$, return no intersection, reject the line

if $t_e > t_E$, then there is no intersection else Accept segment $[t_e , t_E]$ }

else

{If line is not parallel to any edge e_k ($\nparallel$ **$(((x_0-x_1)*(y_0-y_1)=0)$**) }

for k = 1 to 4 {for four edges}

At intesection t – value,

label t-value as entry or Exit point

If label on t-value entry and $t<t_E$ then $t_E=t$

If $t_e>t_E$, return no intersection, reject the line

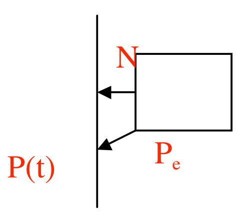
If label on t-value exit and $t<t_E$ then $t_E=t$

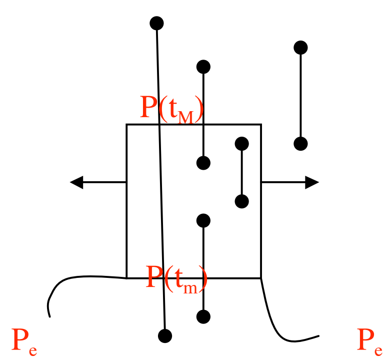
If $t_e>t_E$, return no intersection, reject the line

If $t_e \le t_E$

Accept segment $[t_e , t_E]$

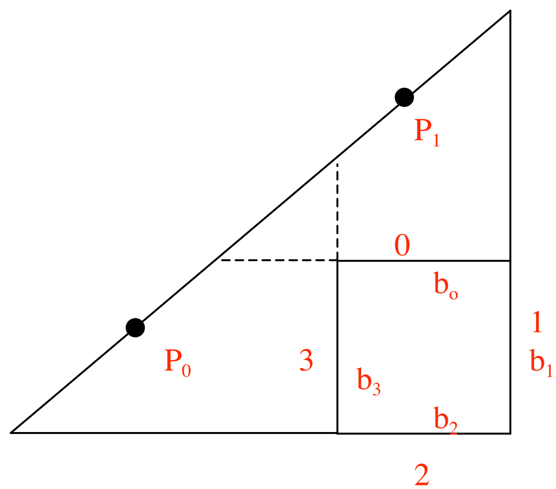
Else reject line





P_e is edge point.

Example



Cohn – Sutherland code(P_0) = 1000 code(P_1) = 0001

C_0&C_1=0 reject? not yet

Cyrus – Beck calculate

t_2	t_0	t_3	t_1
t_e	t_E	t_e	t_E

min $t_E = t_0$, max $t_e = t_3$

$t_E < t_e$ reject

Liang – Barsky

$t_e = 0$ $t_E = 1$

If $t_E < 0$, $t_e > 1$, reject line as soon as calculated

Edge order (2,3, 0,1)

edge 2 $t_e = 0$ larger than t_2, $t_E = 1$

edge 3 $t_e = t_3$, $t_E = 1$

edge 0 $t_e = t_3$, $t_E = t_0$

$t_0 < t_3$ $\therefore$ $t_E < t_e$ quick rejection

No need to go to edge 1

Edge order (3, 0,1,2)

edge 3 $t_e = t_3$, $t_E = 1$

edge 0 $t_e = t_3$, $t_E = t_0$

$t_0 < t_3$ $\therefore$ $t_E < t_e$ quick rejection

No need to go to edge 1 and 2

Edge order (0,1,2,3)

edge 0 $t_e = 0$ $t_E = t_o$

edge 1 $t_e = 0$ $t_E = t_0$ smaller that t_1

edge 2 $t_e = 0$ larger than t_2, $t_E = t_0$

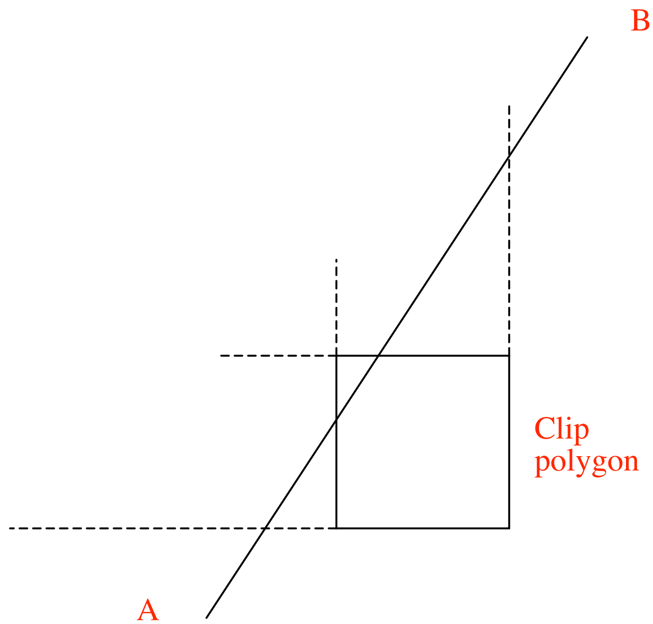
edge 3 $t_e = t_3$, $t_E = t_0$

$t_0 < t_3 \quad \therefore\ t_E < t_e$ rejection

Exercise

Differences between

- Sutherland
- Cyrus-Beck
- Liang- Barsky



P.142

Exc 3.1, 3.2, 3.3