Liang Barsky Algorithm for 2-D Line Clipping

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Outline

- Liang Barsky Algorithm
- Summary

Liang Barsky Algorithm

5.3.2.4 Liang-Barsky Line Clipping Algorithm

In the last section we have seen Cyrus-Beck line clipping algorithm using parametric equations. It is more efficient than Cohen-Sutherland algorithm. Liang and Barsky have developed even more efficient algorithm than Cyrus-Beck algorithm using parametric equations. These parametric equations are given as

$$x = x_1 + t\Delta x$$

$$y = y_1 + t\Delta y, \qquad 0 \le t \le 1$$

where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$

The point clipping conditions (Refer section 5.3.1) for Liang-Barsky approach in the parametric form can be given as

$$x_{wmin} \le x_1 + t\Delta x \le x_{wmax} \text{ and}$$

$$y_{wmin} \le y_1 + t\Delta y \le y_{wmax}$$

Liang-Barsky express these four inequalities with two parameters p and q as follows :

$$tp_i \le q_i \qquad i = 1, 2, 3, 4$$

where parameters p and q are defined as

$$p_1 = -\Delta x, \quad q_1 = x_1 - x_{wmin}$$

$$p_2 = \Delta x, \quad q_2 = x_{wmax} - x_1$$

$$p_3 = -\Delta y, \quad q_3 = y_1 - y_{wmin}$$

$$p_4 = \Delta y, \quad q_4 = y_{wmax} - y_1$$

Liang Barsky Algorithm(Cont.....)

Following observations can be easily made from above definitions of parameters p and q.

If $p_1 = 0$	:	Line is parallel to left clipping boundary.
If $p_2 = 0$	:	Line is parallel to right clipping boundary.
If $p_3 = 0$	:	Line is parallel to bottom clipping boundary.
If $p_4 = 0$	:	Line is parallel to top clipping boundary.
If $p_i = 0$,		and for that value of i,
If $q_i < 0$	:	Line is completely outside the boundary and can be eliminated.
If $q_i \geq 0$	:	Line is inside the clipping boundary.
If $p_i < 0$	:	Line proceeds from outside to inside of the clipping boundary.
If $p_i > 0$	:	Line proceeds from inside to outside of the clipping boundary.

Therefore, for nonzero value of p_i, the line crosses the clipping boundary and we have to find parameter t. The parameter t for any clipping boundary i can be given as

$$t = \frac{q_i}{p_i} \qquad i = 1, 2, 3, 4$$

Liang-Barsky algorithm calculates two values of parameter t : t_1 and t_2 that define that part of the line that lies within the clip rectangle. The value of t_1 is determined by checking

Liang Barsky Algorithm(Cont.....)

the rectangle edges for which the line proceeds from the outside to the inside ($p < 0$). The value of t_1 is taken as a largest value amongst various values of intersections with all edges. On the other hand, the value of t_2 is determined by checking the rectangle edges for which the line proceeds from the inside to the outside ($p > 0$). The minimum of the calculated value is taken as a value for t_2.

Now, if $t_1 > t_2$, the line is completely outside the clipping window and it can be rejected. Otherwise the values of t_1 and t_2 are substituted in the parametric equations to get the end points of the clipped line.

Algorithm

1. Read two endpoints of the line say p_1 (x_1, y_1) and p_2 (x_2, y_2).
2. Read two corners (left-top and right-bottom) of the window, say $(x_{wmin}, y_{wmax}, x_{wmax}, y_{wmin})$
3. Calculate the values of parameters p_i and q_i for $i = 1, 2, 3, 4$ such that

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{wmin}$$
$$p_2 = \Delta x \qquad q_2 = x_{wmax} - x_1$$
$$q_1 = -\Delta y \qquad q_3 = y_1 - y_{wmin}$$
$$q_2 = \Delta y \qquad q_4 = y_{wmax} - y_1$$

Liang Barsky Algorithm(Cont.....)

