



SNS COLLEGE OF ENGINEERING

KURUMBAPALAYAM(PO), COIMBATORE - 641 107

ACCREDITED BY NAAC-UGC WITH 'A' GRADE

APPROVED BY AICTE, RECOGNIZED BY UGC & AFFILIATED TO ANNA UNIVERSITY, CHENNAI

Department of Information Technology

Course Name - COMPUTER GRAPHICS

III Year / V Semester

**Unit 1 - CLIPPING ALGORITHMS FOR LINES, REGULAR
POLYGONS, CIRCLES AND ARCS**

Topic :Graphics Input Primitives and Devices



CLIPPING

- ❑ Clipping is a process of dividing an object into visible and invisible portions and displaying the visible portion and discarding the invisible portion.
- ❑ Types of Clipping:

Generally we have Clipping algorithm for the following primitive type:

- ❖ Point Clipping
- ❖ Line Clipping
- ❖ Area Clipping (Polygon)
- ❖ Curve Clipping
- ❖ Text

Clipping

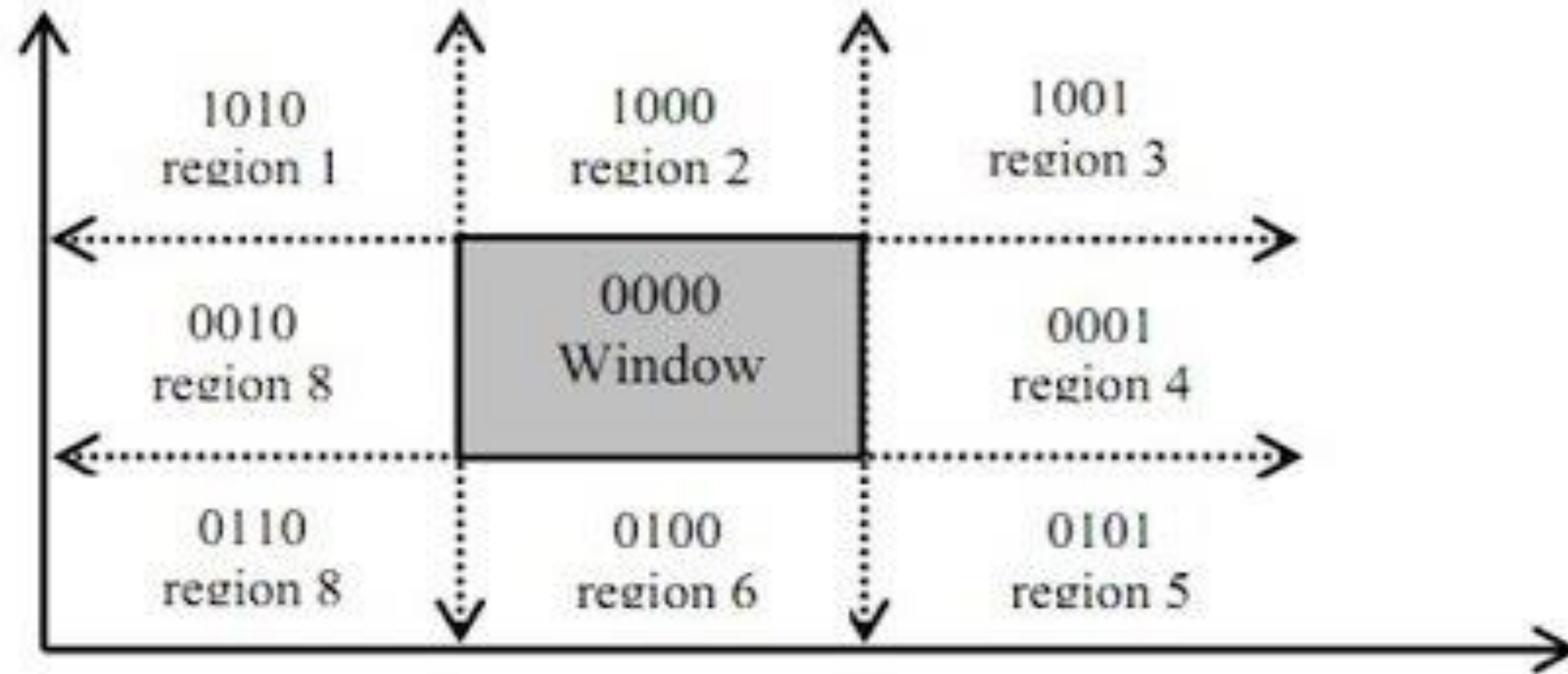


COHEN SUTHERLAND LINE CLIPPING ALGORITHM



- In this algorithm, we will divide the view pane into nine equal segments that only serve the viewport.
- Now, we will represent the top, bottom, left, and right corner of the window with 4 bits. This 4bit can be described with the following point that:
- If an object lies within any particular corner position, that corner value will be 1, else it will be 0.
- The allocation of bits depends on “**TBRL**” (Top, Bottom, Right, Left) rule.

