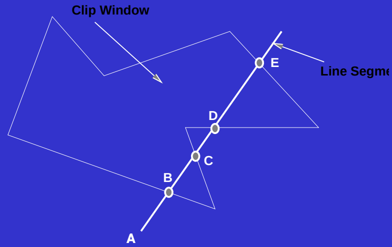
Line Clipping

Problem Definition

Given a clip region and a line segment, determine the sections of the line interior to the region.



Simplification

Clip region (window) is an upright rectangle.

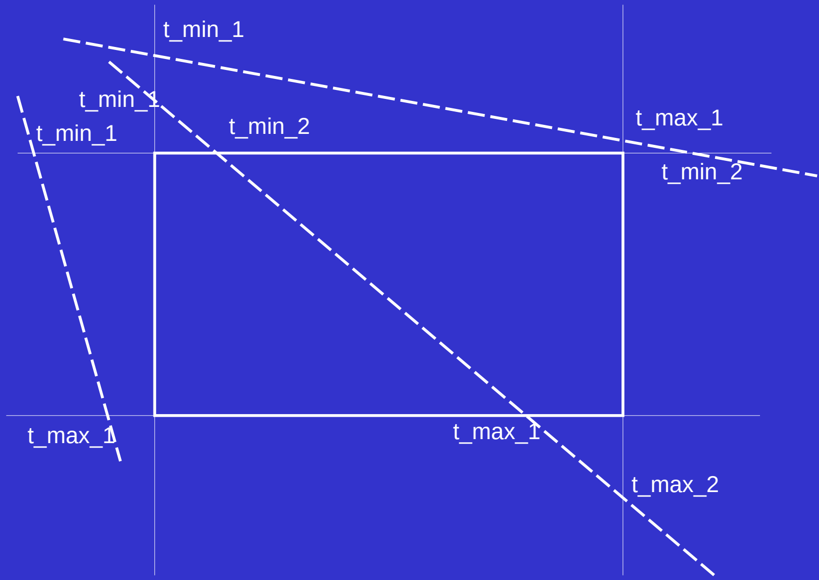
Line Clipping Alg: Strategy

if end points are inside clip region
    entire line segment is visible
else
    calculate intersections between line segment and clip region
    output segments interior to clip region
end

Note:

- we deal with line segments, not infinite lines.
- Slope-Intercept form ($y = mx + c$) of line equation is not convenient
- Parametric representation preferred.

Liang-Barsky Clipping: Idea



- Intersections between a line and the clipping boundaries are distinguished as entering or exiting the boundary (t_{enter} and t_{exit}.
- If $Max(t_{enter\,k}) < Min(t_{exit\,k})$ and the interval is a subset of $(0, 1.0)$, then the line segment intersects the interior of the clipping region.

Liang-Barsky Clipping: Details

A point (x, y) is in the interior if

$$\begin{aligned} xw_{min} &\leq x_0 + t\Delta x \leq xw_{max} \\ yw_{min} &\leq y_0 + t\Delta y \leq yw_{max} \end{aligned}$$

which can be rewritten as

$$tp_k \leq q_k, \quad k = 1, 2, 3, 4$$

$$\begin{aligned} p_1 &= -\Delta x, & q_1 &= x_0 - xw_{min} \\ p_2 &= \Delta x, & q_2 &= xw_{max} - x_0 \\ p_3 &= -\Delta y, & q_3 &= y_0 - yw_{min} \\ p_4 &= \Delta y, & q_4 &= yw_{max} - y_0 \end{aligned}$$

Liang-Barsky Clipping: Details

Intersection

$$t = \frac{q_k}{p_k}$$

- $p_k = 0$: line parallel to the kth clipping boundary; if $q_k < 0.0$, line is trivially rejected.
- $p_k < 0$: line proceeds from outside to inside across clipping boundary.
- $p_k > 0$: line proceeds from inside to outside across clipping boundary.

Algorithm

t_{min} = 0.0, t_{max} = 1.0

if ClipBoundary (p_1 , q_1 , &t_{min}, &t_{max})

    if ClipBoundary (p_2 , q_2 , &t_{min}, &t_{max})

        if ClipBoundary (p_3 , q_3 , &t_{min}, &t_{max})

            if ClipBoundary (p_4 , q_4 , &t_{min}, &t_{max})

            {

                if ($t_{min} > 0.0$) {

                    pt1.x = pt1.x + t_{min} * dx

                    pt1.y = pt1.y + t_{min} * dy

                }

                if ($t_{min} < 1.0$) {

                    pt2.x = pt2.x + t_{max} * dx

                    pt2.y = pt2.y + t_{max} * dy

                }

            }

    “draw line from pt1 to p2”

}