4. if $p_i = 0$, then

{ The line is parallel to i^{th} boundary.

Now, if $q_i < 0$ then

{ line is completely outside the boundary, hence discard the line segment and goto stop.

}

else

{ Check whether the line is horizontal or vertical and accordingly check the line endpoint with corresponding boundaries. If line endpoint/s lie within the bounded area then use them to draw line otherwise use boundary coordinates to draw line. Go to stop.

}

}

5. Initialise values for t_1 and t_2 as
   $t_1 = 0$ and $t_2 = 1$
6. Calculate values for q_i/p_i for $i = 1, 2, 3, 4$
7. Select values of q_i/p_i where $p_i < 0$ and assign maximum out of them as t_1.
8. Select values of q_i/p_i where $p_i > 0$ and assign minimum out of them as t_2.

Liang Barsky Algorithm(Cont.....)

9. If $(t_1 < t_2)$

{ Calculate the endpoints of the clipped line as follows :

$xx_1 = x_1 + t_1 \Delta x$

$xx_2 = x_1 + t_2 \Delta x$

$yy_1 = y_1 + t_1 \Delta y$

$yy_2 = y_1 + t_2 \Delta y$

Draw line (xx_1, yy_1, xx_2, yy_2)

}

10. Stop.

Advantages

1. It is more efficient than Cohen-Sutherland algorithm, since intersection calculations are reduced.
2. It requires only one division to update parameters t_1 and t_2.
3. Window intersections of the line are computed only once.

Ex. 5.5 *Find the clipping coordinates for a line p_1p_2 where $p_1 = (10, 10)$ and p_2 $(60, 30)$, against window with $(x_{wmin}, y_{wmin}) = (15, 15)$ and $(x_{wmax}, y_{wmax}) = (25, 25)$.*

Sol. : Here,

$x_1 = 10 \qquad x_{wmin} = 15$

$y_1 = 10 \qquad y_{wmin} = 15$

Liang Barsky Algorithm(Cont.....)