COHEN-SUTHERLAND LINE CLIPPING ALGORITHM



With respect to a window, the line can be

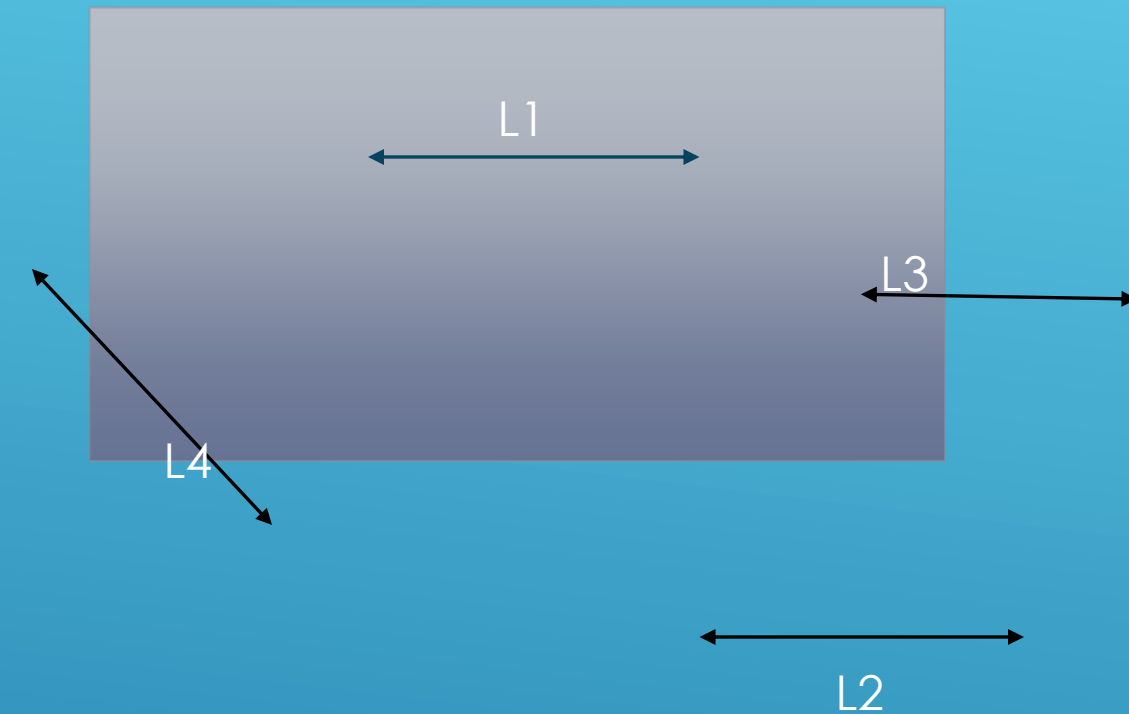
- 1) perfectly inside (L_1)
- 2) perfectly outside (L_2)
- 3) partially inside (L_3)
- 4) end points outside } (L_4)
Line segment inside }

if the line is

perfectly inside (L_1) \rightarrow accept the line.

perfectly outside (L_2) \rightarrow reject the line.

crosses window boundary (L_3, L_4)
 \rightarrow clip the line.





Encode the endpoints of the line

- If the two endpoints have the code 0000 & AND is 0000 the line is completely inside. So accept the line
- If the two endpoints are non zero & their AND is non zero the line is completely outside. So reject the line
- If the two endpoints are non zero & their AND is 0000 the line is partially inside. So clip the line.
 - ❖ Clipping needs the intersection points(s)
 - ❖ If a point is outside any window boundary find the intersection point on the window boundary



