Plan

Ou comment passer du format vectoriel au format bitmap

Ou comment passer du format vectoriel au format bitmap

```
segment( 2,2 , 16,28 ).tracer(bleu)
cercle( 35,14 , 12 ).remplir(rouge)
```

Description *vectorielle*

Ou comment passer du format vectoriel au format bitmap

```
segment( 2,2 , 16,28 ).tracer(bleu)
cercle( 35,14 , 12 ).remplir(rouge)
```

Description *vectorielle*

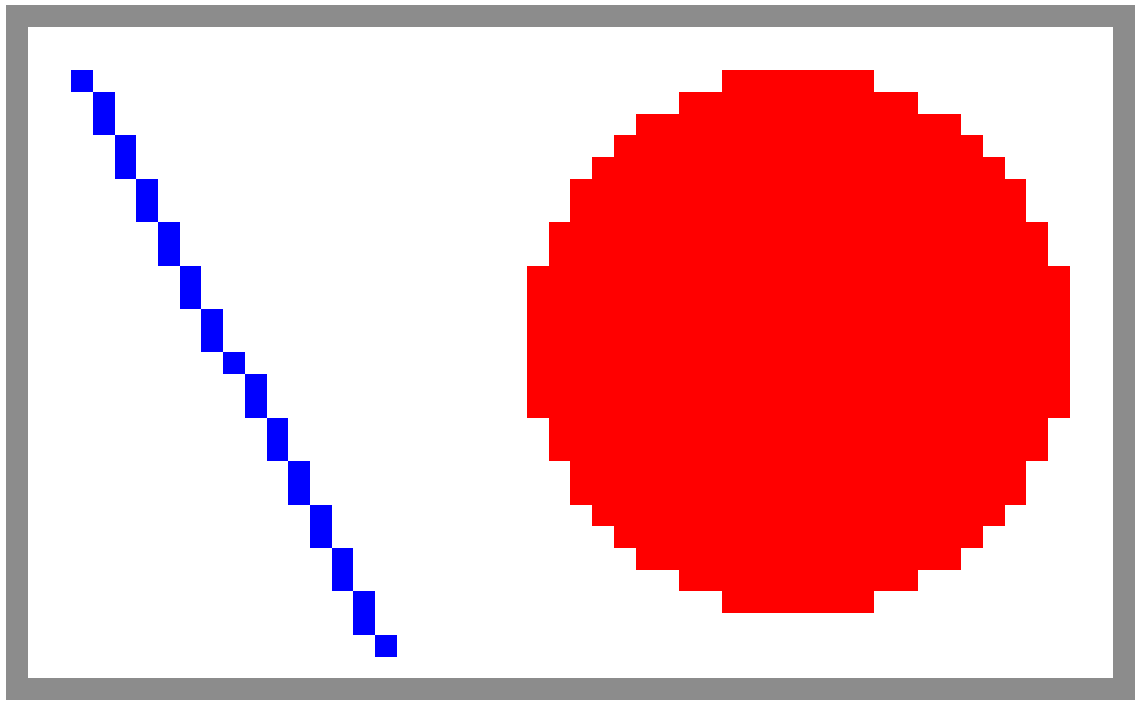


Image *bitmap* correspondante

Point ou pixel

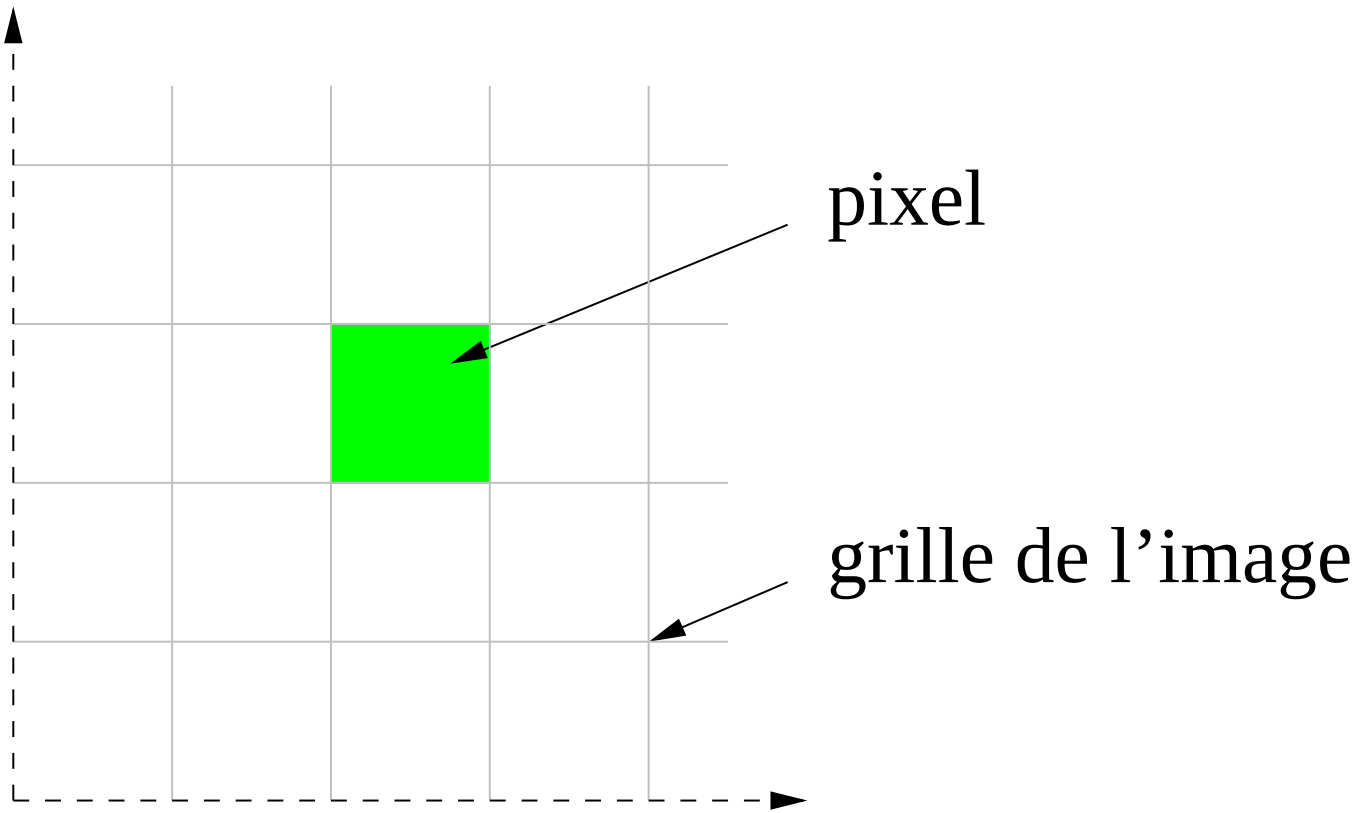
Grille en coordonnées entières

Point ou pixel

Grille en coordonnées entières

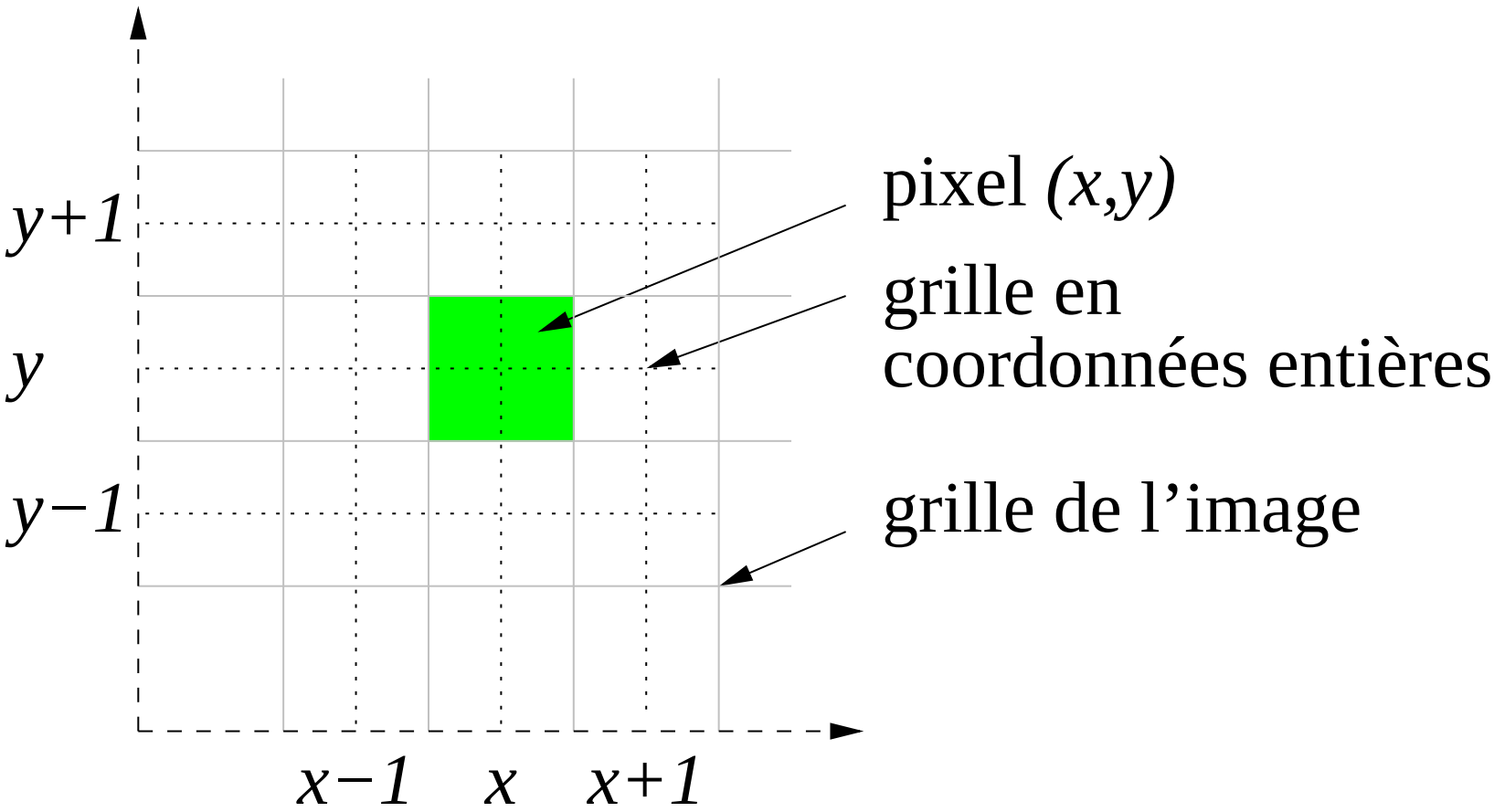
Point ou pixel