$x_2 = 60$ $\quad x_{wmax} = 25$

$y_2 = 30$ $\quad y_{wmax} = 25$

$p_1 = -50$ $\quad q_1 = -5$ $\quad p_1/q_1 = 0.1$

$p_2 = 50$ $\quad q_2 = 15$ $\quad p_2/q_2 = 0.3$

$p_3 = -20$ $\quad q_3 = -5$ $\quad p_3/q_3 = 0.25$

$p_4 = 20$ $\quad q_4 = 15$ $\quad p_4/q_4 = 0.75$

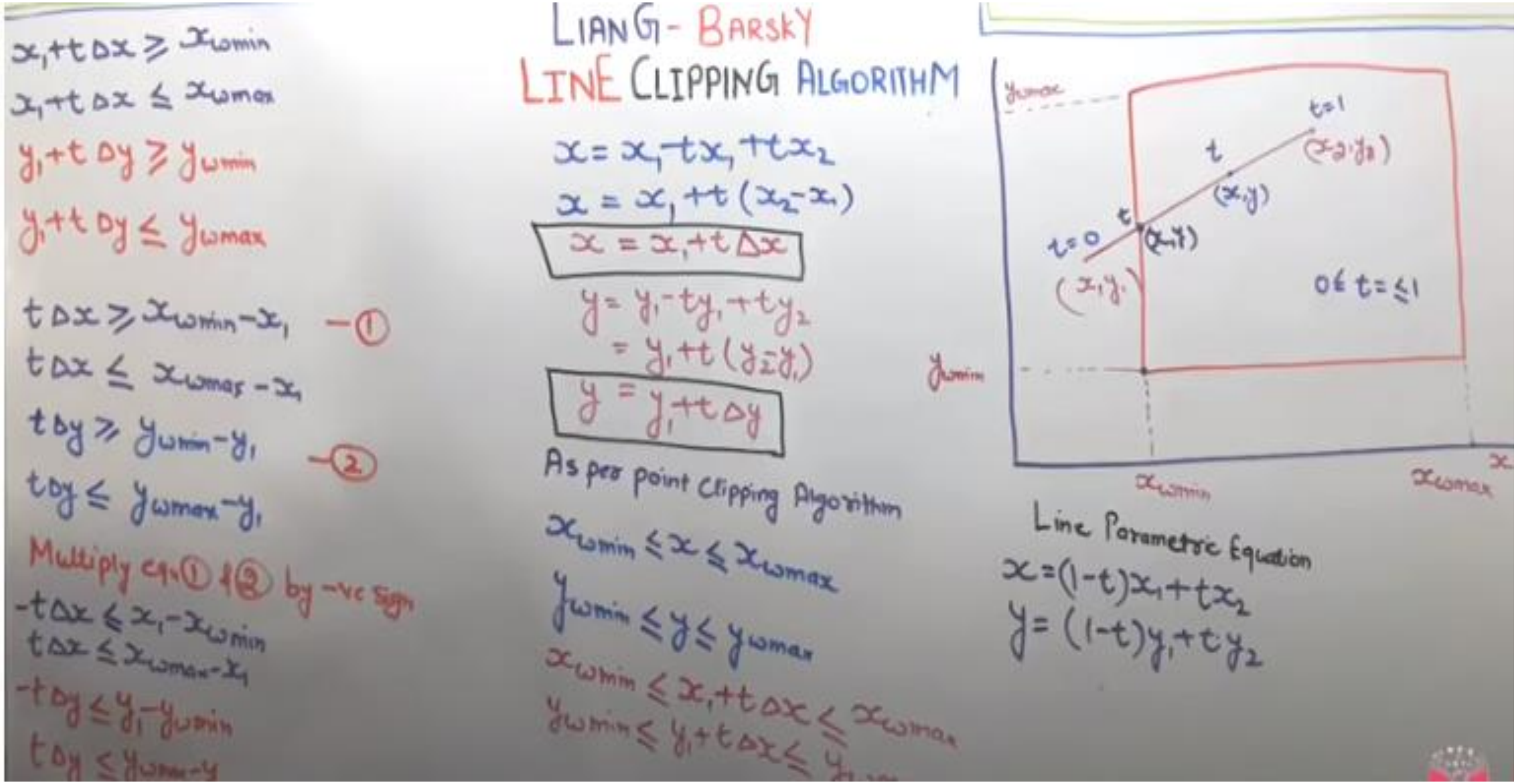
$t_1 = \max(0.25, 0.1) = 0.25$ since for these values $p < 0$