1) End points L1

0000

0000

-----AND

0000

Completely inside accept the line

2) End points L2

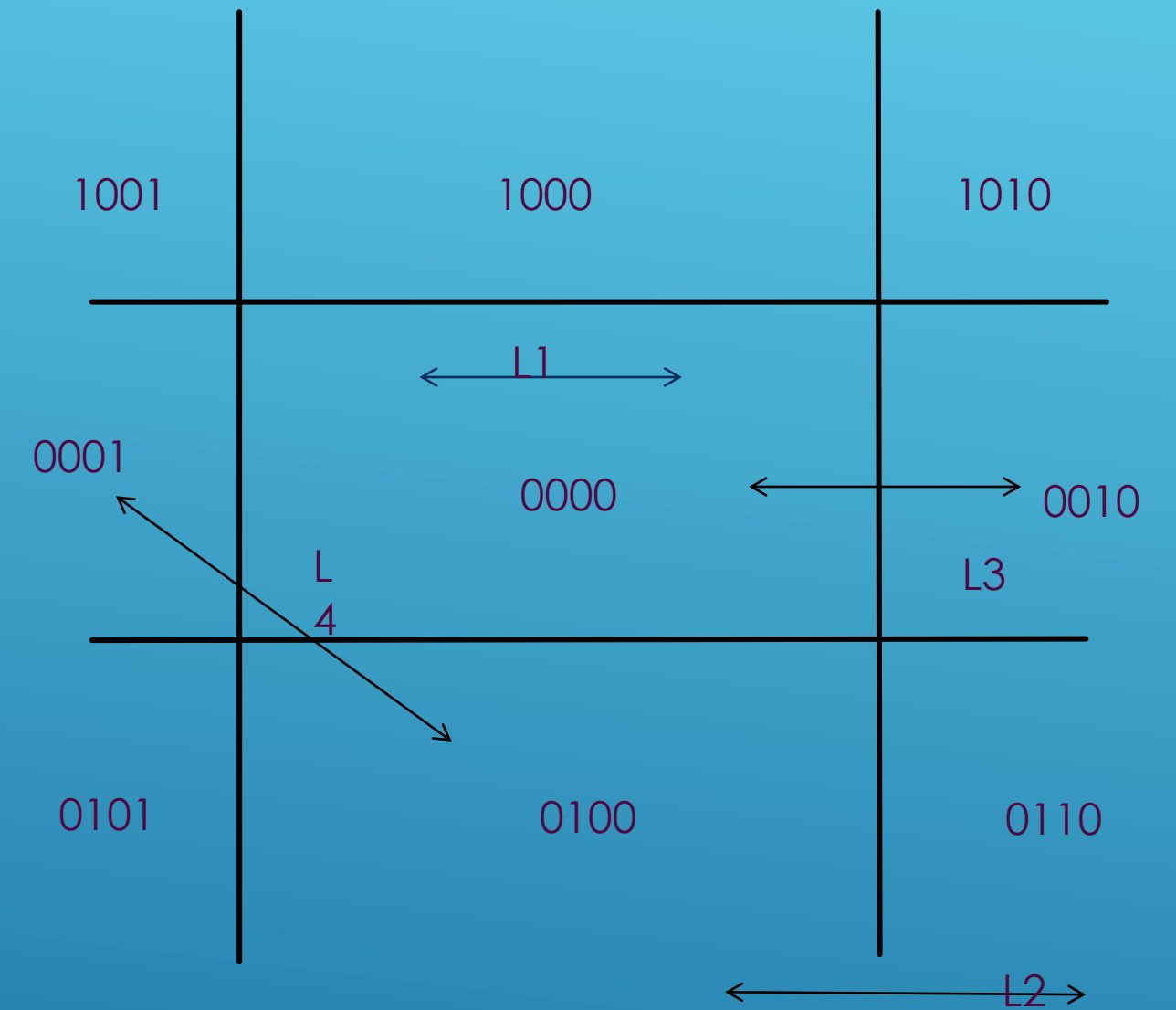
0100

0110

-----AND

0100

Completely outside, so reject the line





End points L3

0000

0010

-----AND

0000

partially inside , need clipping , find intersection point

End points L4

0001

0100

-----AND

0000

partially inside , need clipping , find intersection point

New points

0000

0000

-----AND

0000

Completely inside, so accept the line

New points

0001

0000

-----AND

0000

partially inside , need clipping , find intersection point



New points