Grille en coordonnées entières



Point ou pixel

Grille en coordonnées entières

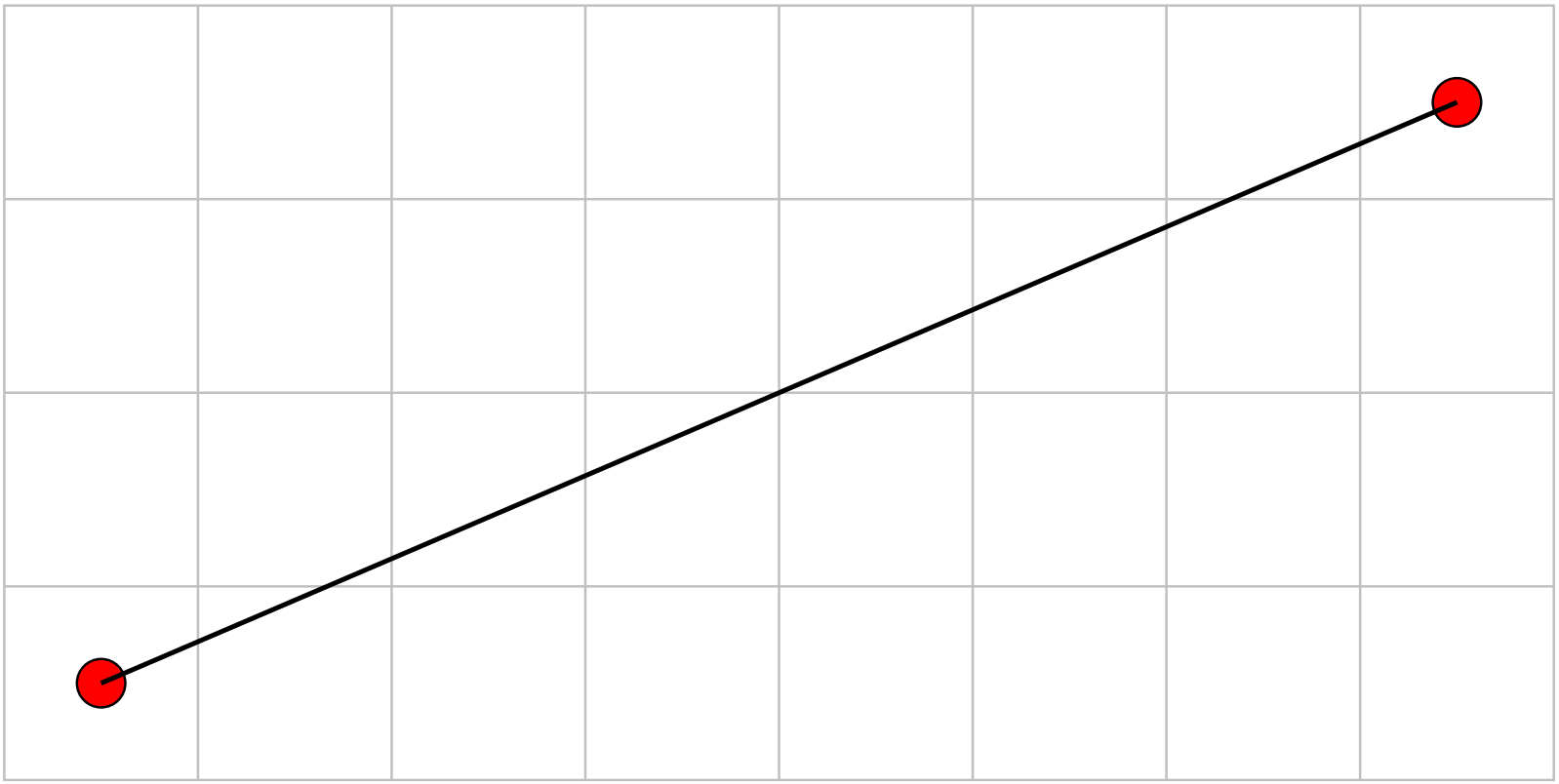


Segment de droite

cas d'une pente inférieure à 1

Segment de droite

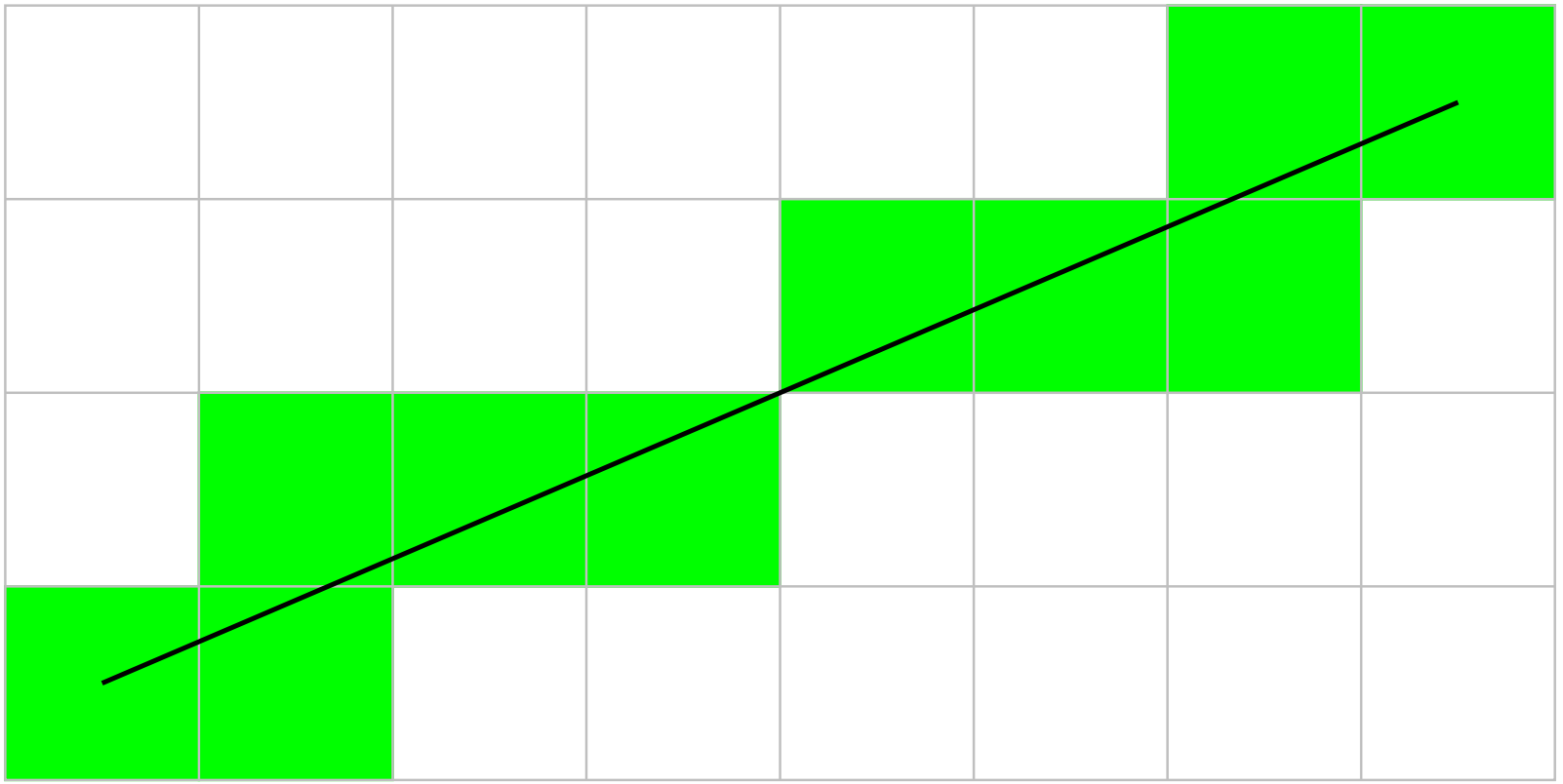
cas d'une pente inférieure à 1



Tracé idéal

Segment de droite

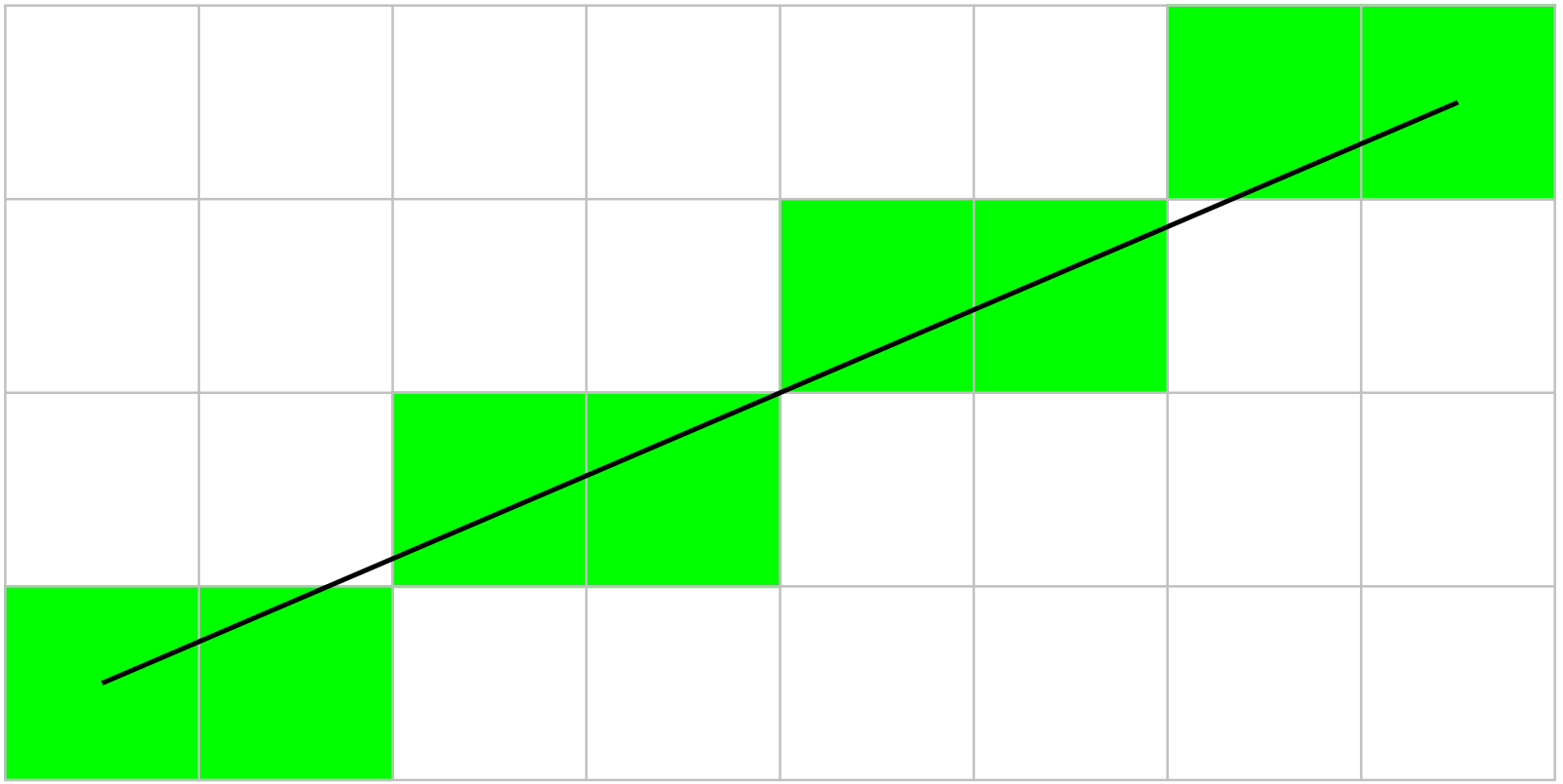