$t_2 = \min(0.3, 0.75) = 0.3$ since for these values $p > 0$

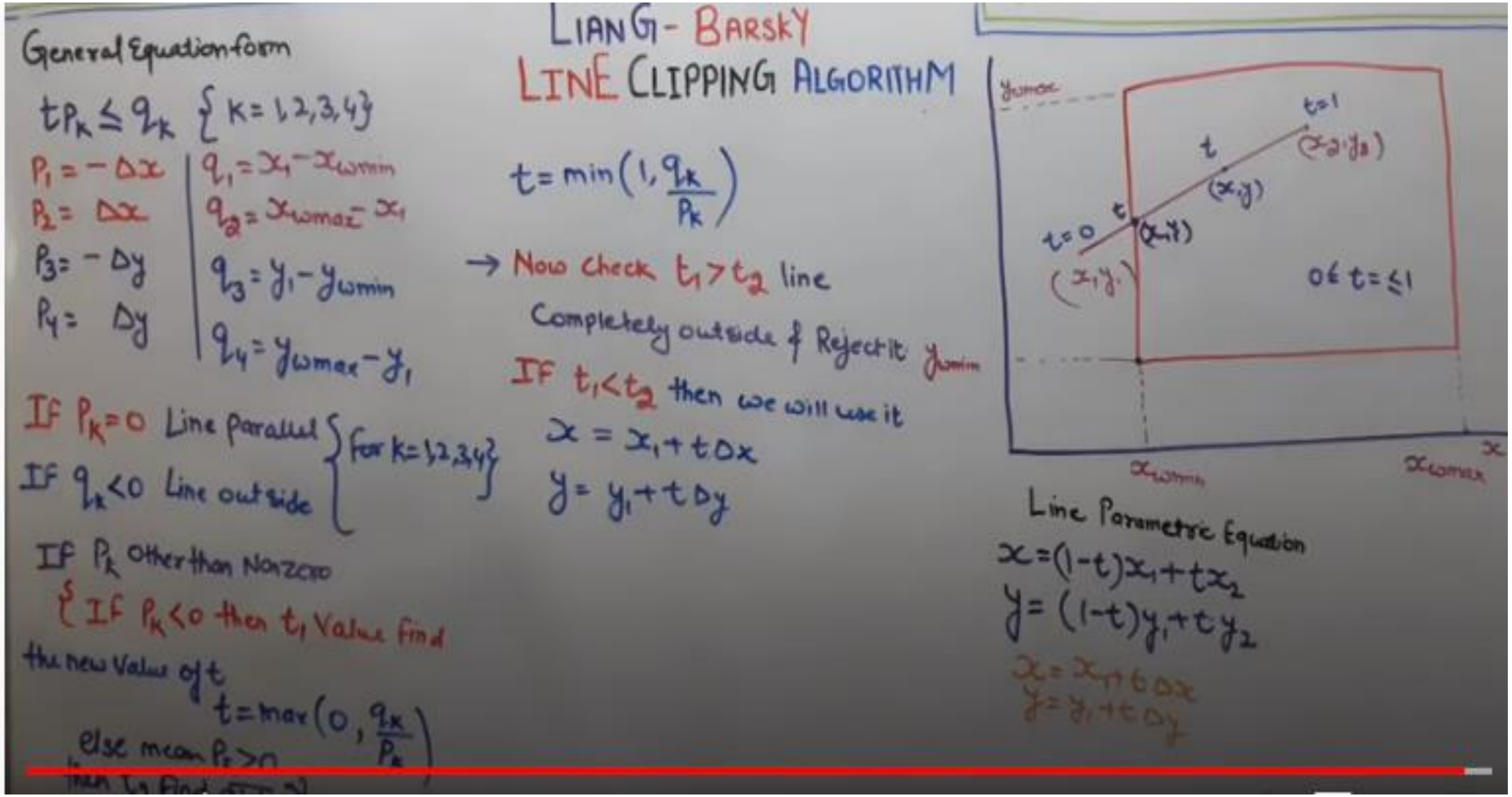
Here, $t_1 < t_2$ and the endpoints of clipped line are :

$$\begin{aligned} xx_1 &= x_1 + t_1 \Delta x \\ &= 10 + 0.25 \times 50 \\ &= 22.5 \\ yy_2 &= y_1 + t_1 \Delta y \\ &= 10 + 0.25 \times 20 \\ &= 15 \\ xx_2 &= x_1 + t_2 \Delta x \\ &= 10 + 0.3 \times 50 \\ &= 25 \end{aligned}$$

