Assignment ~ 02

Submitted by : TASNIM RAHMAN MOUMITA

ID : 22301689

Course Title : Computer Graphics

Course Code : CSE423

Section : 14

Date of submission : 23.07.2025

Ans. to the Q. NO-01(a)

Given,

starting point $\Rightarrow A\ (-5, 2)$

ending point $\Rightarrow B\,(25, 40)$

Now, $\boxed{\text{let, } P(t) = A + t(B-A),\ t \in [0,1]}$

the parametric equation,

$$P(t) = (x(t), y(t)) = x_0 + t\,(x_1 - x_0),\ y_0 + t\,(y_1 - y_0)$$

$\therefore$

$$x(t) = -5 + t\,(25-5)) = -5 + 30t$$

$$y(t) = 2 + t\,(40-2) = 2 + 38t$$

here,
$x_0 = -5$
$x_1 = 25$
$y_0 = 2$
$y_1 = 40$

$\therefore$ parametric equation will be,

$$P(t) = (-5 + 30t,\ 2 + 38t), \quad t \in [0,1]$$

(Ans:)

And,

Coordinates at $t = 2/3$:

$$x(2/3) = -5 + 30 \cdot 2/3$$
$$= 15$$

and $$y(2/3) = 2 + 38 \cdot 2/3$$
$$\approx 27 \cdot 33$$

[P.T.O.]

$\therefore P(2/3) = \left(15, \frac{82}{3}\right)$.

$\approx (15, 27.33)$

(Ans:)

Again,

when $t = 5$,

It is not inside the segment.

$\because$ start point in range $\rightarrow 0$

end point in range $\rightarrow 1$

$\therefore t = 5$ is lying outside the line segment.

(Ans:)

Ans. to the Q. NO - 01 (b)

Role of parametric equation in the cyrus-Beck algorithm:

1) → Representation of line as $P(t) = P_0 + t(P_1 - P_0)$

2) → The value of t : determines entry and exit points of the line with respect to the convex polygon.

3) → Allows calculating intersections with edges using dot products and normal vectors.

4) → Clipping is done by finding valid value of t in the range $[t_{max}, t_{min}]$ where the line is inside the clipping window.

(Ans:)

[P.T.O.]

Ans. to the Q. NO - 01 (c)

Given,

window clipping → $x \in [-15, 60]$, $y \in [5, 120]$

line segment → $P_0 = (20, 30)$

$P_1 = (90, 80)$

According to parametric equation using given values: $P(t) = (x(t), y(t)) = (x_0 + t(x_1 - x_0), y_0 + t(y_1 - y_0))$

$\Rightarrow P(t) = (20 + 70t, 30 + 50t)$, $t \in [0, 1]$

here,
$x_0 = 20$
$x_1 = 90$
$y_0 = 30$
$y_1 = 80$

Now,

4 edges of the window and their normal vectors:

Edge	Point on edge	Normal Vector (n)	Direction.
Left	$x = (-15, 5)$ to $(-15, 120)$	$(-1, 0)$	Faces left to right
Right	$(x = 60, 5)$ to $(60, 120)$	$(1, 0)$	Faces right-to-left
Bottom	$y = (60, 5)$ to $(-15, 5)$	$(0, -1)$	Faces bottom-top
Top	$(-15, 120)$ to $(60, 120)$	$(0, 1)$	Faces top-bottom

We Know,

Cyrus - Beck Line Clipping Formula:

$$t = \frac{(n \cdot (P_e - P_0))}{n \cdot D}$$

Here,

$P_0 = (20, 30)$

Direction vector,

$D = ((90-20), (80-30))$

$= (70, 50)$

P_e = a point on the edge

n = normal vector of the edge

for left edge: $(x = -15)$

$n_L = (-1, 0)$

$P_e = (-15, y)$

$\approx$ [let, use $(-15, 0)$]

$D \cdot n_L = (70, 50) \cdot (-1, 0)$

$= -70$

$-(P_0 - P_e) \cdot n_L = -((20, 30) - (-15, 0) \cdot (-1, 0)$

$= 35$

$\because D \cdot n = -70 < 0$ (entering)

$t_L = \frac{35}{-70} = -0.5$ [potential exit]

$t_{enter} = \max(0, -0.5) = 0$

$t_{exit} = 1$

[P.T.O.]

for right edge: $(x=60)$

$P_e = (60,0)$

$n_R = (1,0)$

$D.n = (70, 50) \cdot (1,0) = 70$ (exit)

$-(P_0 - P_e) \cdot n_R = -((20,30) - (60,0)) \cdot (1,0)$

$= 40$

$t_R = 40/70 = 4/7 \approx 0.5714$

$t_{enter} = 0$

$t_{exit} = \min(1, 4/7) = 4/7$

for bottom edge: $(y=5)$

$P_e = (0,5)$

$n_b = (0,-1)$

$D \cdot n_b = (70,50) \cdot (0,-1) = -50$ (entry)

$-(P_0 - P_e) \cdot n_b = -((20,30) - (0,5)) \cdot (0,-1)$

$= 25$

$t_b = 25/-50 = 0.5$

$t_{enter} = \max(0, -0.5) = 0$

$t_{exit} = 4/7$

For top edge ($y = 120$):

$P_e = (0, 120)$

$n_T = (0, 1)$

$D \cdot n_T = (70, 50) \cdot (0, 1) = 50$ (exit)

$-(P_0 - P_e) \cdot n_T = -((20, 30) - (0, 120)) \cdot (0, 1)$

$= 90$

$t_T = \dfrac{90}{50} = 1.8$

$$\left.\begin{aligned} t_{enter} &= 0.5714 \\ t_{exit} &= \min(1) = 1 \end{aligned}\right\} \because (t_e \notin [0,1), >1) \; \therefore \text{no update to } t_{enter}.$$

$\because$

$0.5714 > 1$

$\therefore$ the line is not rejected.

