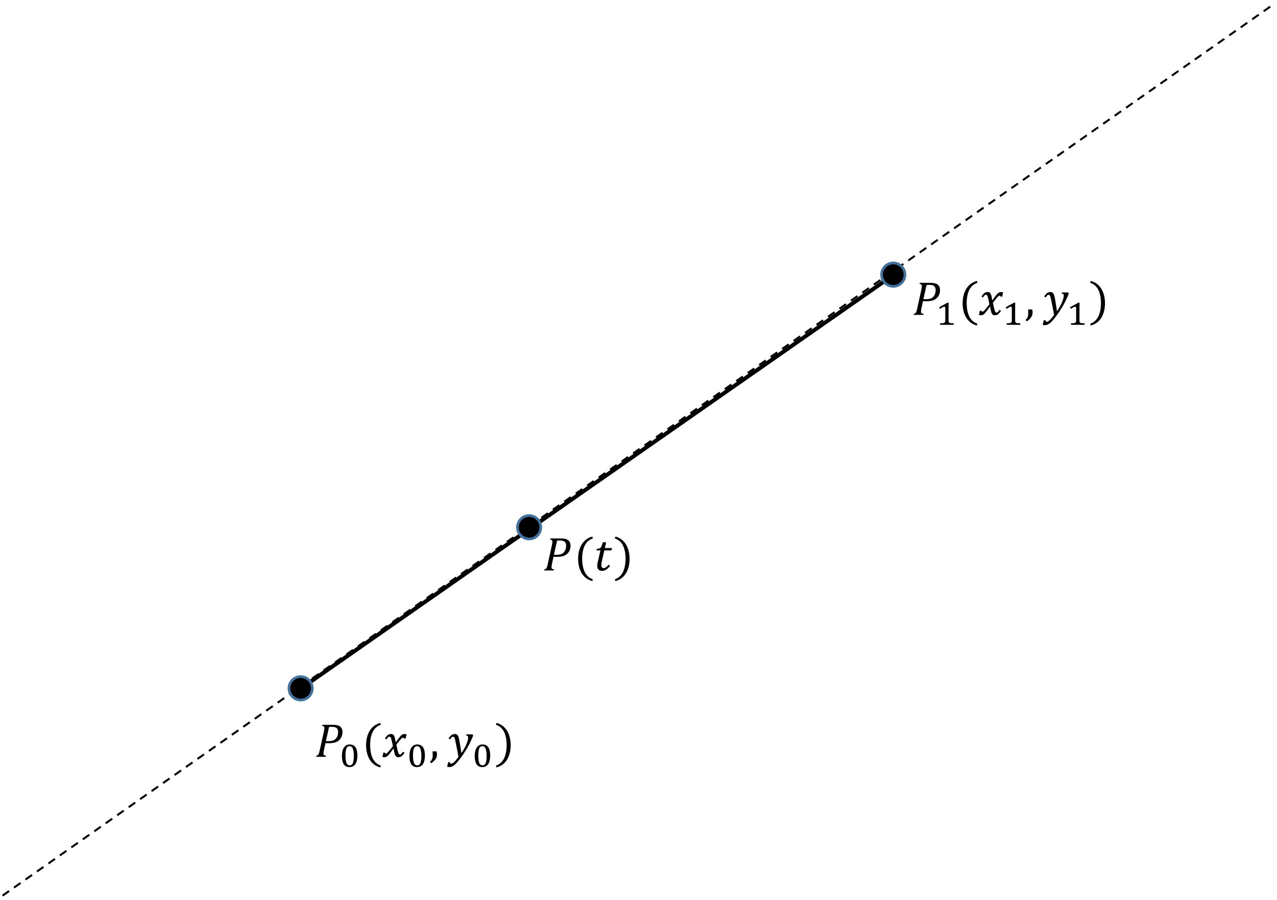
Clipping

Cyrus-Beck Algo

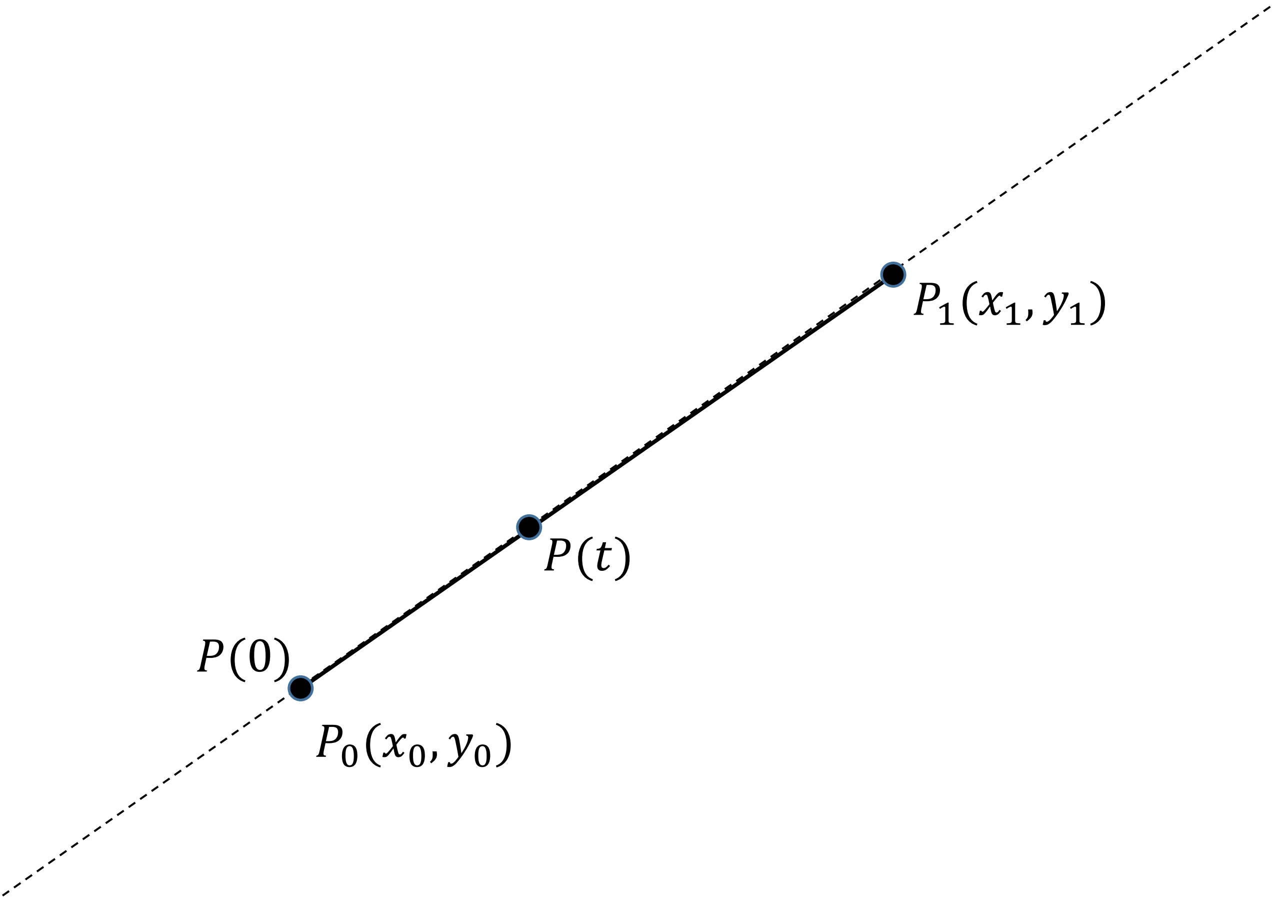
Parametric equation of Line



Parametric equation,

$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0) \\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\
&= (x_0 + t(x_1 - x_0), y_0 + t(y_1 - y_0))
\end{aligned}
$$