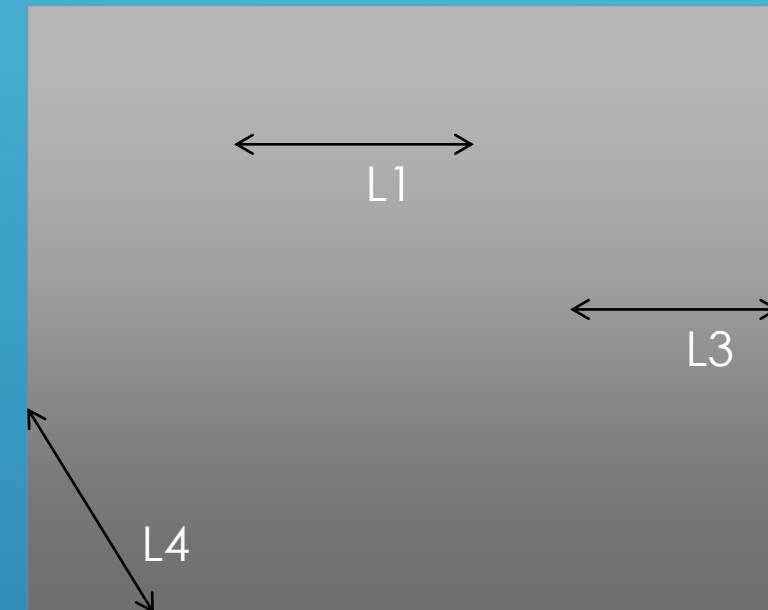
0000

0000

-----AND

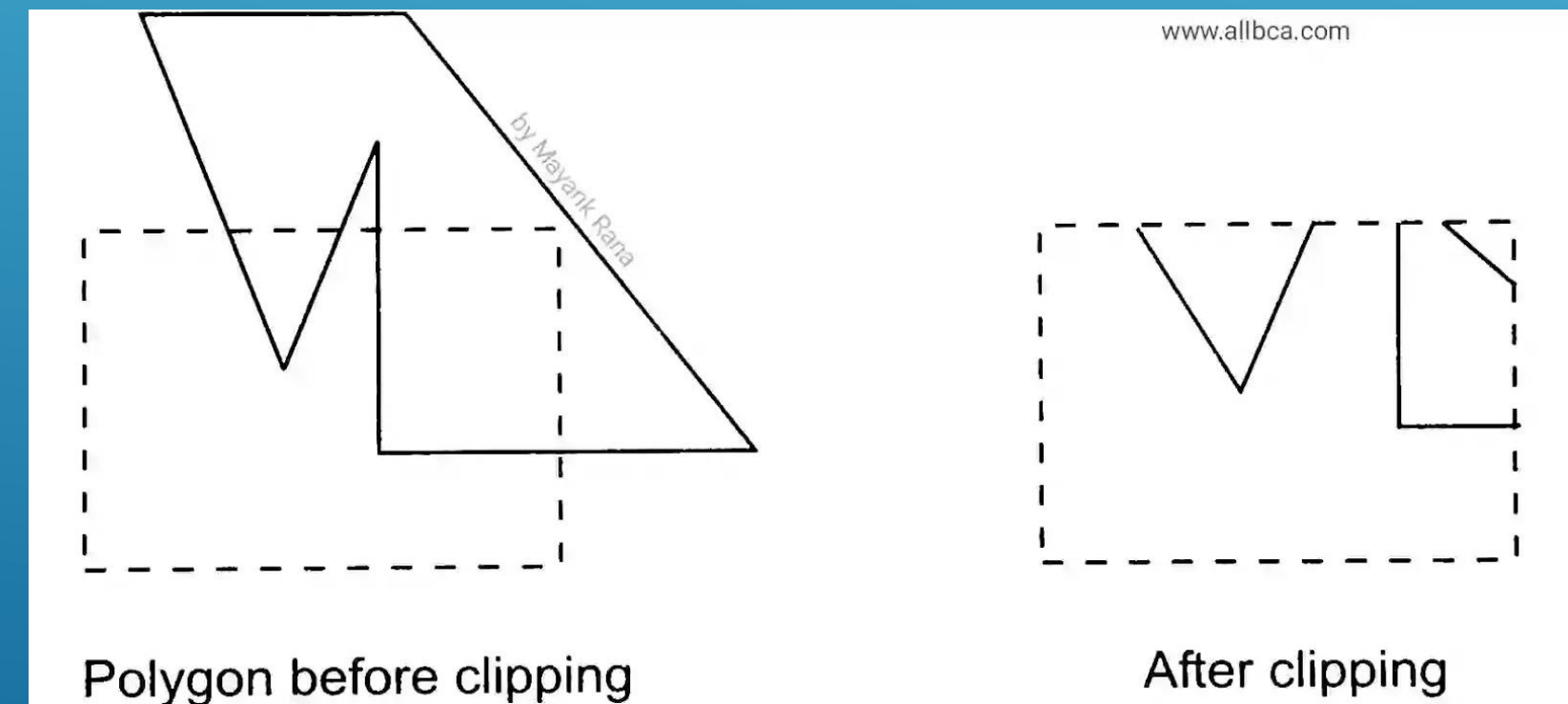
0000

Completely inside, so accept the line



POLYGON CLIPPING

- ❑ Sutherland Hodgeman polygon clipping algorithm is used for polygon clipping. In this algorithm, all the vertices of the polygon are clipped against each edge of the clipping window.
- ❑ Polygon clipping is the process of cutting off parts of a polygon that lie outside a given boundary
- ❑ The polygon clipping algorithm deals with four different clipping cases.
 - ❖ Left clip
 - ❖ Right clip
 - ❖ Top clip
 - ❖ Bottom clip