cas d'une pente inférieure à 1



Tracé des pixels intersectés par le segment

Segment de droite

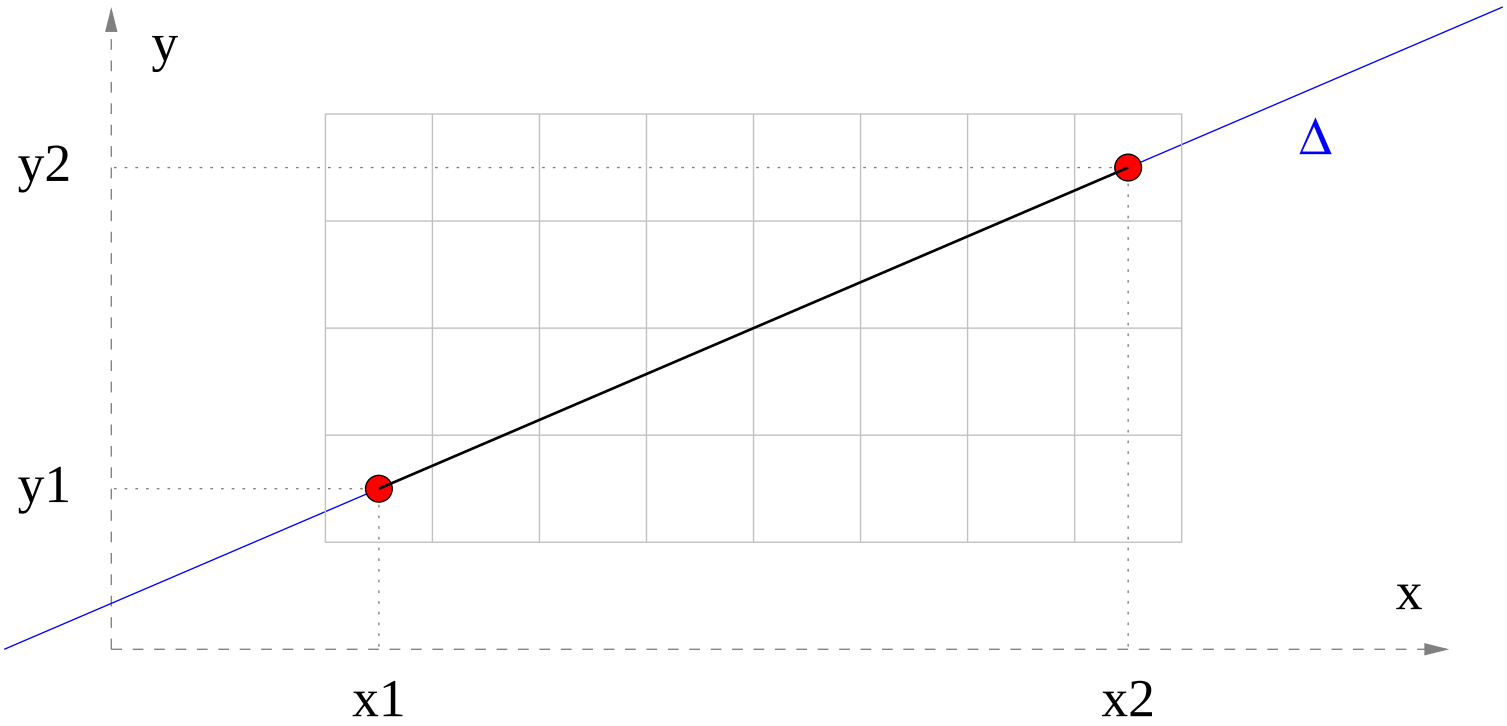
cas d'une pente inférieure à 1



Tracé d'un pixel par abscisse

Segment de droite

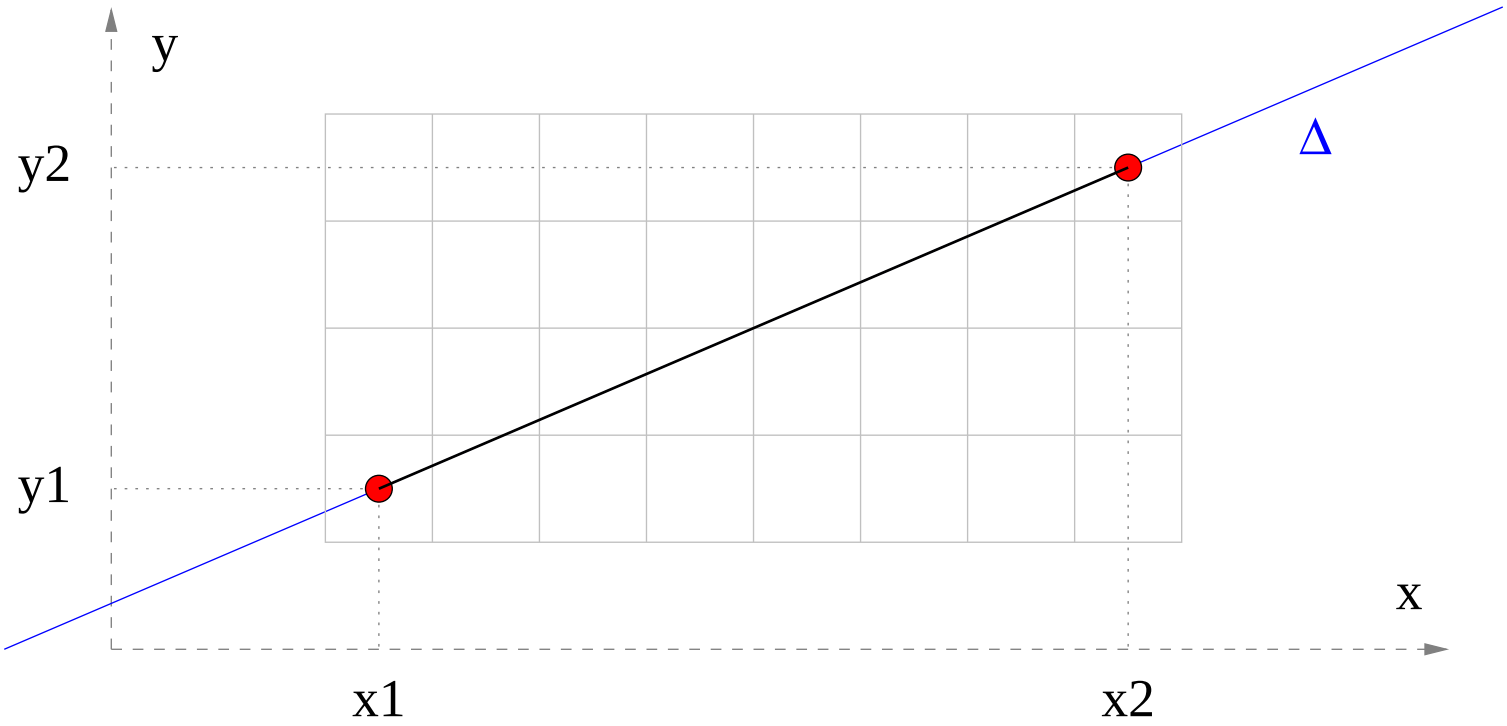
Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)