Parametric equation of Line

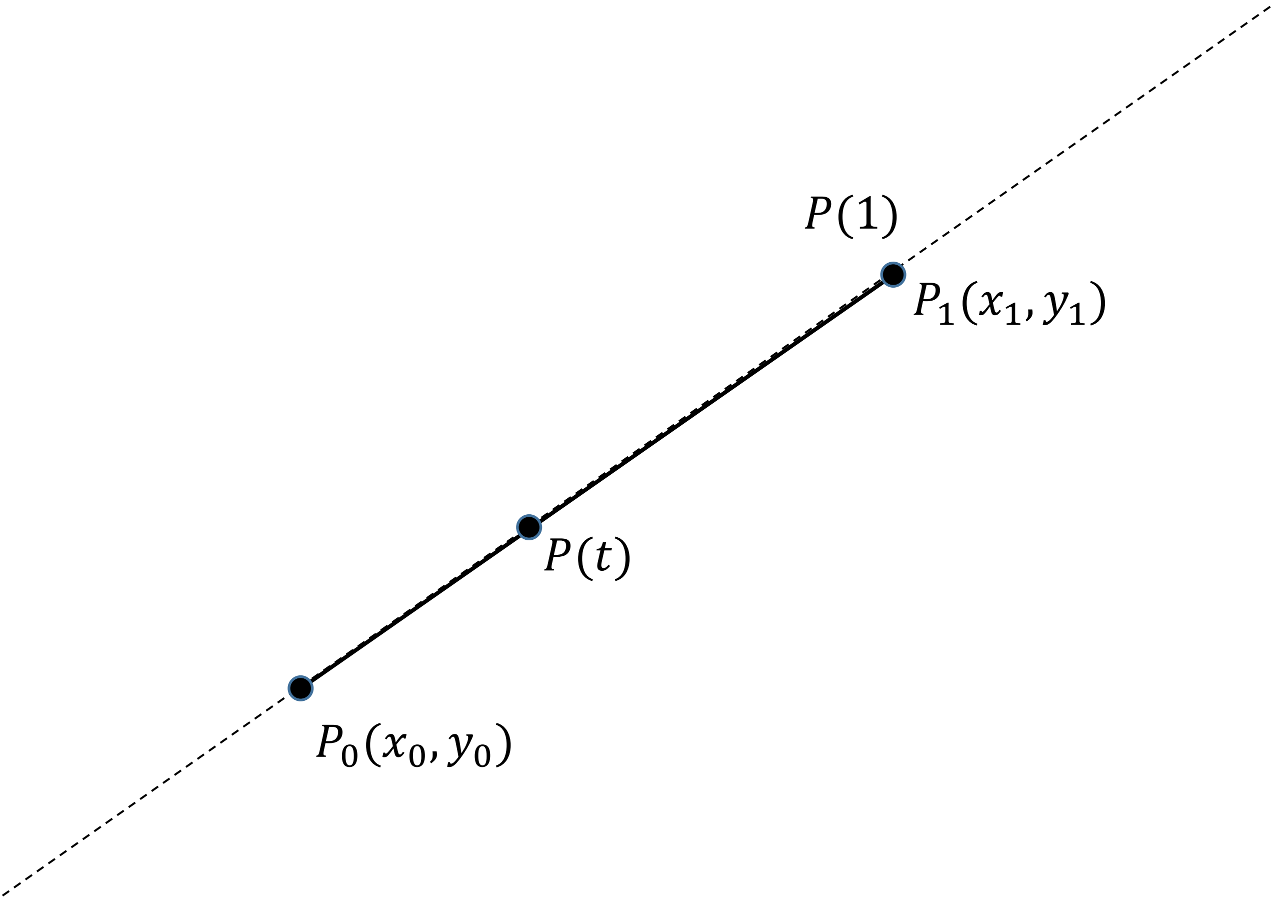


Parametric equation,

$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0) \\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\
&= (x_0 + t(x_1 - x_0), y_0 + t(y_1 - y_0))
\end{aligned}
$$

$$
\begin{aligned}
P(0) &= (x_0, y_0) + 0.(x_1 - x_0, y_1 - y_0) \\
&= (x_0, y_0)
\end{aligned}
$$

Parametric equation of Line

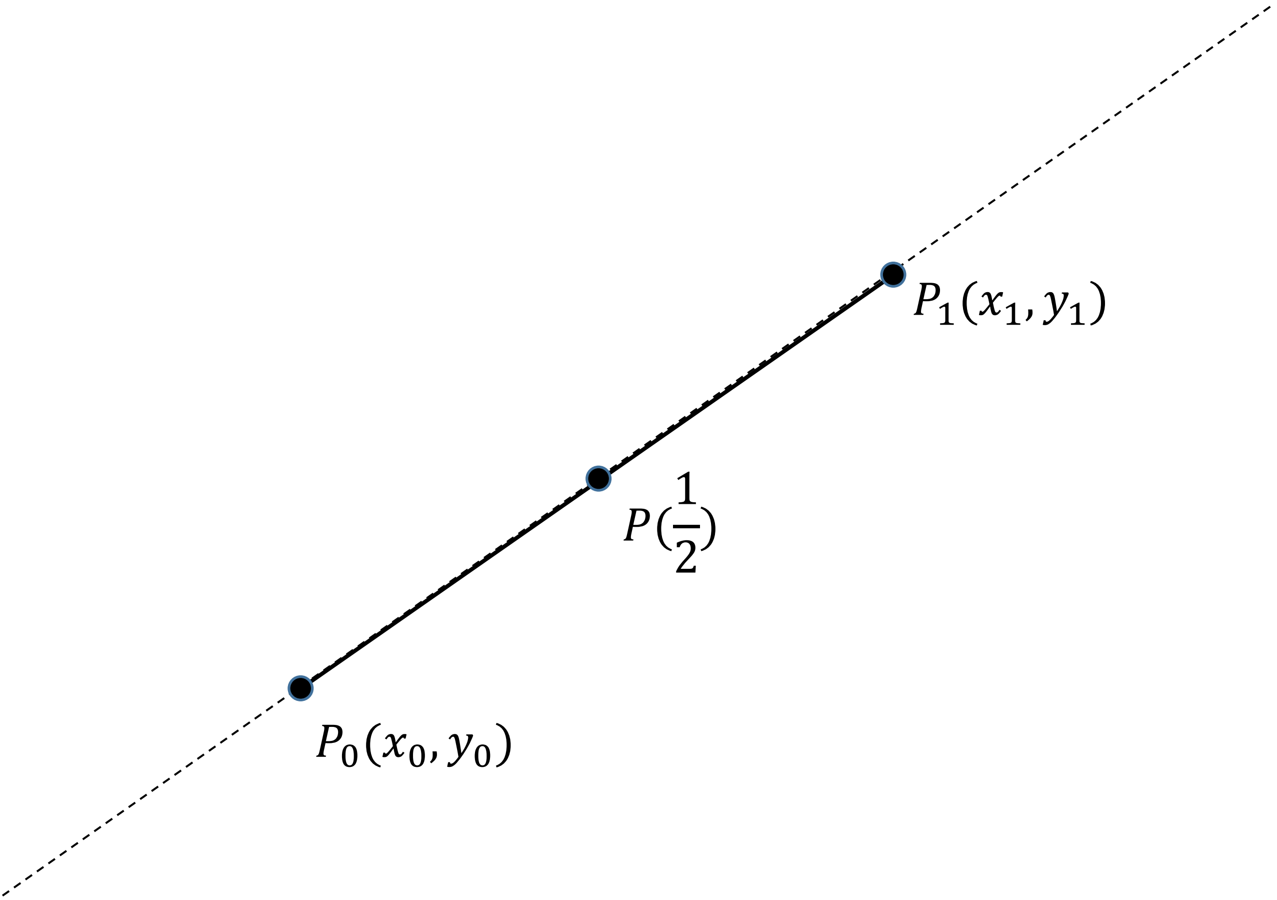


Parametric equation,

$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0) \\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\
&= (x_0 + t(x_1 - x_0), y_0 + t(y_1 - y_0))
\end{aligned}
$$

$$
\begin{aligned}
P(1) &= (x_0, y_0) + 1.(x_1 - x_0, y_1 - y_0) \\
&= (x_0, y_0) + (\mathrm{x}_1 - \mathrm{x}_0, y_1 - y_0) \\
&= (x_0 + x_1 - x_0, y_0 + y_1 - y_0) \\
&= (x_1, y_1)
\end{aligned}
$$

Parametric equation of Line



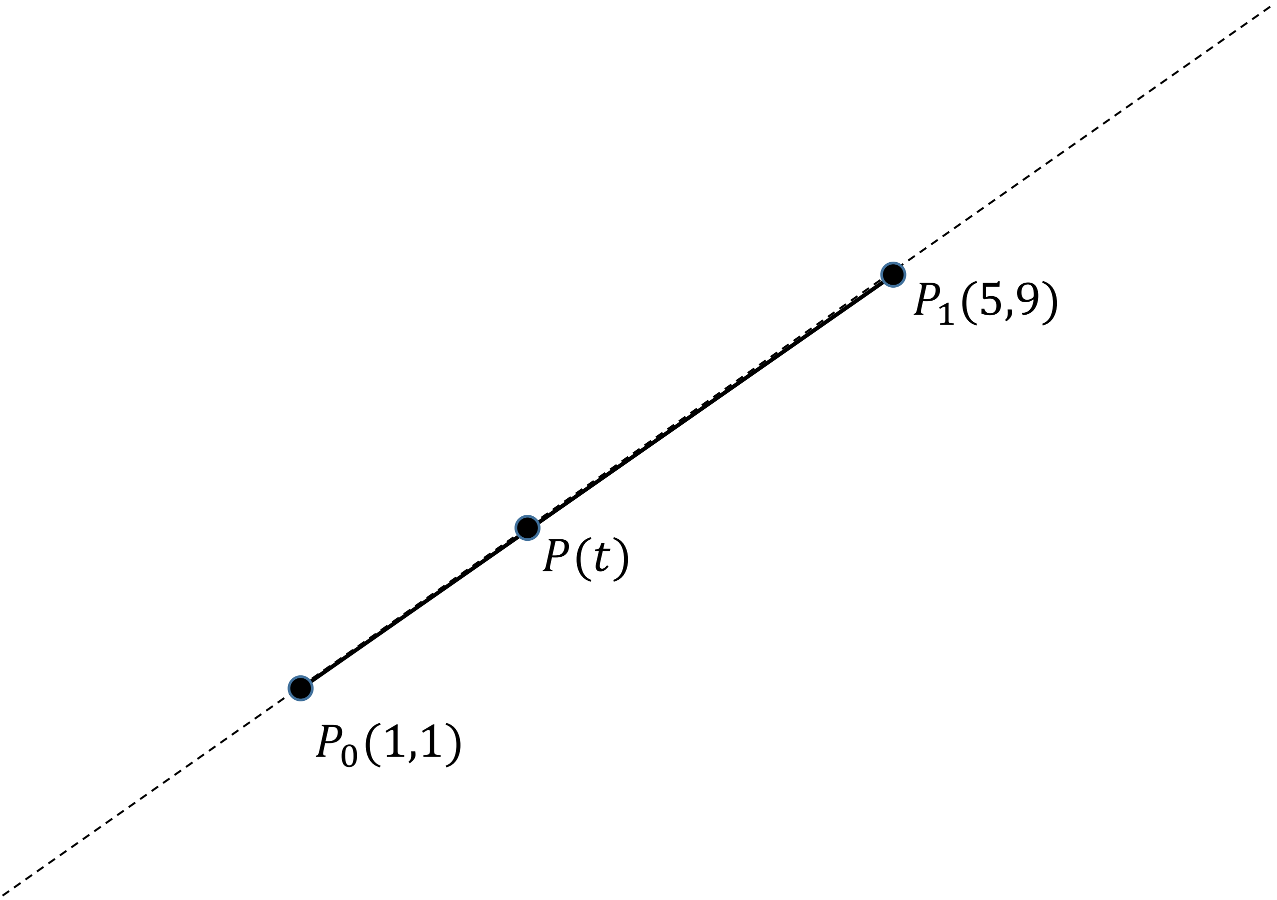
Parametric equation,

$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0)\\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0)\\
&= (x_0 + t(x_1 - x_0), y_0 + t(y_1 - y_0))
\end{aligned}
$$

