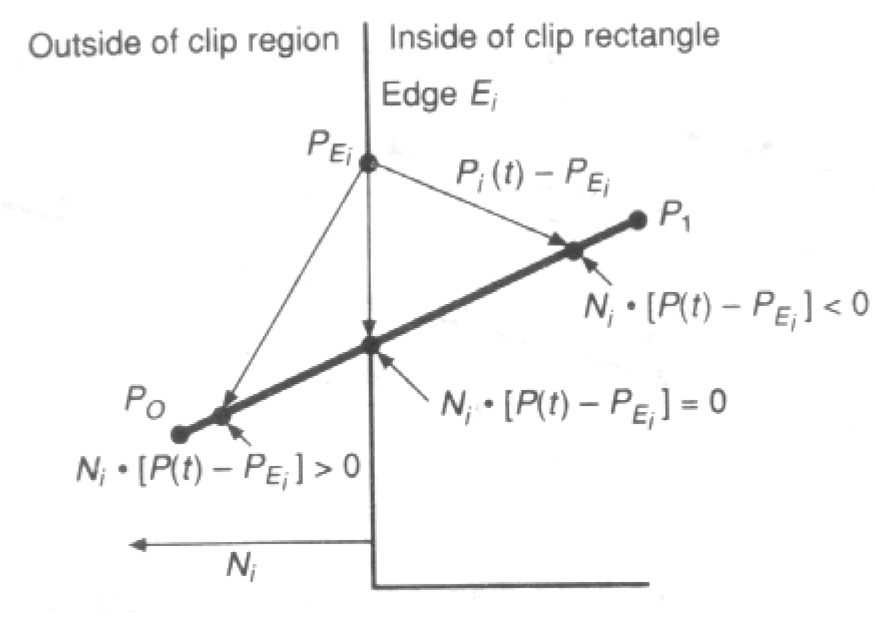
Parametric Line-Clipping Algorithm

- Introduced by Cyrud and Beck in 1978
- Efficiently improved by Liang and Barsky
- Essentially find the parameter t from $P(t) = P_0 + (P_1\text{-}P_0)t$



$$N_i \square [P(t) - P_{Ei}] = 0$$

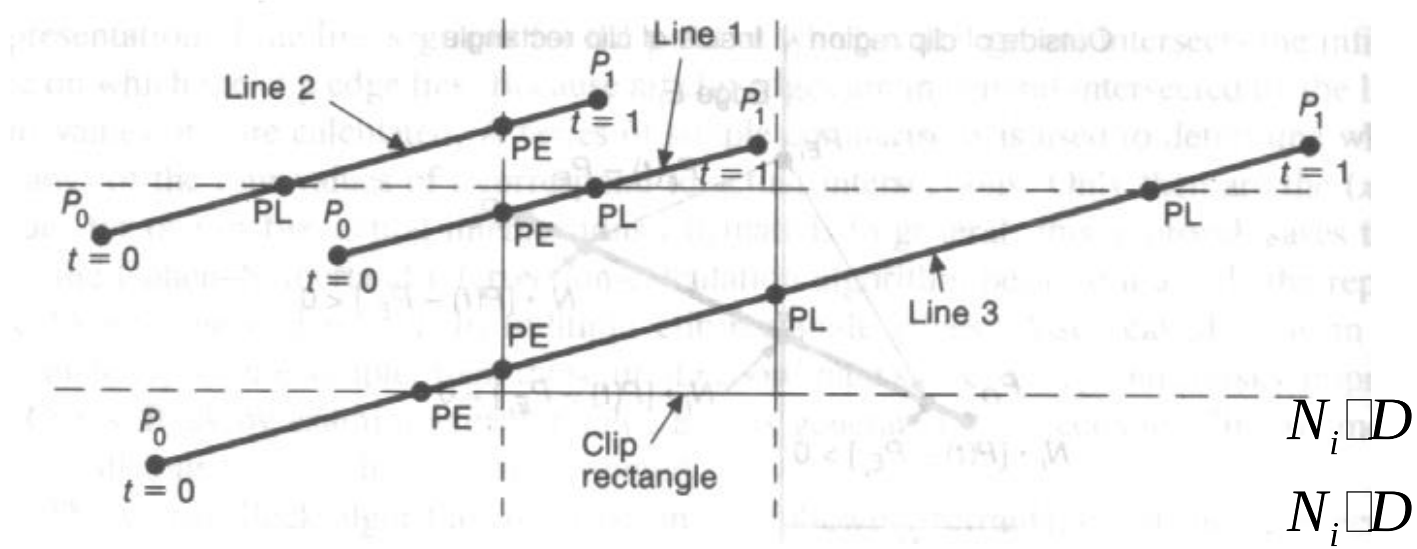
$$N_i \square [P_0 + (P_1 - P_0)t - P_{Ei}] = 0$$

$$N_i \square [P_0 - P_{Ei}] + N_i \square [P_1 - P_0]t = 0$$

$$t = -\frac{N_i \square [P_0 - P_{Ei}]}{N_i \square D}$$

$$where\, D = (P_1 - P_0)$$

Parametric Line-Clipping Algorithm (cont.)



$$N_i \cdot D < 0 => PE(angle > 90)$$

$$N_i \cdot D > 0 => PL(angle < 90)$$

- Formally, intersections can be classified as PE (potentially entering) and PL (potentially leaving) on the basis of the angle between P_0P_1 and N_i
- Determine t_E or t_L for each intersection
- Select the line segment that has maximum t_E and minimum t_L
- If $t_E > t_L$, then trivially rejected

Parametric Line-Clipping Algorithm (cont.)