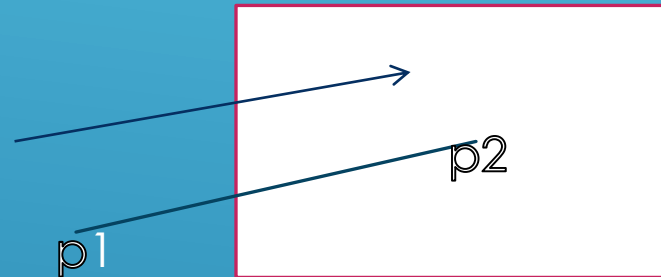


For each stage there are four cases to be checked for

Case 1:

if moving from outside to inside

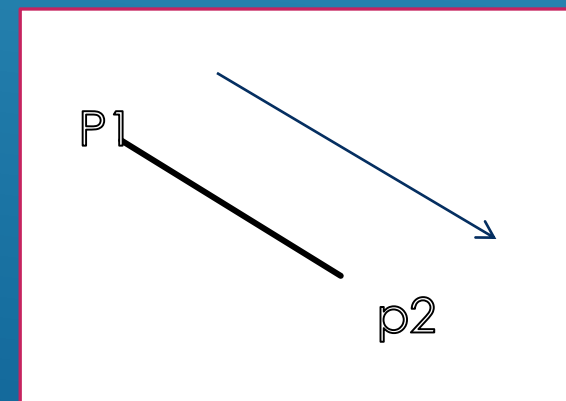
- Reject the start point and save the intersection point on wind boundary and vertex



Case 2:

if moving from inside to inside

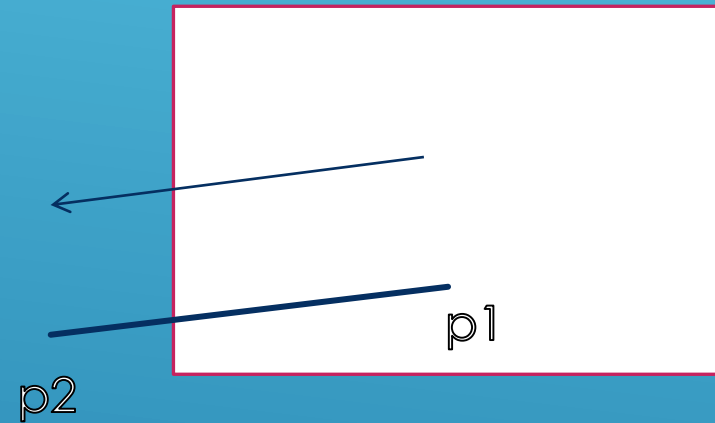
- Save the second vertex



Case 3:

if moving from inside to outside

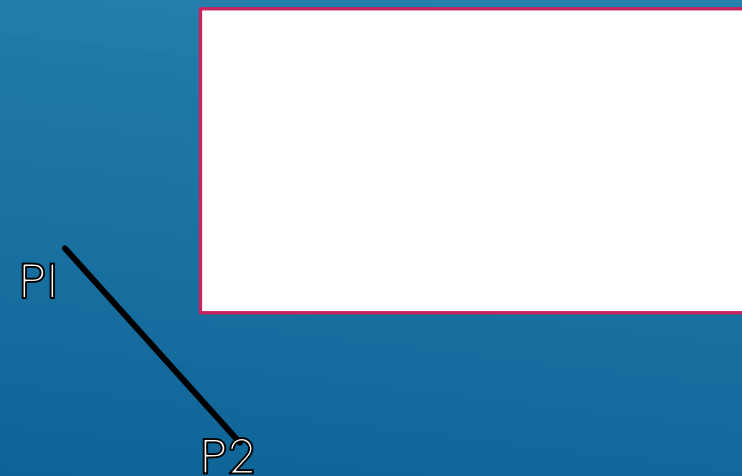
- Save intersection point and reject the end point