The cyrus-Beck algorithm fundamentally finds the segment of the line $(P_1 + t \cdot D)$ lies within convex polygon.

We get,

$t_L = -0.5$ (exit left - before P_1)

$t_b = -0.5$ (exit bottom - before P_1)

$t_T = 1.8$ (enters top - after P_2) [P.T.O.]

the only intersection point within $[0,1]$ where the line is exiting the window is at $t = 4/7$ [with right boundary].

$\therefore$ the clipped segment $\Rightarrow$ $P(0)$ to $P(4/7)$

$\therefore$ the line segment from $(20, 30)$ to $(90, 80)$ needs to be clipped.

The clipped line's endpoints:

(i) first endpoint: $P(0) = (20, 30)$ $\left[\because P_1 \text{ is inside the window}\right]$

(ii) second endpoint: $P(4/7)$

$$\Rightarrow x = 20 + 70 \cdot (4/7) = 60$$

$$\Rightarrow y = 30 + 50 \cdot (4/7) \approx 58.57$$

$$\therefore (x, y) = (60, 58.57)$$

$\therefore$ The clipped line segment $\Rightarrow (20,30)$ to $(60, 58.57)$

(Ans:)

Ans. to the Q. NO - 02 (a)

Scenarios where Cohen - Sutherland fails:

→ (i) Non-rectangular Clipping window:

This algorithm fails, becomes inefficient for arbitrary polygonal clip regions.

→ (ii) Inefficient for lines that require multiple clipping steps.

→ (iii) In floating point arithmetic, precision errors may cause incorrect values of endpoints.

* Drawbacks:

(i) Only supports rectangular clipping regions.

(ii) Repetitive calculation for partial segments, especially in complex scenarios.

(iii) It may clip unnecessarily, even when the line is trivially accepted/rejected.

[P.T.O.]

(iv) Inefficient for clipping large numbers of lines, especially in dynamic scenes/animations.

Ans. to the Q. NO – 02(b)

Given,

Clip region: $x_{min} = -40$, $y_{min} = -20$, $x_{max} = 20$, $y_{max} = 30$

Line segment: $P_1 = (-10, -50)$, $P_2 = (15, 25)$

Now,

Region	Condition
Top	$y > y_{max}$ (30)
Bottom	$y < y_{min}$ (−20)
Right	$x > x_{max}$ (20)
Left	$x < x_{min}$ (−40)

for P_1 $(-10, -50)$:

here,

$x = -10$, → within x range → Right $= 0$ left $= 0$

$y = -50 < -20$ → Bottom $= 1$

And Top $= 0$

∴ ABRL → 0100 → (4-bit code) → P_1

for P_2 $(15, 25)$:

$x = 15$ [within x range]

$y = 25$ [within y range]

∴ ABRL → 0000 [4-bit code] → P_2

Now,

P_1 → 0100

P_2 → 0000

(AND) → 0000

→ Not both outside in the same region

→ Partially inside

→ not trivially rejected

→ continue clipping.

[P.T.O.]

Now,

$P_1 = (-10, -15)$ is outside the bottom.

bottom edge $\rightarrow (y = -20)$

the intersections:

We Know,

$$x = x_1 + (x_2 - x_1)\,t$$

$$y = y_1 + (y_2 - y_1)\,t$$

from:

$P_1(-10, -15)$

$P_2(15, 25)$

(i) at $y = -20$:

$$y = y_1 + (y_2 - y_1)t$$

$$\Rightarrow -20 = -50 + 75\,t$$

$$\Rightarrow t = 0.4$$

(ii) at $t = 0.4$, value of x:

$$x = -10 + (15 - (-10)) \cdot 0.4$$

$$= 0$$

$\therefore$ New Clipped point $(0, -20)$ [intersection]

Now,

Replacing P_1 with $P_1'(0,-20)$ and repeating the calculations:

New line segment: $P_1'(0,-20)$ to $P_2(15,25)$

for $P_1'(0,-20)$:

$x_1' = 0$ [between x_{min} & x_{max}]

$y_1' = -20$ [on y_{min}]

$\therefore$ P_1' is inside the window.

$\therefore$ Outcode (P_1') = 0000 (4-bit code)

for $P_2(15,25)$: 0000 [from previous calculation]

$\downarrow$

outcode (P_2)

Now,

Outcode (P_1') AND Outcode (P_2)

(0000) AND (0000)

= 0000

$\therefore$ the line segment is fully accepted.

$\therefore$ final clipped line segment $(0,-20)$ to $(15,25)$ (Ans.)