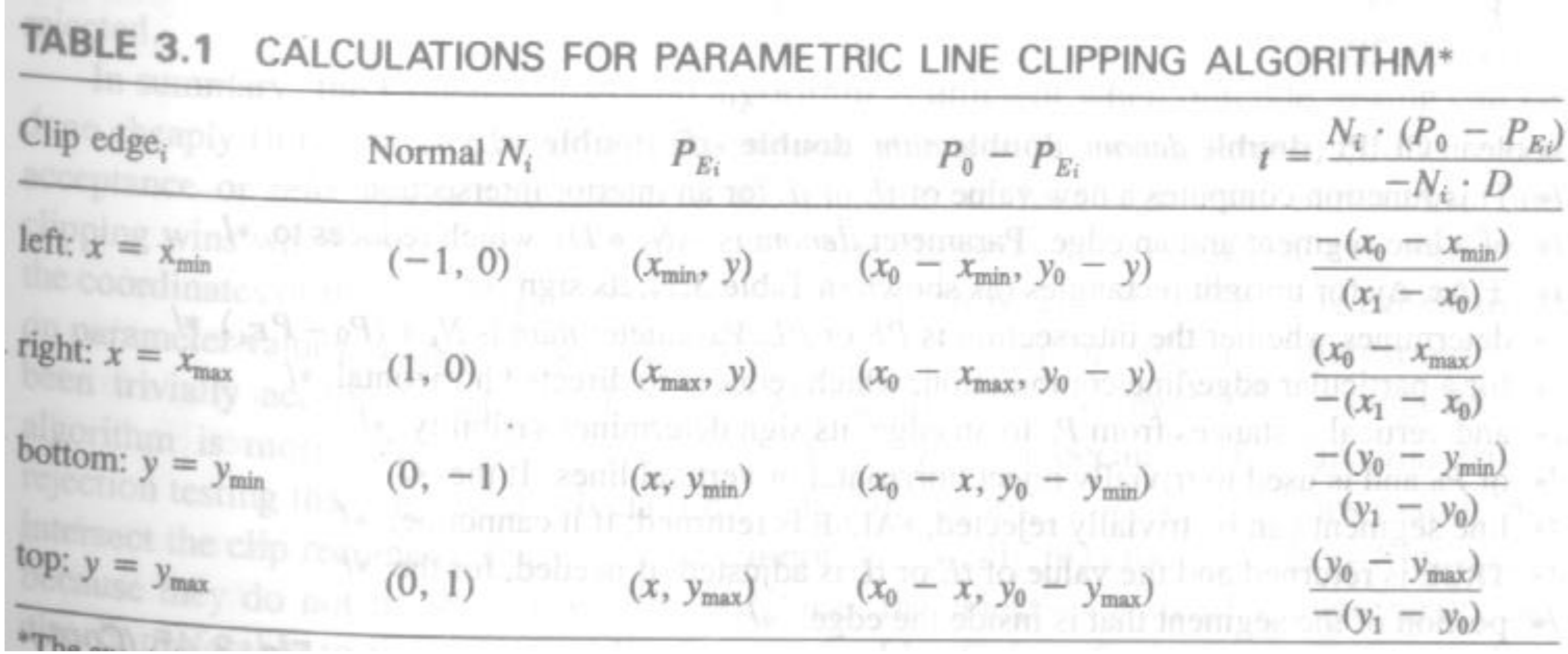
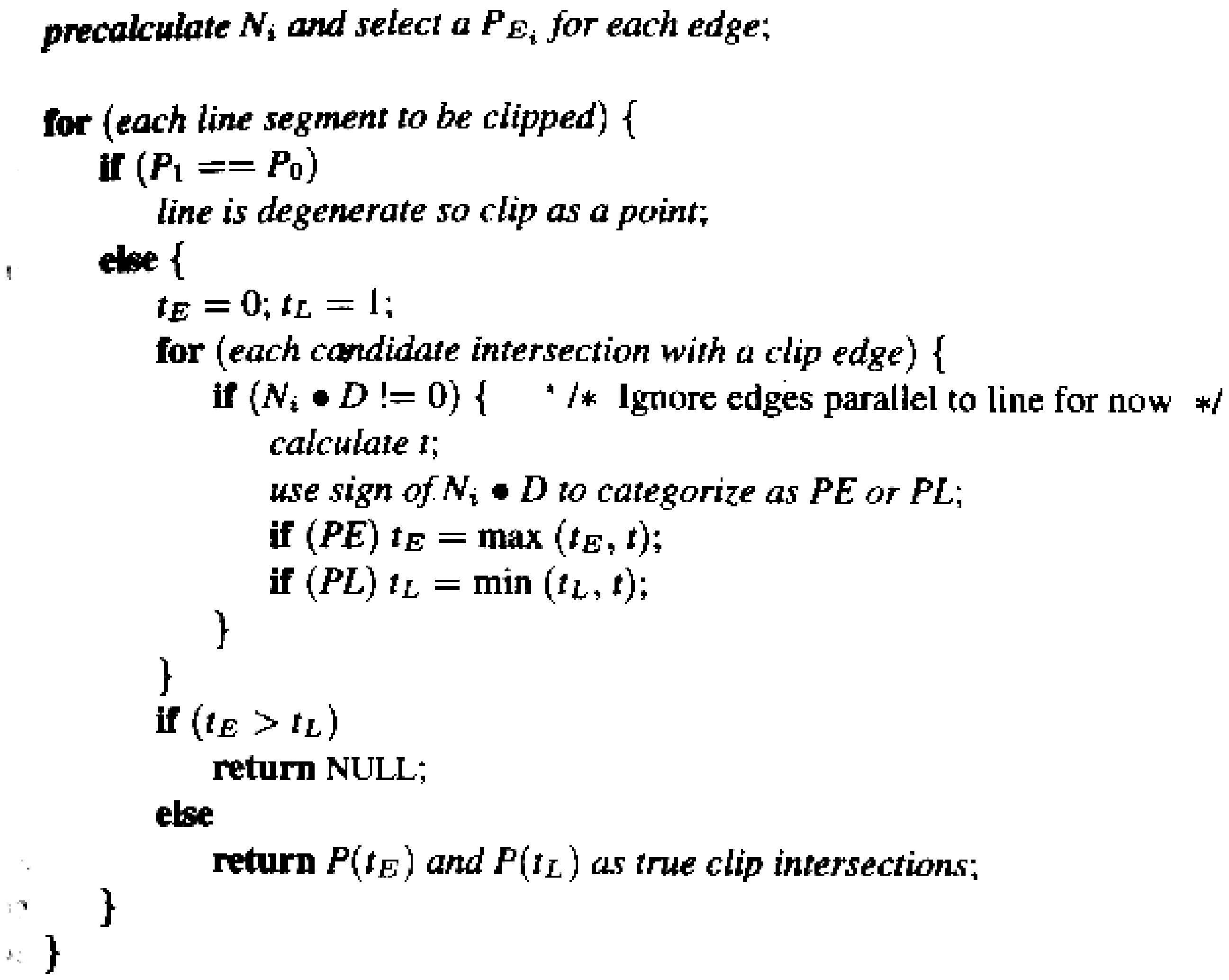