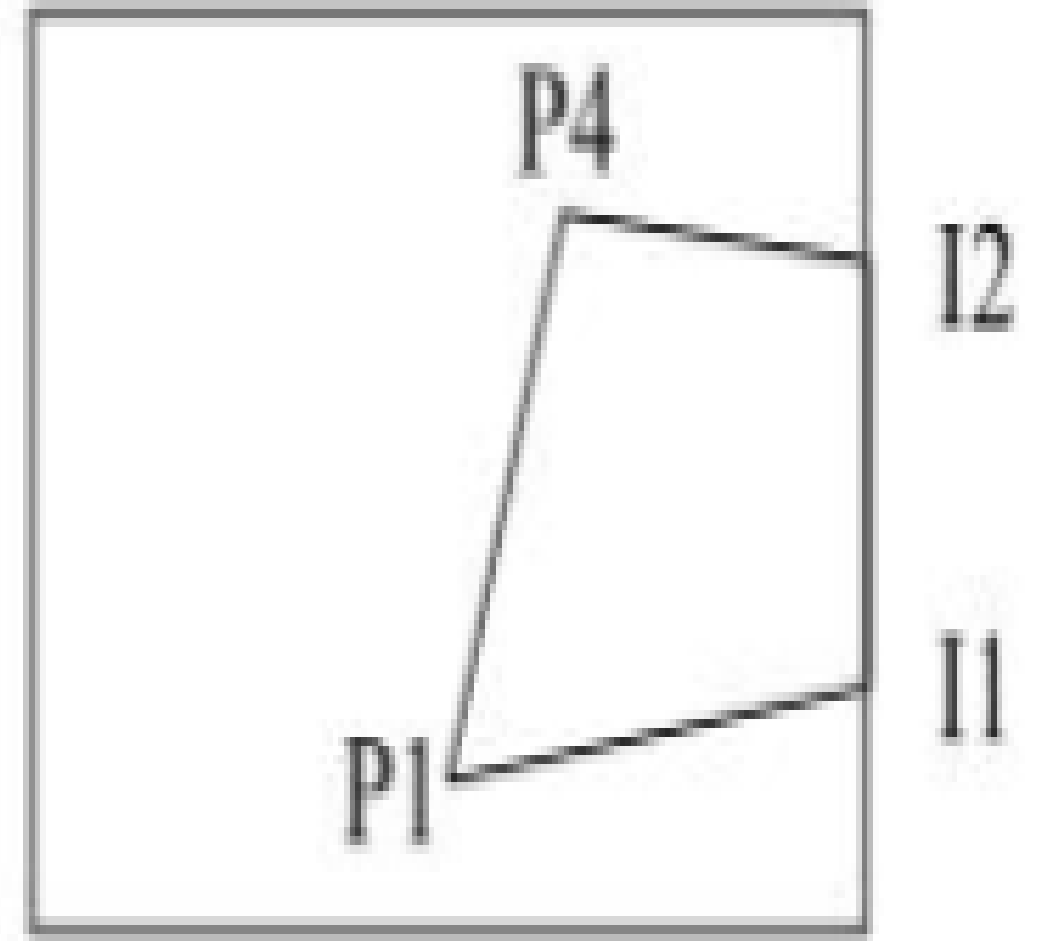