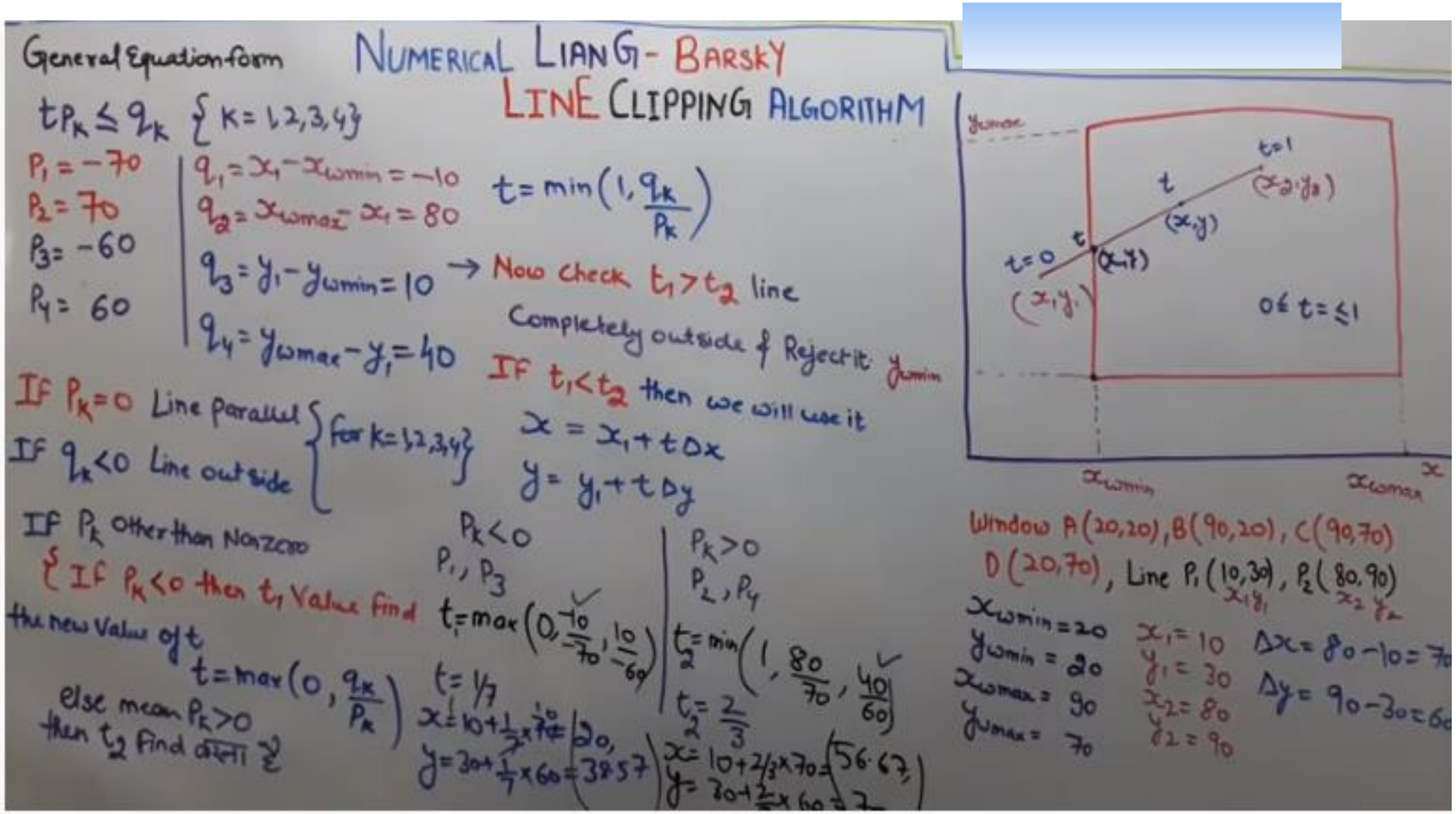
Liang Barsky Algorithm(Cont.....)



Liang Barsky Algorithm(Cont.....)



Liang Barsky Algorithm(Cont.....)



Summary

- It is more efficient than cohen-sutherland algorithm,since intersection calculations are reduced
- Require one division to update parameters t1 and t2

Resources

- https:// iq.opengenus.org/liang-basrky-line-clipping-algorithm/
- https:// https://www.tutorialandexample.com/line-clipping/
- https://www.cs.helsinki.fi/group/goa/viewing/leikkaus/lineClip.html