$$P\left(\frac{1}{2}\right) = (x_0, y_0) + \frac{1}{2}.(x_1 - x_0, y_1 - y_0)$$

$$= \left(\frac{x_0 + x_1}{2}, \frac{y_0 + y_1}{2}\right)$$

Parametric equation of Line



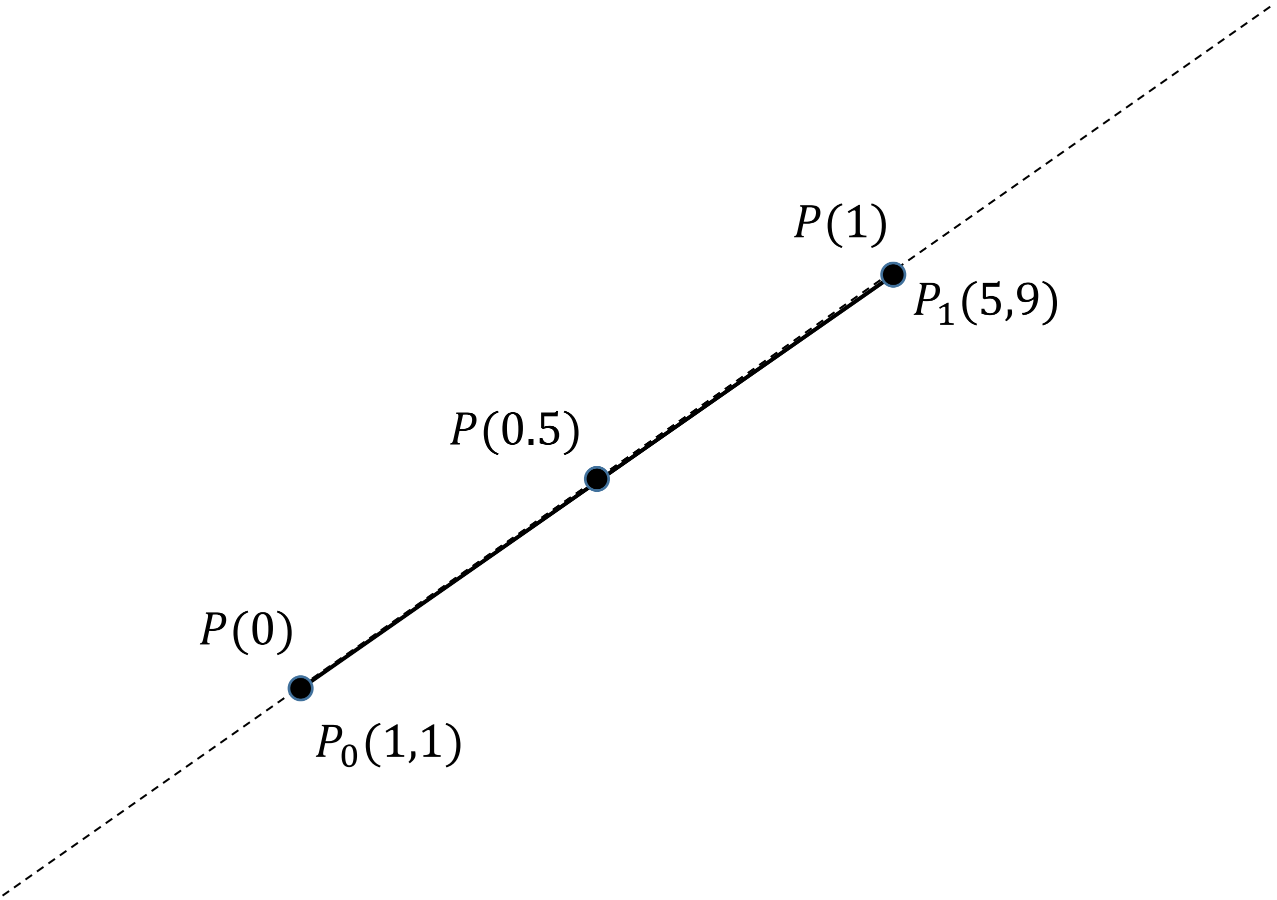
Suppose, endpoints of a line are (1, 1) and (5, 9)

Parametric equation,

$$\begin{aligned} P(t) &= P_0 + t.(P_1 - P_0) \\ &= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\ &= (1, 1) + t(5 - 1, 9 - 1) \\ &= (1, 1) + t(4, 8) \\ &= (1 + 4t, 1 + 8t) \end{aligned}$$

$$P(0) = (1, 1)$$
$$P(1) = (5, 9)$$

Parametric equation of Line



Suppose, endpoints of a line are (1, 1) and (5, 9)

Parametric equation,

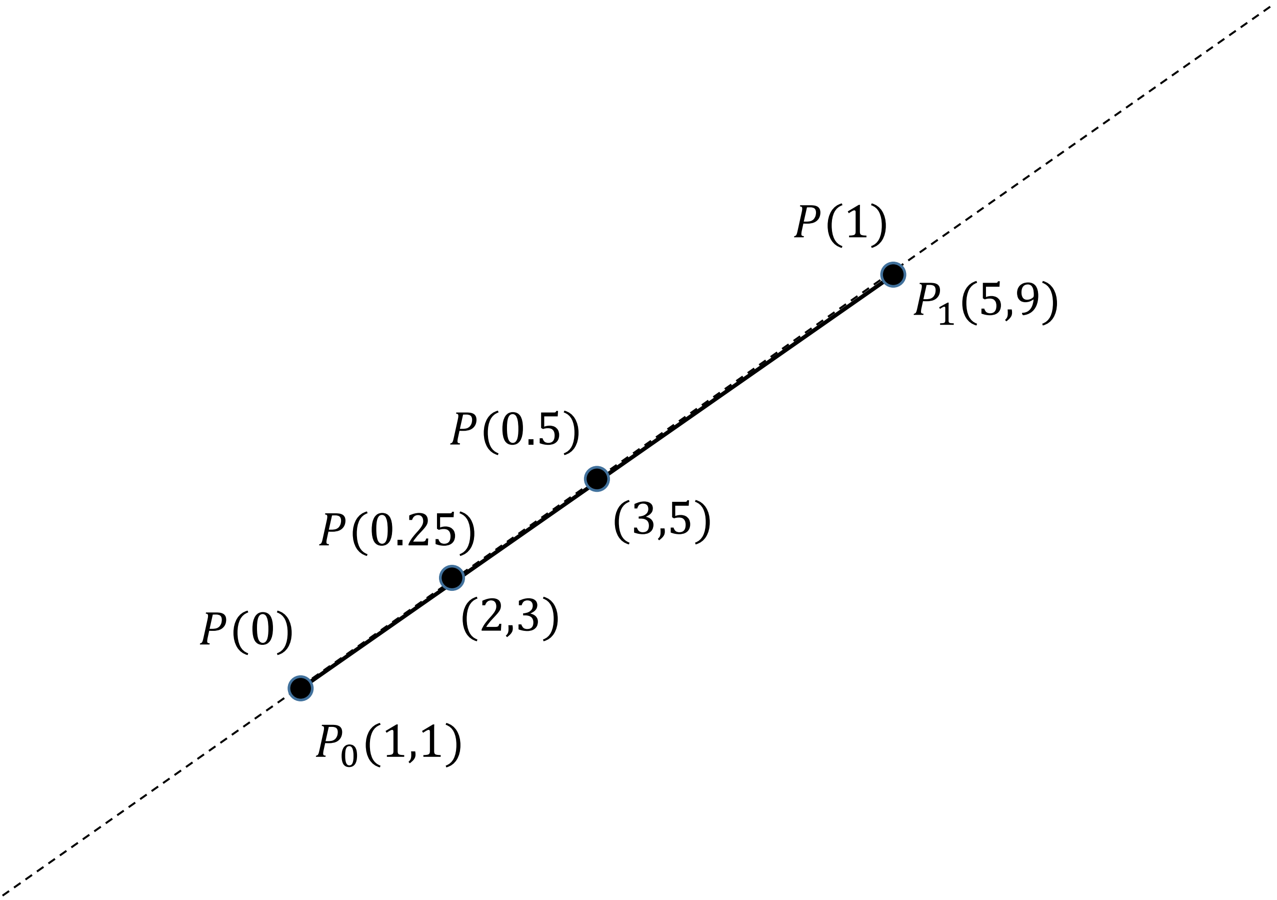
$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0) \\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\
&= (1,1) + t(5-1, 9-1) \\
&= (1,1) + t(4,8) \\
&= (1+4t, 1+8t)
\end{aligned}
$$