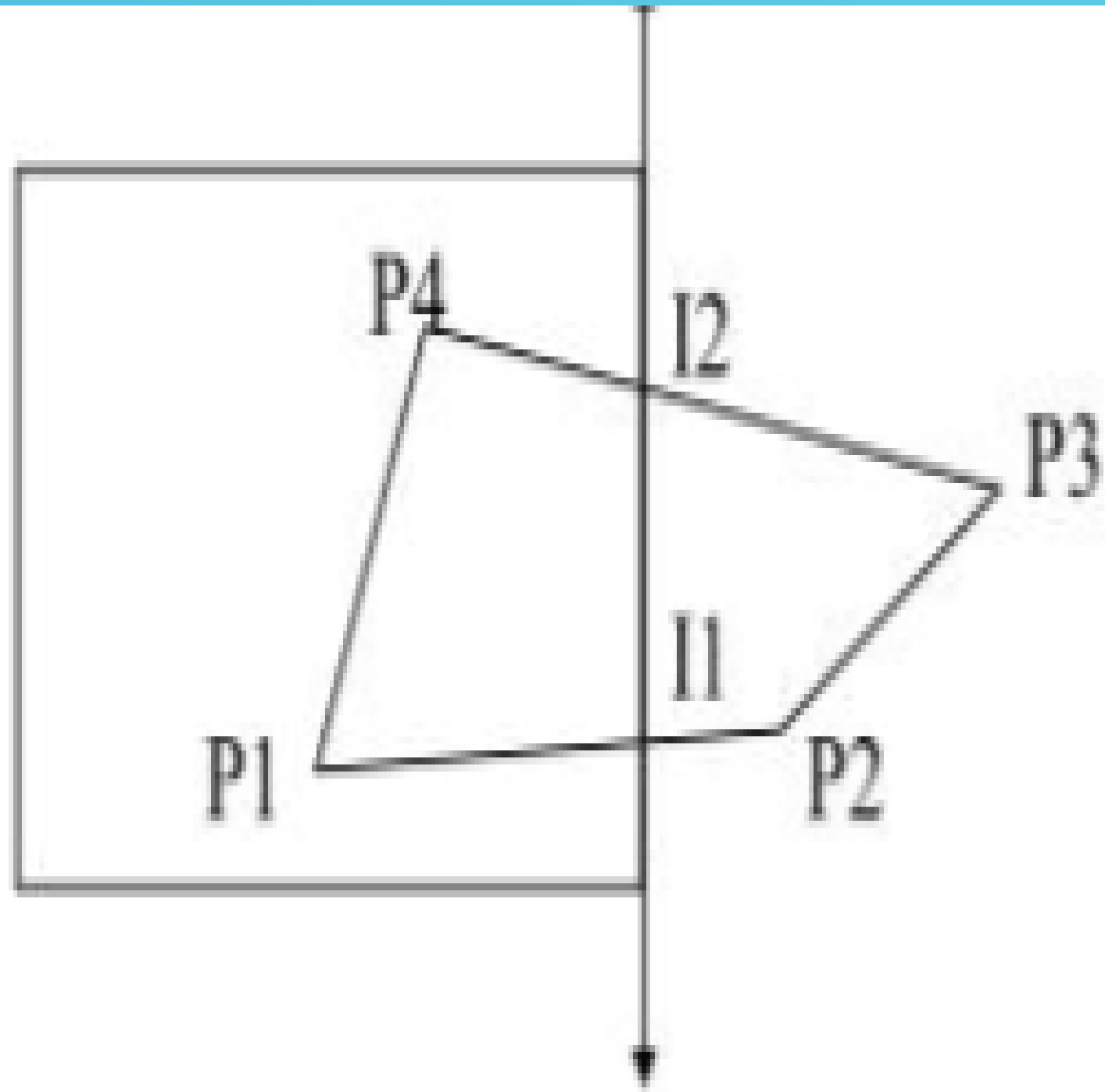