Equation de la droite Δ contenant le segment $S = [(x1, y1), (x2, y2)]$

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

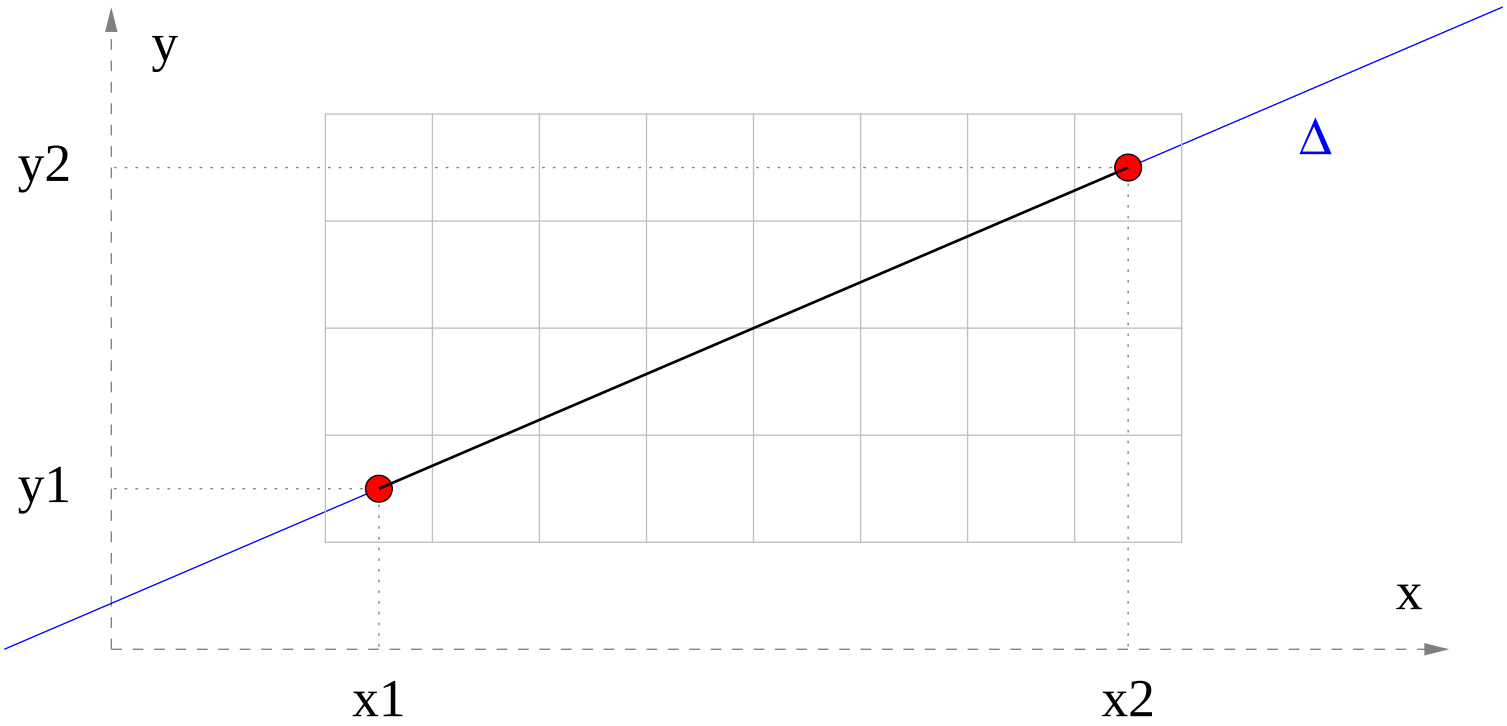


Equation de la droite Δ contenant le segment $S = [(x1, y1), (x2, y2)]$

$$y = y1 + \frac{dy}{dx}(x - x1)$$

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