$P(0) = (1,1)$

$P(1) = (5,9)$

$P(0.5) = (1 + 4 \times 0.5, 1 + 8 \times 0.5) = (3,5)$

Parametric equation of Line



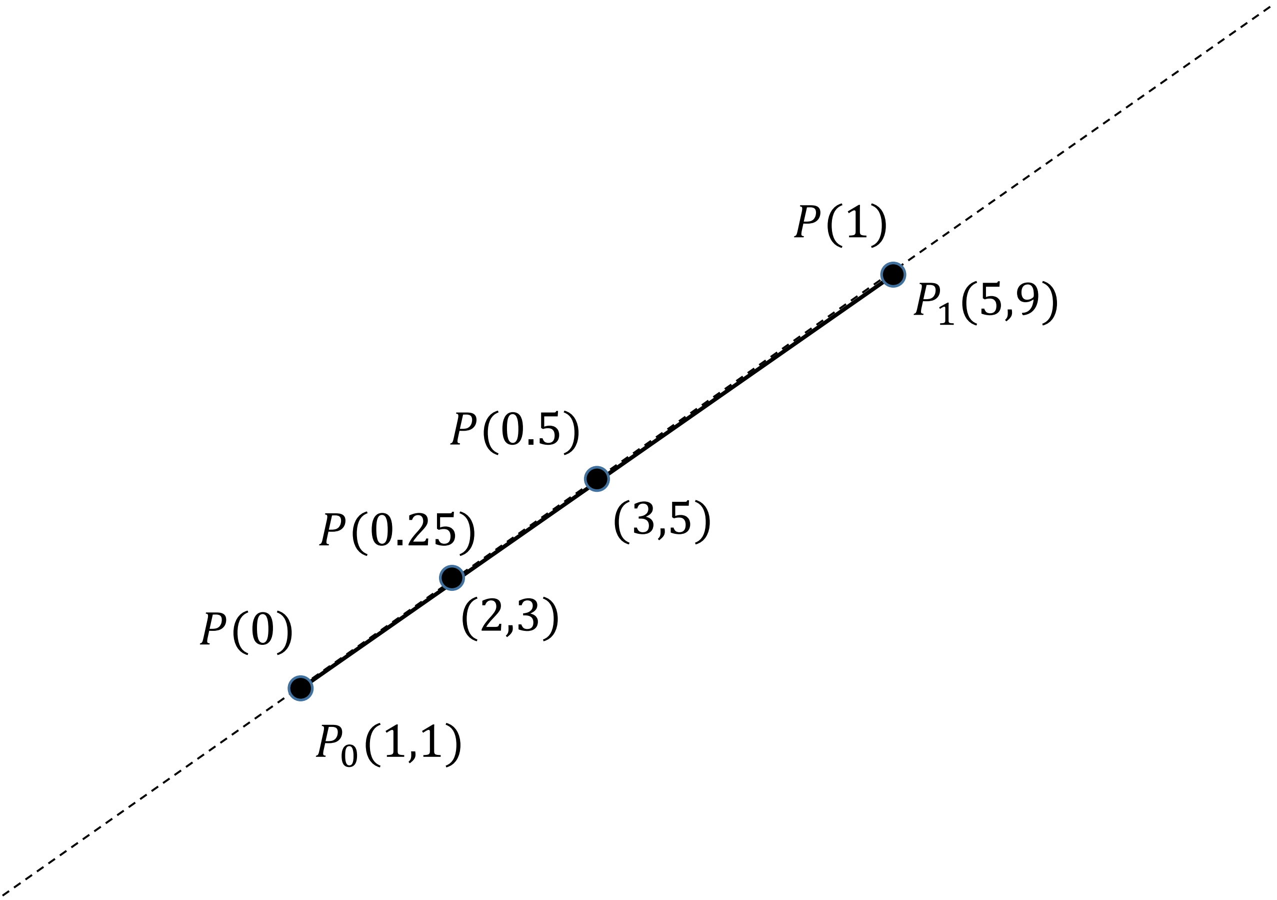
Suppose, endpoints of a line are (1, 1) and (5, 9)

Parametric equation,

$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0) \\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\
&= (1, 1) + t(5 - 1, 9 - 1) \\
&= (1, 1) + t(4, 8) \\
&= (1 + 4t, 1 + 8t)
\end{aligned}
$$

$P(0) = (1, 1)$
$P(1) = (5, 9)$
$P(0.5) = (1 + 4 \times 0.5, 1 + 8 \times 0.5) = (3, 5)$
$P(0.25) = (1 + 4 \times 0.25, 1 + 8 \times 0.25) = (2, 3)$

Parametric equation of Line



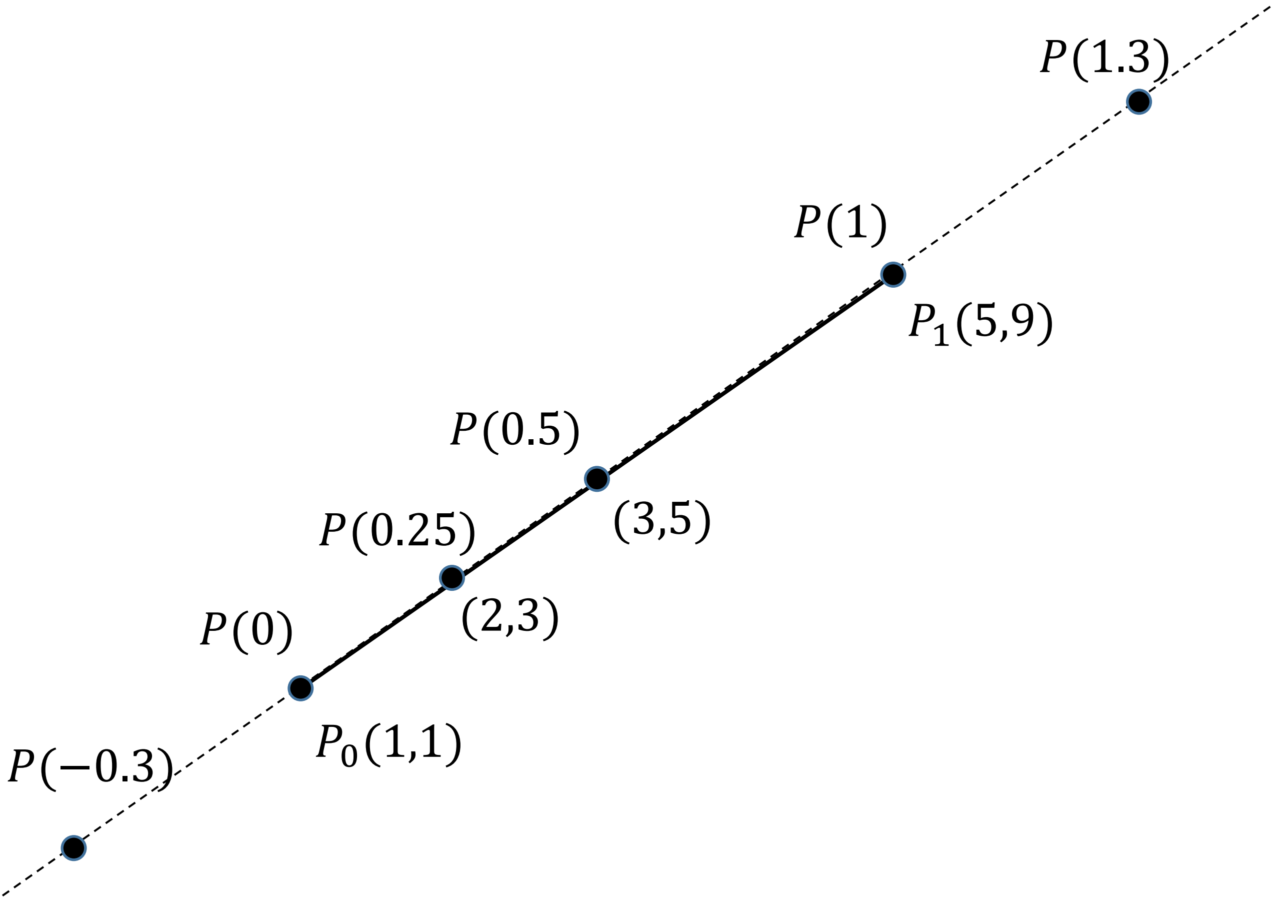
Suppose, endpoints of a line are (1, 1) and (5, 9)

Parametric equation,

$$\begin{aligned} P(t) &= P_0 + t.(P_1 - P_0) \\ &= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\ &= (1,1) + t(5-1, 9-1) \\ &= (1,1) + t(4,8) \\ &= (1+4t, 1+8t) \end{aligned}$$