Case 4:

if moving from outside to outside

- Save none





CLIPPING ALGORITHMS FOR CIRCLES AND ARCS

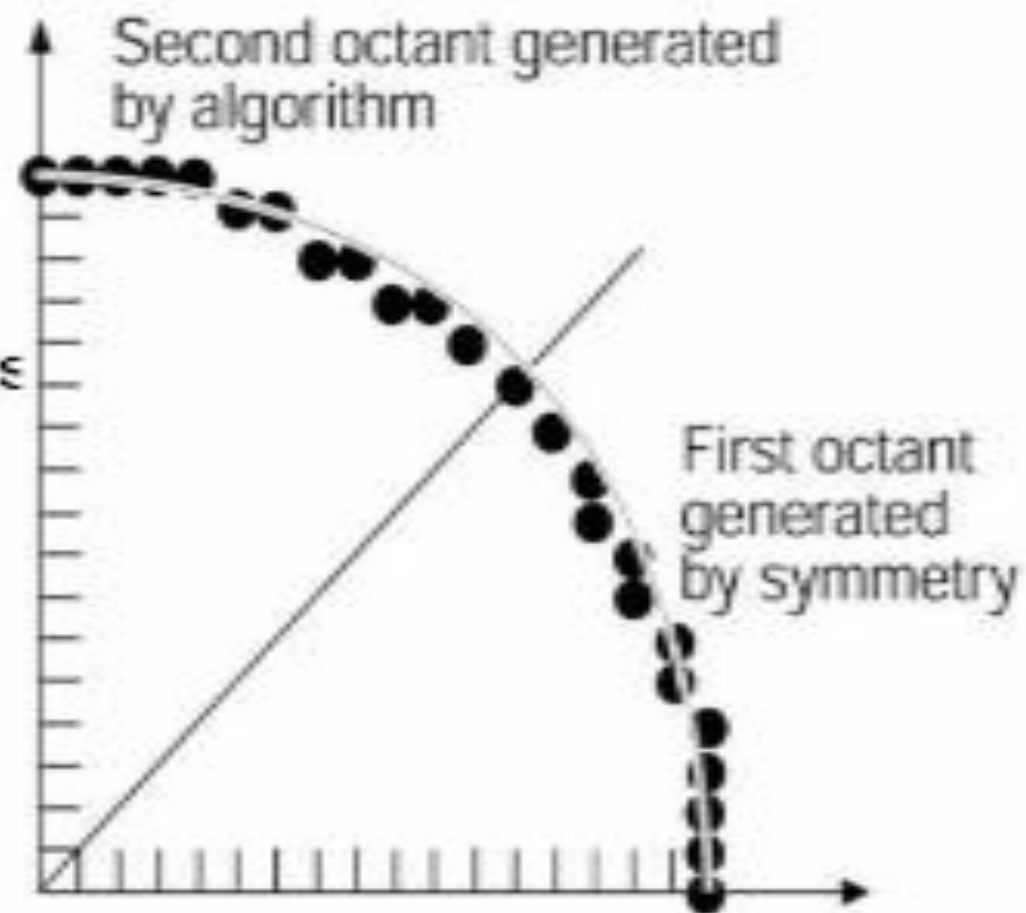
Clipping Circles

- Accept/Reject test
 - Does bounding box of the circle intersect with clipping box?
- If yes, condition pixel write on clipping box inside/outside test

Also we can test Circle points by Point Clipping .

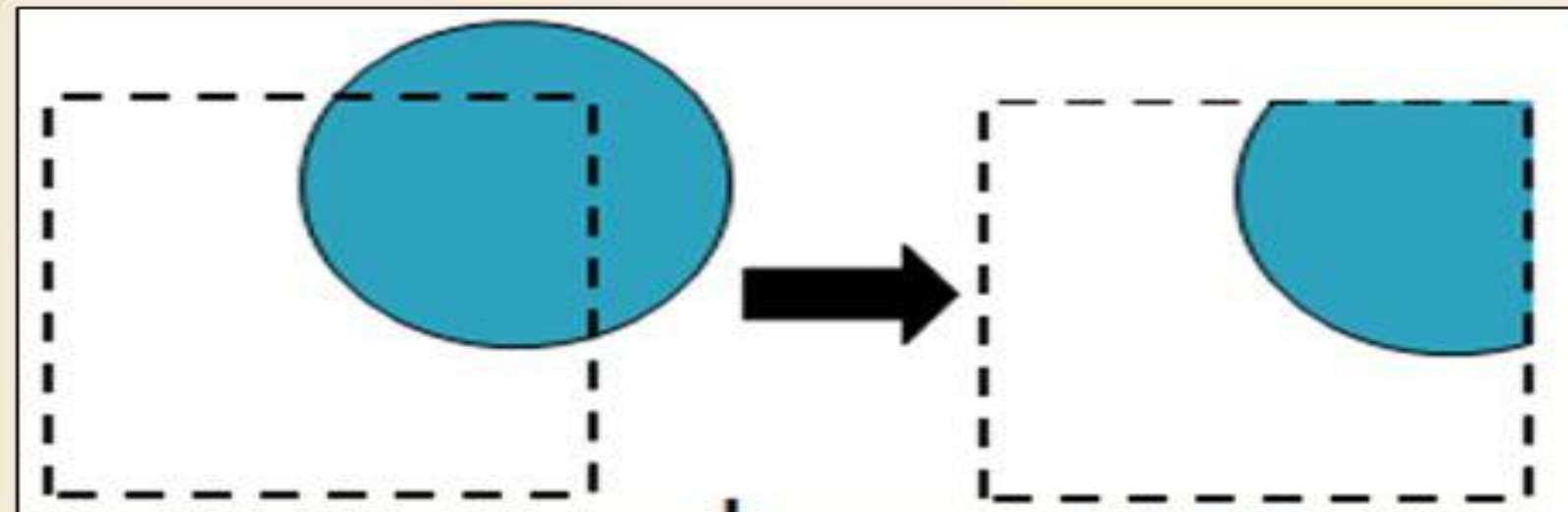
-the point $P=(x, y)$ is display in clipping Boundary if

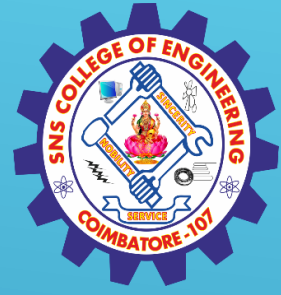
$$x_{min} \leq x \leq x_{max} \text{ and } y_{min} \leq y \leq y_{max}$$



CURVE CLIPPING

- Curve clipping procedures will involve non-linear equations (so requires more processing than for objects with linear boundaries In general, methods depend on how characters are represented).
- Clipping curves requires more work
 - For circles we must find the two intersection points on the window boundary





THANK YOU