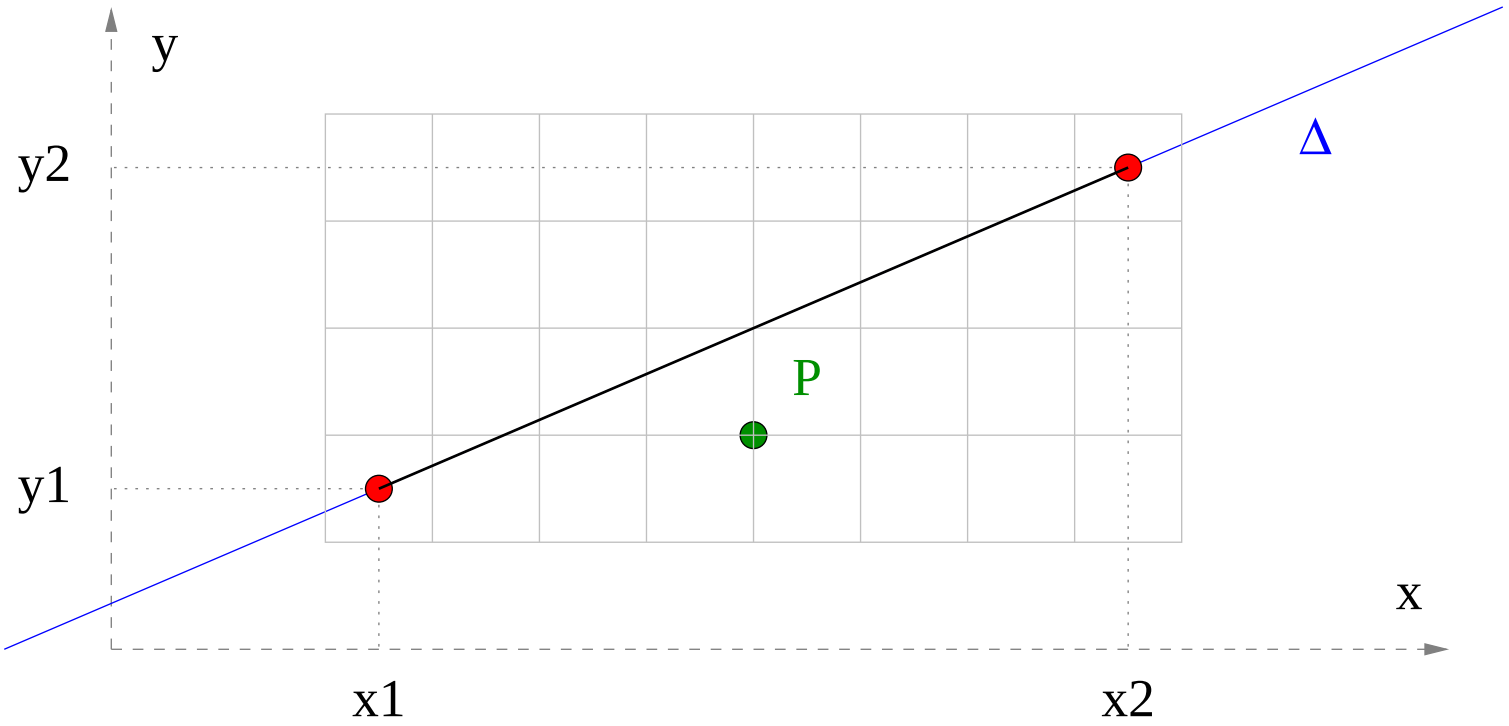


Equation de la droite Δ contenant le segment $S = [(x1, y1), (x2, y2)]$

$$\Longleftrightarrow \; F(x, y) = 2\ dx\ (y - y1) - 2\ dy\ (x - x1) = 0$$

Segment de droite

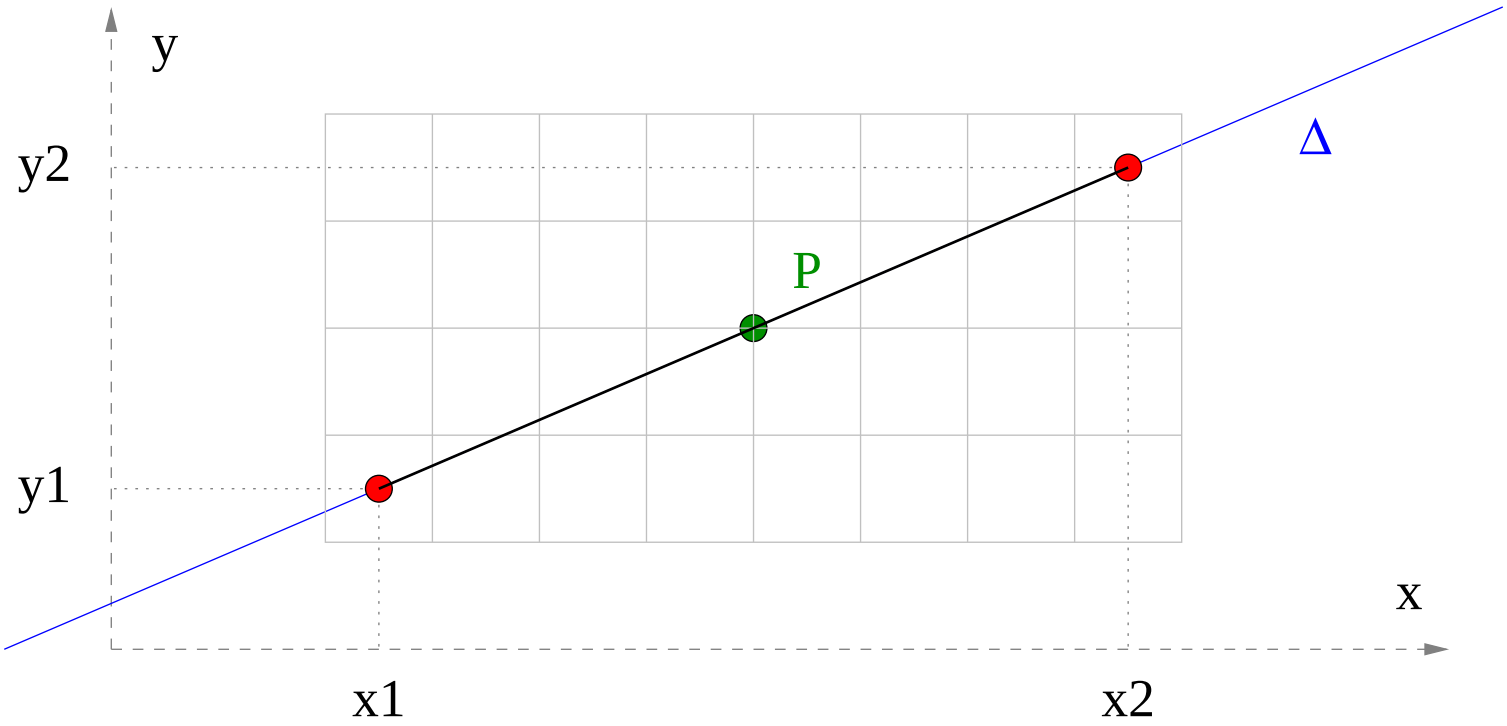
Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)