TABLE 3.1 CALCULATIONS FOR PARAMETRIC LINE CLIPPING ALGORITHM*

Clip edge$_i$	Normal N_i	P_{E_i}	$P_0 - P_{E_i}$	$t = \frac{N_i \cdot (P_0 - P_{Ei})}{-N_i \cdot D}$
left: $x = x_{\min}$	$(-1, 0)$	$(x_{\min}, y)$	$(x_0 - x_{\min}, y_0 - y)$	$\frac{-(x_0 - x_{\min})}{(x_1 - x_0)}$
right: $x = x_{\max}$	$(1, 0)$	$(x_{\max}, y)$	$(x_0 - x_{\max}, y_0 - y)$	$\frac{(x_0 - x_{\max})}{-(x_1 - x_0)}$
bottom: $y = y_{\min}$	$(0, -1)$	$(x, y_{\min})$	$(x_0 - x, y_0 - y_{\min})$	$\frac{-(y_0 - y_{\min})}{(y_1 - y_0)}$
top: $y = y_{\max}$	$(0, 1)$	$(x, y_{\max})$	$(x_0 - x, y_0 - y_{\max})$	$\frac{(y_0 - y_{\max})}{-(y_1 - y_0)}$

Cyrus-Beck Algorithm (Pseudocode)



```
precalculate $N_i$ and select a $P_{E_i}$ for each edge;

for (each line segment to be clipped) {
    if ($P_1 == P_0$)
        line is degenerate so clip as a point;
    else {
        $t_E = 0$; $t_L = 1$;
        for (each candidate intersection with a clip edge) {
            if ($N_i \bullet D$ != 0) {      /* Ignore edges parallel to line for now */
                calculate $t$;
                use sign of $N_i \bullet D$ to categorize as PE or PL;
                if (PE) $t_E = \max(t_E, t)$;
                if (PL) $t_L = \min(t_L, t)$;
            }
        }
        if ($t_E > t_L$)
            return NULL;
        else
            return $P(t_E)$ and $P(t_L)$ as true clip intersections;
    }
}
```