So all points between $0 \leq t \leq 1$ are inside the line segment

Parametric equation of Line

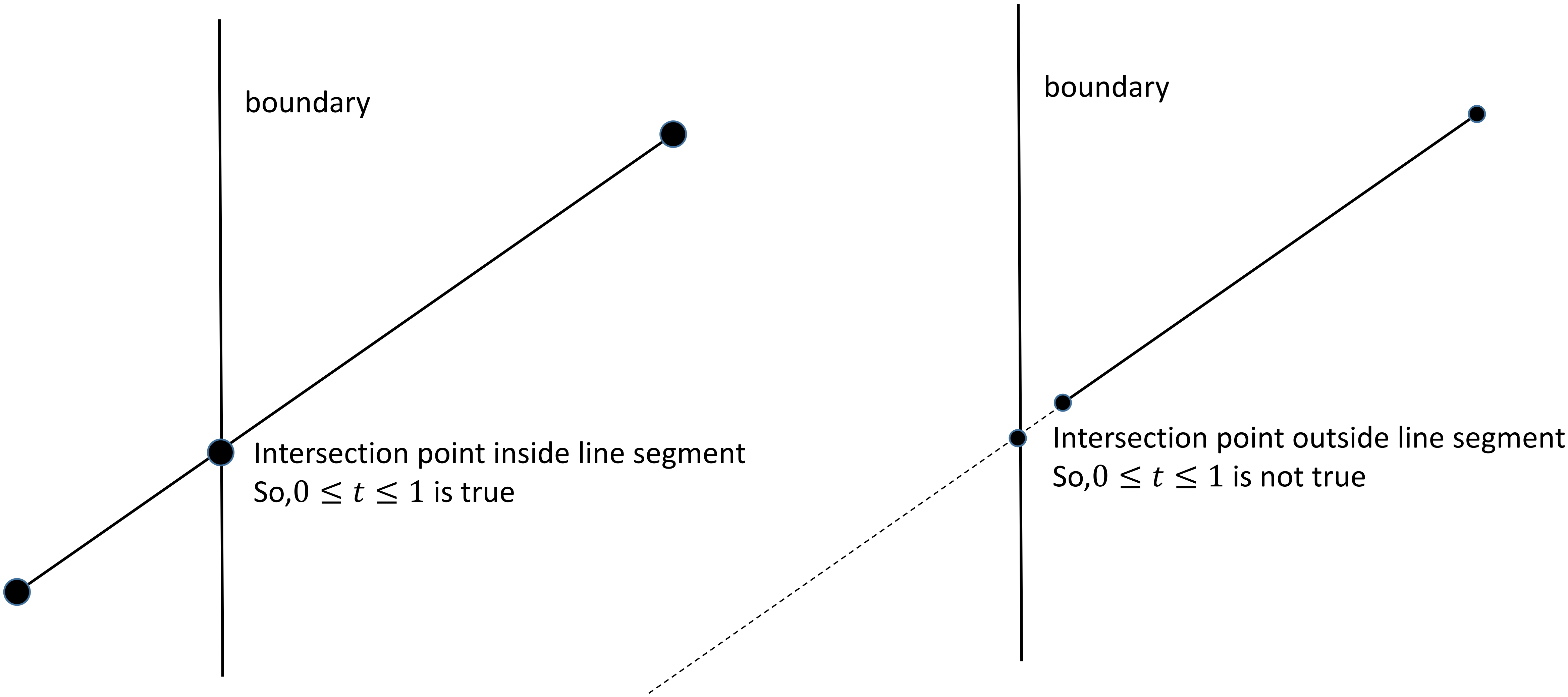


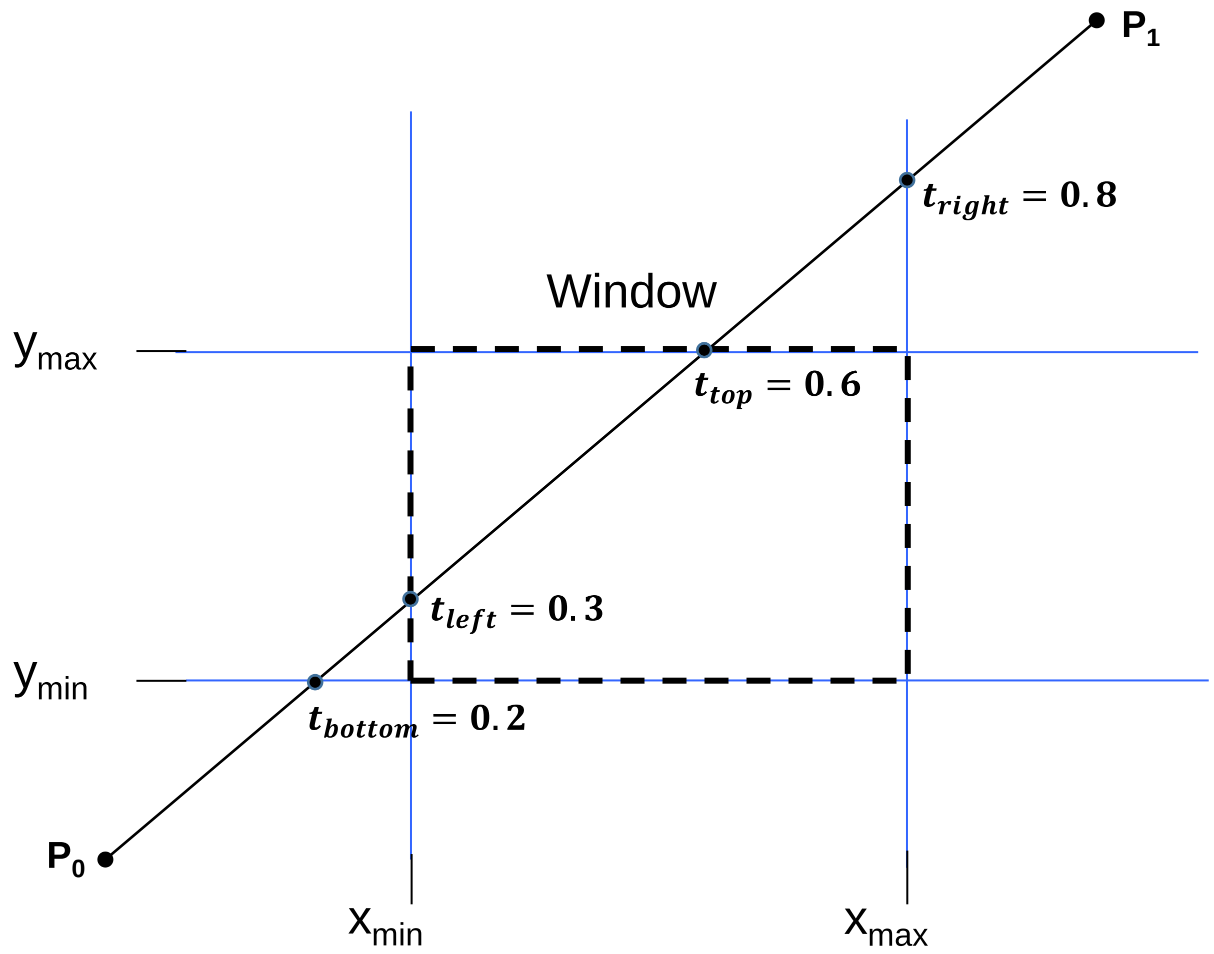
Suppose, endpoints of a line are (1, 1) and (5, 9)

Parametric equation,

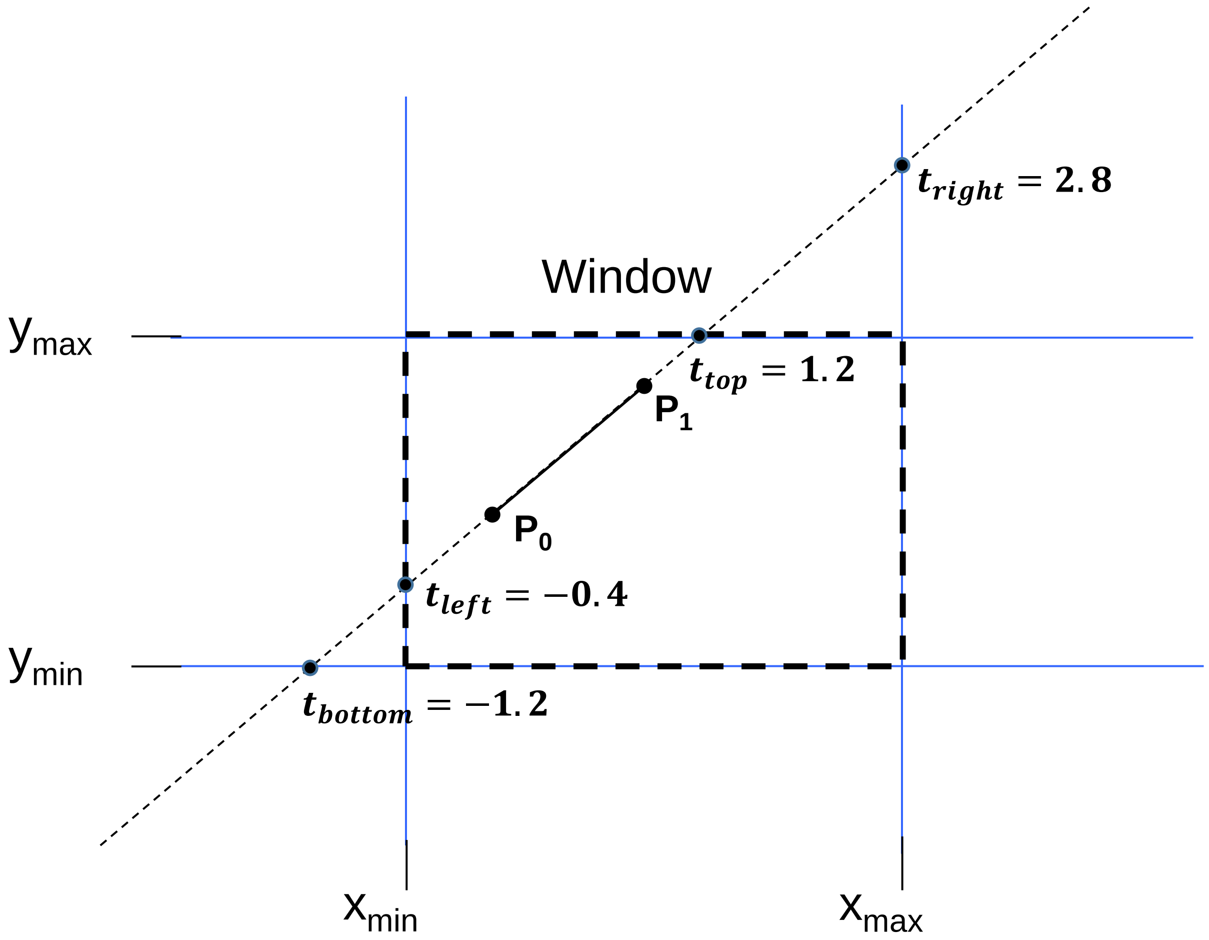
$$
\begin{aligned}
P(t) &= P_0 + t.(P_1 - P_0) \\
&= (x_0, y_0) + t(x_1 - x_0, y_1 - y_0) \\
&= (1, 1) + t(5 - 1, 9 - 1) \\
&= (1, 1) + t(4, 8) \\
&= (1 + 4t, 1 + 8t)
\end{aligned}
$$

P(-0.3) and P(1.3) are outside the line segment


boundary
Intersection point inside line segment
So,$0 \leq t \leq 1$ is true
boundary
Intersection point outside line segment
So,$0 \leq t \leq 1$ is not true