$P = (x, y)$ en dessous de la droite $\Delta \iff F(P) = F(x, y) < 0$

Segment de droite

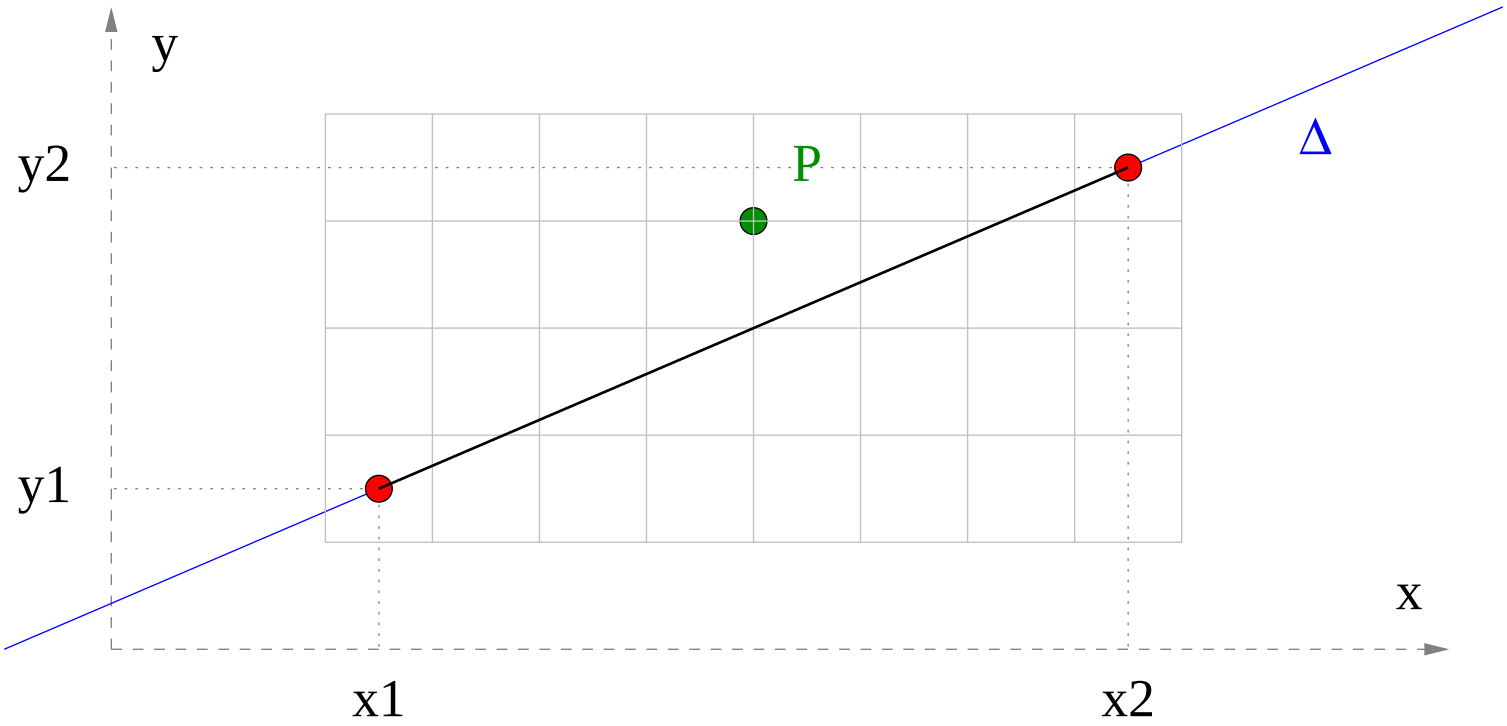
Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)



$$P = (x, y) \text{ sur la droite } \Delta \iff F(P) = F(x, y) = 0$$

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)



$$P = (x, y) \text{ en dessus de la droite } \Delta \iff F(P) = F(x, y) > 0$$

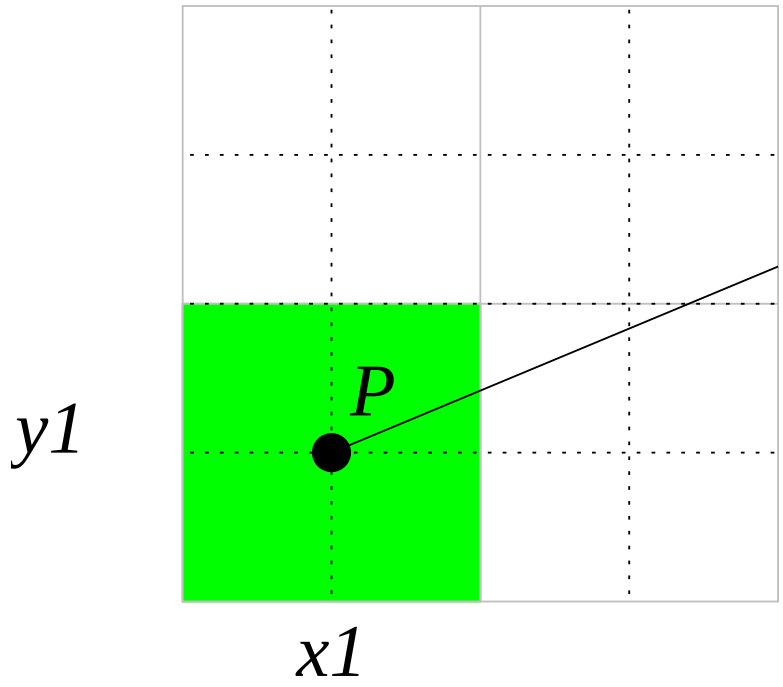
Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