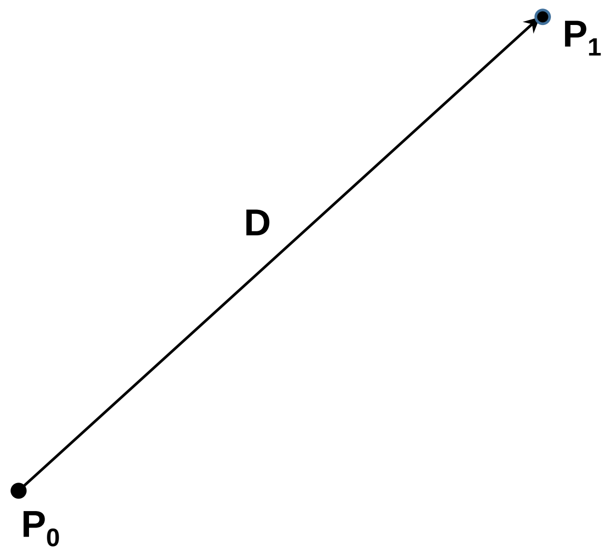


Both endpoints outside
All intersections have t between
0 and 1



Both endpoints inside
All intersections have t outside 0 to 1

Vector

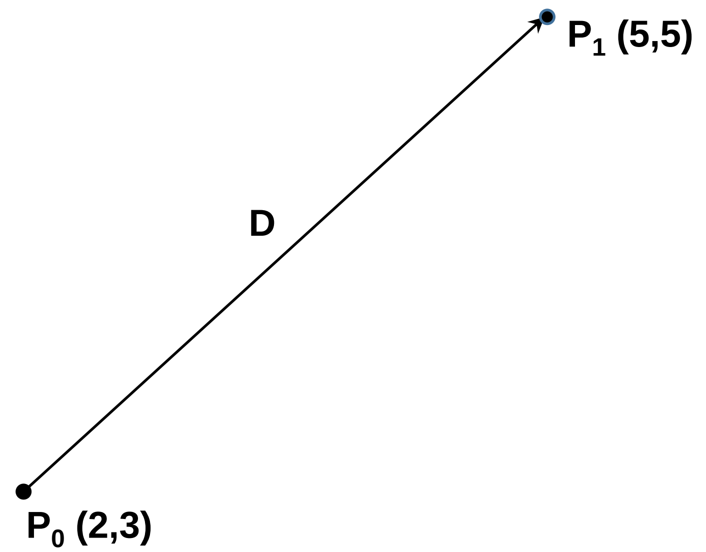


D is a vector from P_0 to P_1

$D = P_1 - P_0$

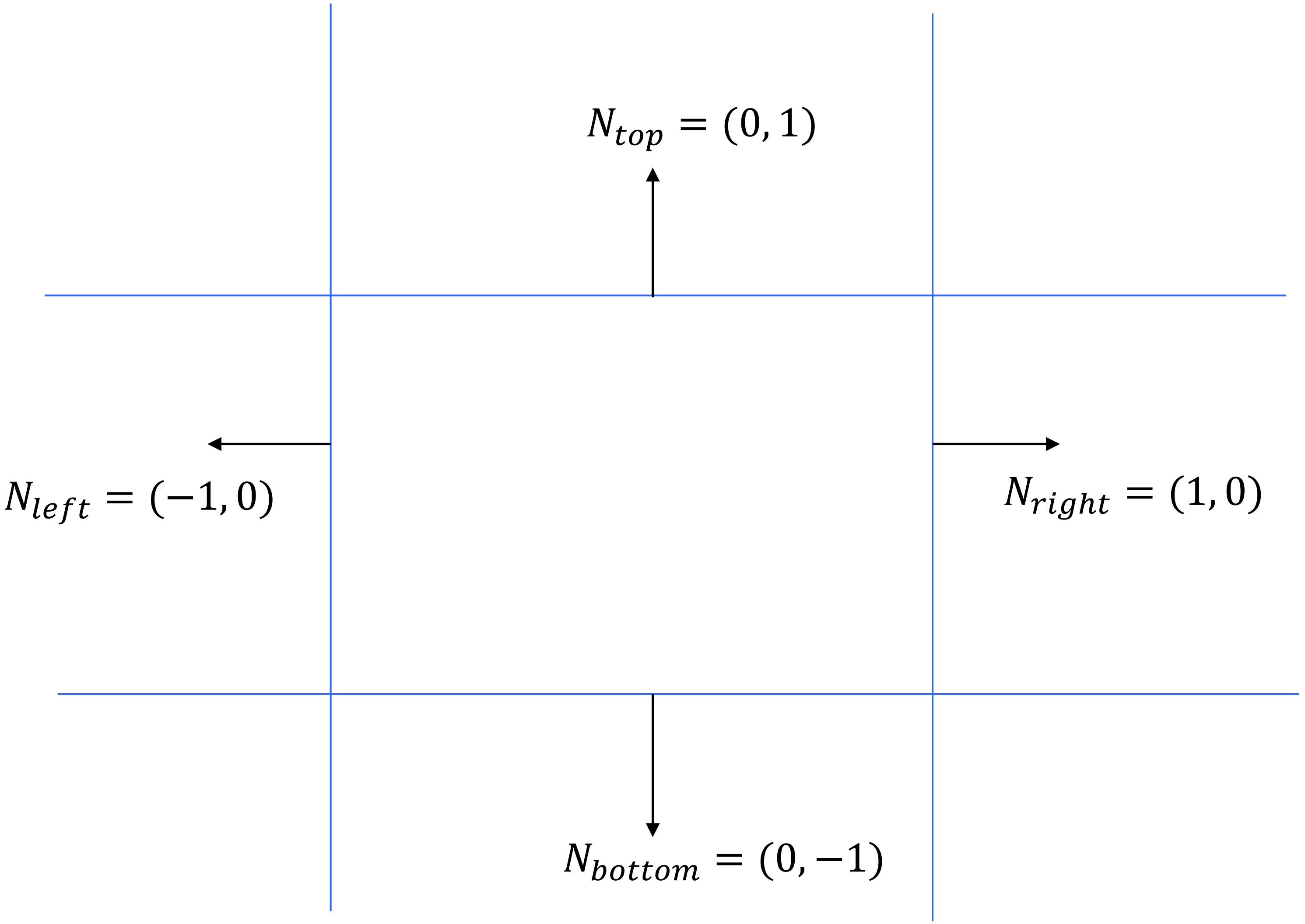
$= (x_1 - x_0, y_1 - y_0)$

Vector



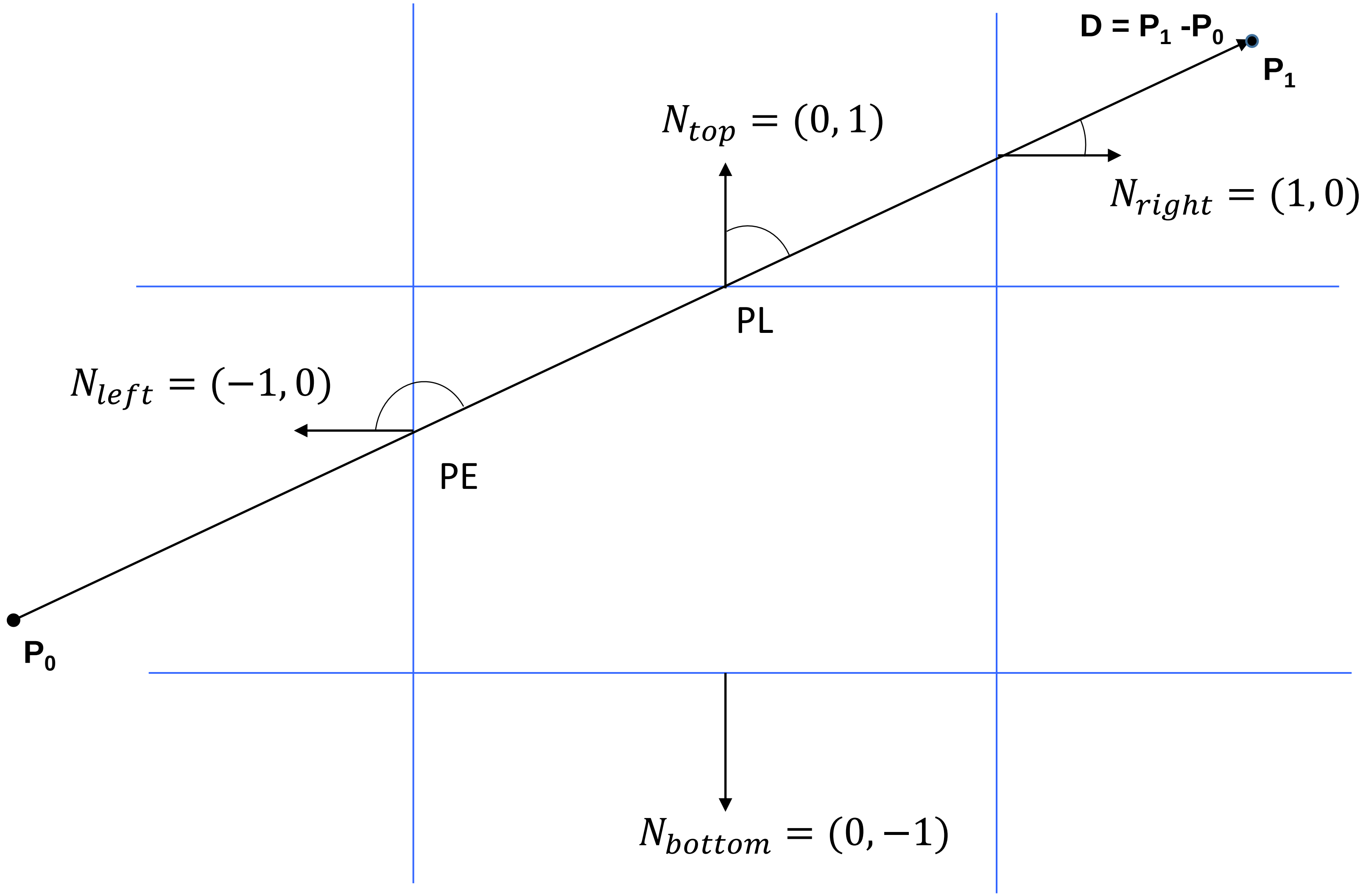
$$
\begin{aligned}
D &= P_1 - P_0 \\
&= (x_1 - x_0, y_1 - y_0) \\
&= (5 - 2, 5 - 3) \\
&= (3, 2) \\
&\cong 3i + 2j
\end{aligned}
$$

Normal vectors to boundary



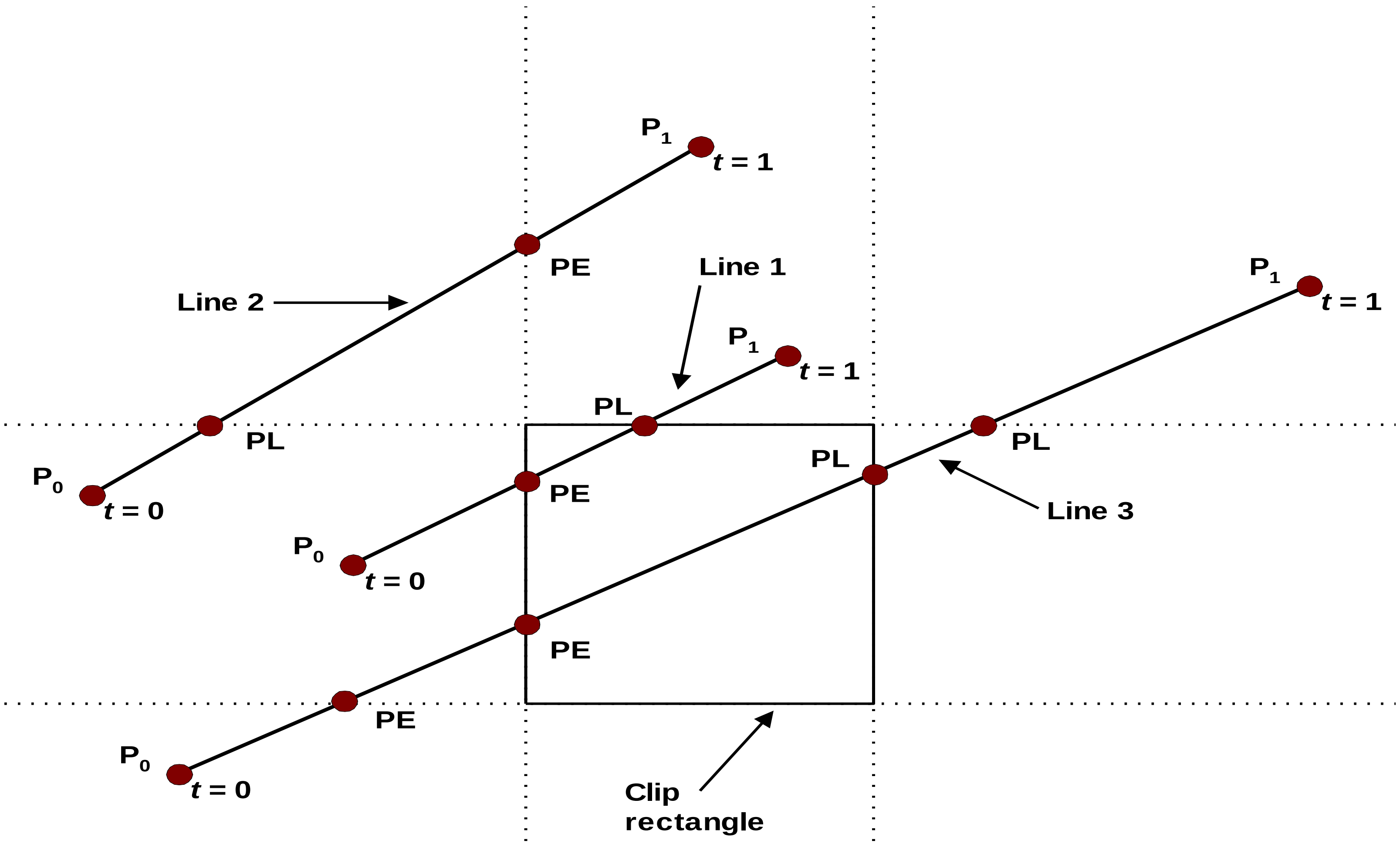
Each N_i is a perpendicular unit vector to each of the boundaries: left, right, top, bottom. N_i points away from the clip window

Normal vectors to boundary



When angle between N_i and D is more than 90 degree-
Or, $N_i \cdot D < 0$
The intersection point is PE
(Potentially entering)

When angle between N_i and D is less than 90 degree-
Or, $N_i \cdot D > 0$
The intersection point is PL
(Potentially leaving)



PE = Potentially Entering

$$N_i \cdot D < 0 \Rightarrow PE$$
$$\Rightarrow Angle > 90^\circ$$

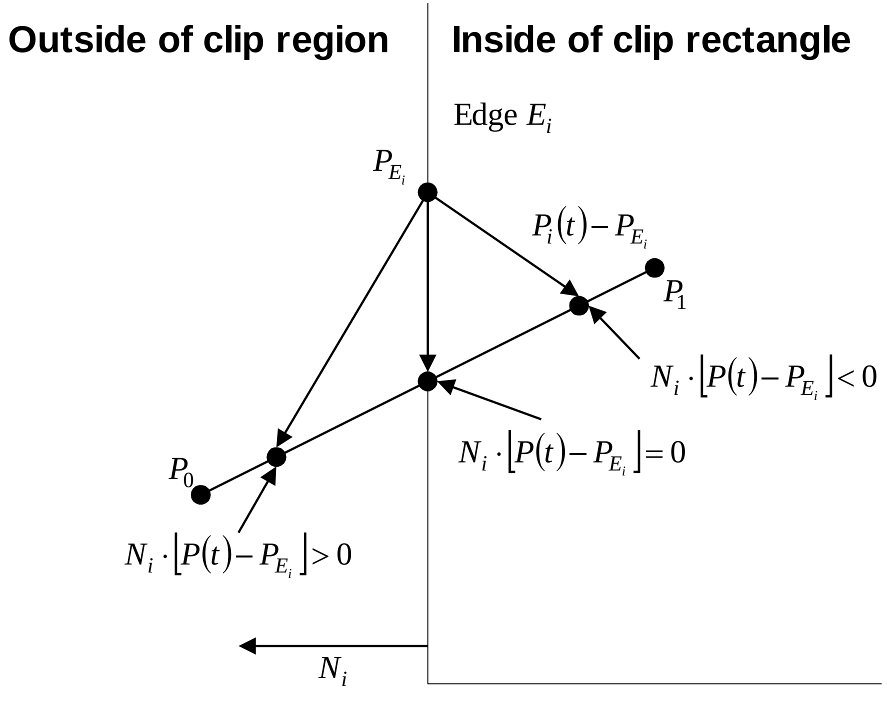
PL = Potentially Leaving

$$N_i \cdot D > 0 \Rightarrow PL$$
$$\Rightarrow Angle < 90^\circ$$

Cyrus-Beck algo

- Calculate t values of intersection points with each 4 boundaries
- Classify intersection points whether PE/PL
- Select the PE with highest t and the PL with the lowest t
- Using parametric line eqn. find the clipped points

The Cyrus-Beck Algorithm



$$Line\ P_0P_1 : P(t) = P_0 + (P_1 - P_0)t$$

$$N_i \cdot \left[P(t) - P_{E_i}\right] = 0$$

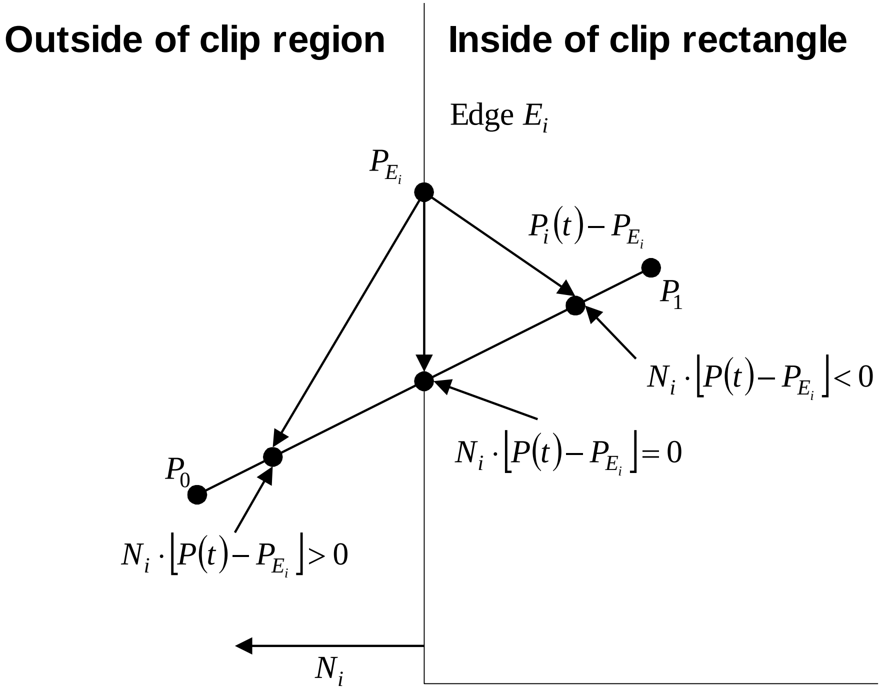
$$\Rightarrow N_i \cdot \left[P_0 + (P_1 - P_0)t - P_{E_i}\right] = 0$$

$$\Rightarrow N_i \cdot \left[P_0 + (P_1 - P_0)t - P_{E_i}\right] = 0$$

$$\Rightarrow t = \frac{N_i \cdot \left[P_0 - P_{E_i}\right]}{-N_i \cdot (P_1 - P_0)}$$

$$\Rightarrow t = \frac{N_i \cdot \left[P_0 - P_{E_i}\right]}{-N_i \cdot D},\ D = P_1 - P_0$$

The Cyrus-Beck Algorithm



For left boundary,

$$t_{left} = \frac{N_i.[P_0 - P_E]}{-N_i.[P_1 - P_0]}$$

$$= \frac{(-1,0).[(x_0, y_0) - (x_{min}, y)]}{-(-1,0).[(x_1, y_1) - (x_0, y_0)]}$$

$$= \frac{(-1,0).(x_0 - x_{min}, y_0 - y)}{-(-1,0).(x_1 - x_0, y_1 - y_0)}$$

$$= \frac{-1 \times (x_0 - x_{min}) + 0 \times (y_0 - y)}{-(-1 \times (x_1 - x_0) + 0 \times (y_1 - y_0))}$$

$$= \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$

Similarly,

$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

The Cyrus-Beck Algorithm

```
Precalculate N_i and P_Ei for each edge
for (each line segment to be clipped) {
   if (P_1 == P_0)
            line is degenerated, so clip as a point;
   else {
            t_E = 0;  t_L = 1;
            for (each candidate intersection with a clip edge) {
                        if (N_i • D != 0) {   /* Ignore edges parallel to line */
                                    calculate t;
                                    use sign of  N_i • 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 intersection;
   }
}
```

a) Calculate the value of t of the lines given below for all edges and specify whether they are entering or leaving t. [Given (0,0) to (300,200) be the clip region.]

- (i) (50, -125) to (-100, 225).
- (ii) (-250, 200) to (250, -200).
- (iii) (-250, 200) to (150, 100)

Also, find the line segment within the clipping window

a)(i) boundary:

$$x_{min} = 0,\ \ x_{max} = 300,\ \ y_{min} = 0,\ \ y_{max} = 200$$

points:

$$x_0 = 50, y_0 = -125, x_1 = -100, y_1 = 225$$

$$D = (x_1 - x_0, y_1 - y_0) = (-150, 350)$$

Initially, $t_E = 0, t_L = 1$

$$t_{left} = \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$

$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

Boundary	N_i	$N_i .D$	t	PE/PL	t_E	t_L
Left	(-1,0)	150	$\frac{-(50-0)}{-100-50} = 0.33$	PL	0	0.33
Right						
Bottom						
Top						

Algorithm

Precalculate N_i and P_{Ei} for each edge
for *(each line segment to be clipped)* {
  if (P_1 == P_0)
    line is degenerated, so clip as a point;
  else {
    t_E = 0; t_L = 1;
    for (*each candidate intersection with a clip edge*) {
      if (N_i • *D* != 0) { /* Ignore edges parallel to line */
        calculate t;
        use sign of N_i • 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 intersection;*
  }
}

$$t_{left} = \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$

$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

a)(i) boundary:

$$x_{min} = 0, \quad x_{max} = 300, \quad y_{min} = 0, \quad y_{max} = 200$$

points:

$$x_0 = 50, y_0 = -125, x_1 = -100, y_1 = 225$$

$$D = (x_1 - x_0, y_1 - y_0) = (-150, 350)$$

Initially, $t_E = 0, t_L = 1$

Boundary	N_i	$N_i.D$	t	PE/PL	t_E	t_L
Left	(-1,0)	150	$\frac{-(50-0)}{-100-50} = 0.33$	PL	0	0.33
Right	(1,0)	-150	$\frac{-(50-300)}{-100-50} = -1.67$	PE	0	0.33

Algorithm

Precalculate N_i *and* P_{Ei} *for each edge*

```
for (each line segment to be clipped) {
    if (P1 == P0)
            line is degenerated, so clip as a point;
    else {
            tE = 0;  tL = 1;
            for (each candidate intersection with a clip edge) {
                    if (Ni • D != 0) {   /* Ignore edges parallel to line */
                            calculate t;
                            use sign of  Ni • D to categorize as PE or PL;
                            if (PE) tE = max(tE ,  t);
                            if (PL) tL = min(tL ,  t);
                    }
            }
            if (tE > tL) return NULL;
            else return P(tE) and P(tL) as true clip intersection;
    }
}
```

a)(i) boundary:

$$x_{min} = 0, \ \ x_{max} = 300, \ \ y_{min} = 0, \ \ y_{max} = 200$$

points:

$$x_0 = 50, y_0 = -125, x_1 = -100, y_1 = 225$$

$$D = (x_1 - x_0, y_1 - y_0) = (-150, 350)$$

Initially, $t_E = 0, t_L = 1$

$$t_{left} = \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$

$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

Boundary	N_i	$N_i .D$	t	PE/PL	t_E	t_L
Left	(-1,0)	150	$\frac{-(50-0)}{-100-50} = 0.33$	PL	0	0.33
Right	(1,0)	-150	$\frac{-(50-300)}{-100-50} = -1.67$	PE	0	0.33
Bottom	(0, -1)	-350	$\frac{-(-125-0)}{225-(-125)} = 0.357$	PE	0.357	0.33

Algorithm

Precalculate N_i and P_{Ei} for each edge

for *(each line segment to be clipped)* {

  if (P_1 == P_0)

    line is degenerated, so clip as a point;

  else {

    t_E = 0; t_L = 1;

    for (*each candidate intersection with a clip edge*) {

      if (N_i • *D* != 0) { /* Ignore edges parallel to line */

        calculate t;

        use sign of N_i • 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 intersection;*

  }

}

a)(i) boundary:

$$x_{min} = 0, \;\; x_{max} = 300, \;\; y_{min} = 0, \;\; y_{max} = 200$$

points:

$$x_0 = 50, y_0 = -125, x_1 = -100, y_1 = 225$$

$$D = (x_1 - x_0, y_1 - y_0) = (-150, 350)$$

Initially, $t_E = 0, t_L = 1$

$$t_{left} = \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$

$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

Boundary	N_i	$N_i .D$	t	PE/PL	t_E	t_L
Left	(-1,0)	150	$\frac{-(50-0)}{-100-50} = 0.33$	PL	0	0.33
Right	(1,0)	-150	$\frac{-(50-300)}{-100-50} = -1.67$	PE	0	0.33
Bottom	(0, -1)	-350	$\frac{-(-125-0)}{225-(-125)} = 0.357$	PE	0.357	0.33
Top	(0, 1)	350	$\frac{-(-125-200)}{225-(-125)} = 0.93$	PL	0.357	0.33

Since $t_E > t_L$, *line segment is outside clip window*

Algorithm

Precalculate N_i *and* P_{Ei} *for each edge*

for *(each line segment to be clipped)* {
  if ($P_1 == P_0$)
    line is degenerated, 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 */
        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 intersection;*
  }
}

a)(ii) boundary:

$$x_{min} = 0, \;\; x_{max} = 300, \;\; y_{min} = 0, \;\; y_{max} = 200$$

points:

$$x_0 = -250, y_0 = 200, x_1 = 250, y_1 = -200$$
$$D = (x_1 - x_0, y_1 - y_0) = (500, -400)$$

Initially, $t_E = 0, t_L = 1$

Boundary	N_i	$N_i .D$	t	PE/PL	t_E	t_L
Left	(-1,0)	-500	$\frac{-(-250-0)}{500} = 0.5$	PE	0.5	1
Right	(1,0)	500	$\frac{-(-250-300)}{500} = 1.1$	PL	0.5	1
Bottom	(0, -1)	400	$\frac{-(200-0)}{-400} = 0.5$	PL	0.5	0.5
Top	(0, 1)	-400	$\frac{-(200-200)}{-400} = 0$	PE	0.5	0.5

P(0.5) and P(0.5) (same point) are the true clip intersection

$$P(0.5) = (x_0, y_0) + 0.5 \times D$$
$$= (-250, 200) + 0.5 \times (500, -400)$$
$$= (0,0)$$

$$t_{left} = \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$
$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$
$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$
$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

Algorithm

```
Precalculate Ni and PEi for each edge
for (each line segment to be clipped) {
    if (P1 == P0)
            line is degenerated, so clip as a point;
    else {
            tE = 0;  tL = 1;
            for (each candidate intersection with a clip edge) {
                    if (Ni • D != 0) {   /* Ignore edges parallel to line */
                            calculate t;
                            use sign of  Ni • D to categorize as PE or PL;
                            if (PE) tE = max(tE ,  t);
                            if (PL) tL = min(tL ,  t);
                    }
            }
            if (tE > tL) return NULL;
            else return P(tE) and P(tL) as true clip intersection;
    }
}
```

a)(iii) boundary:

$$x_{min} = 0, \quad x_{max} = 300, \quad y_{min} = 0, \; y_{max} = 200$$

points:

$$x_0 = -250, y_0 = 200, x_1 = 150, y_1 = 100$$

$$D = (x_1 - x_0, y_1 - y_0) = (400, -100)$$

Initially, $t_E = 0, t_L = 1$

Boundary	N_i	$N_i.D$	t	PE/PL	t_E	t_L
Left	(-1,0)	-400	$\frac{-(-250-0)}{400} = 0.625$	PE	0.625	1
Right	(1,0)	400	$\frac{-(-250-300)}{400} = 1.375$	PL	0.625	1
Bottom	(0, -1)	100	$\frac{-(200-0)}{-100} = 2$	PL	0.625	1
Top	(0, 1)	-100	$\frac{-(200-200)}{-100} = 0$	PE	0.625	1

P(0.625) and P(1) are the true clip intersection

$$P(0.625) = (x_0, y_0) + 0.625 \times D$$
$$= (-250, 200) + 0.625 \times (400, -100)$$
$$= (0, 137.5)$$
$$P(1) = (150, 100)$$

(0, 137.5) and (150, 100) are the endpoints of the clipped line

$$t_{left} = \frac{-(x_0 - x_{min})}{(x_1 - x_0)}$$

$$t_{right} = \frac{-(x_0 - x_{max})}{(x_1 - x_0)}$$

$$t_{top} = \frac{-(y_0 - y_{max})}{(y_1 - y_0)}$$

$$t_{bottom} = \frac{-(y_0 - y_{min})}{(y_1 - y_0)}$$

Algorithm

```
Precalculate Ni and PEi for each edge
for (each line segment to be clipped) {
    if (P1 == P0)
            line is degenerated, so clip as a point;
    else {
            tE = 0;  tL = 1;
            for (each candidate intersection with a clip edge) {
                    if (Ni • D != 0) {   /* Ignore edges parallel to line */
                            calculate t;
                            use sign of  Ni • D to categorize as PE or PL;
                            if (PE) tE = max(tE ,  t);
                            if (PL) tL = min(tL ,  t);
                    }
            }
            if (tE > tL) return NULL;
            else return P(tE) and P(tL) as true clip intersection;
    }
}
```