1) **Initialisation**

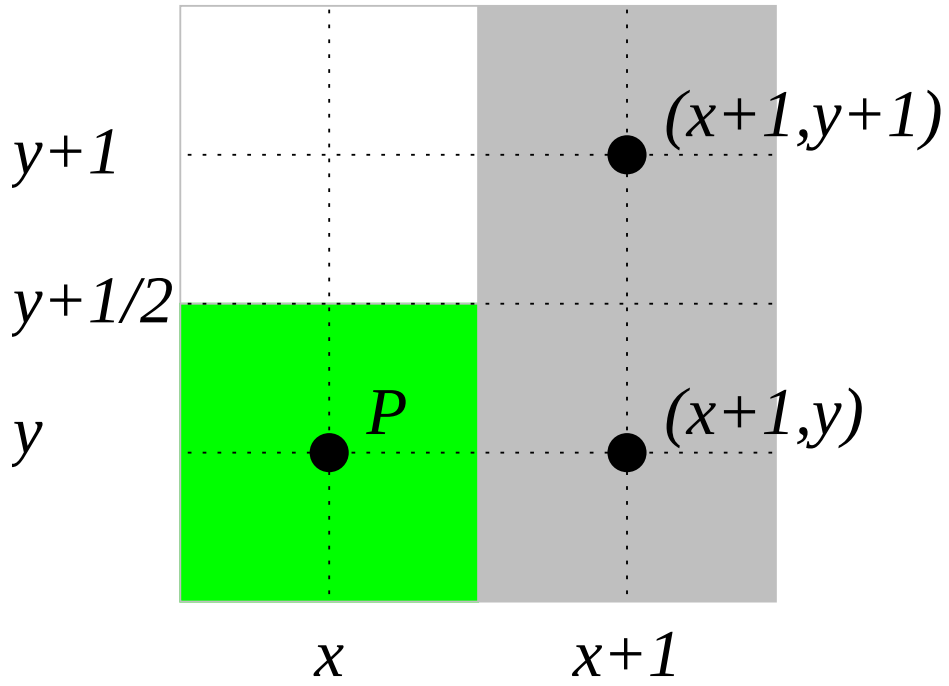


Point initial $P = (x1, y1)$ sur Δ : $FP = F(P) = 0$

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

2) **Boucle**

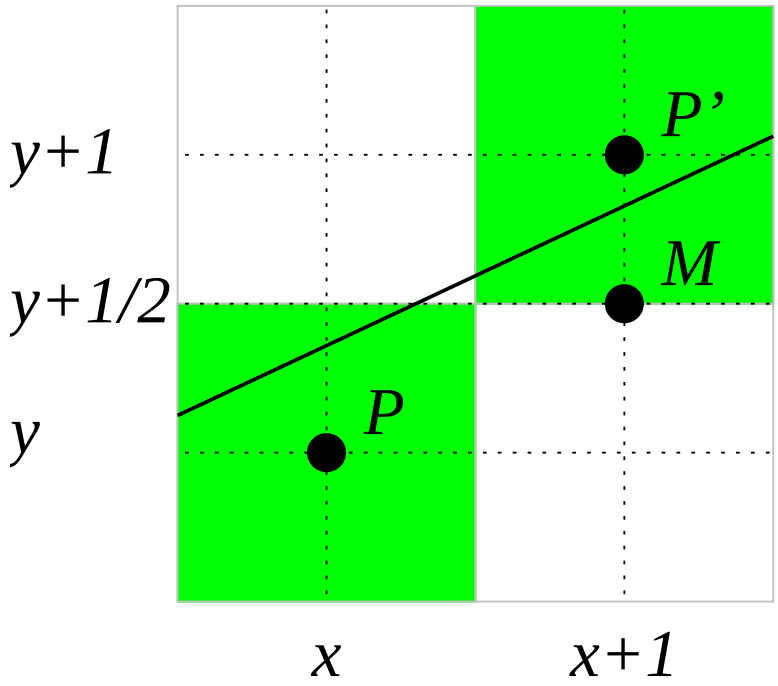


Connaissant $P = (x, y) \rightarrow$ choisir le pixel suivant P'
deux possibilités : $P' = (x + 1, y)$ ou $P' = (x + 1, y + 1)$

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

2) **Boucle**



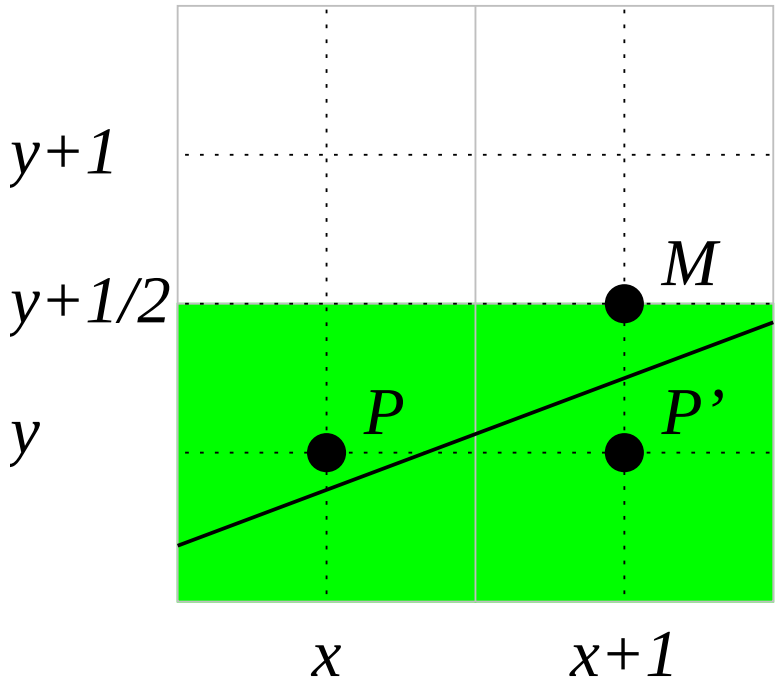
Point M au dessous de Δ : $FM = F(M) = FP + dx - 2\ dy < 0$

$\Rightarrow P' = (x+1, y+1) : F(P') = FP + 2\ dx - 2\ dy$

Segment de droite

Algo. de Bresenham (cas $dx = x2 - x1 \geq dy = y2 - y1 \geq 0$)

2) **Boucle**



Point M au dessus de (ou sur) Δ : $FM = F(M) = FP + dx - 2\ dy \geq 0$

$\Rightarrow P' = (x+1, y)$: $F(P') = FP - 2\ dy$

Segment de droite

Algo. de Bresenham

Cas $x_1 \leq x_2$, $y_1 \leq y_2$, $dx = |x2 - x1| \geq dy = |y2 - y1|$

```
// Initialisation
dx ← |x2 − x1|
dy ← |y2 − y1|
x ← x1
y ← y1

F ← 0
dFM ← dx − 2 dy
dFcas1 ← 2 dx − 2 dy
dFcas2 ← −2 dy

// Boucle principale
tant_que x ≠ x2 faire
    DessinerPixel(x,y)
    si F + dFM < 0 alors
        y ← y + 1
        F ← F + dFcas1
    sinon
        F ← F + dFcas2
    fin_si
    x ← x + 1
fin_tant_que
DessinerPixel(x,y)
```

Segment de droite

Algo. de Bresenham

Cas $x_1 \geq x_2$, $y_1 \leq y_2$, $dx = |x2 - x1| \geq dy = |y2 - y1|$

```
// Initialisation
dx ← |x2 − x1|
dy ← |y2 − y1|
x ← x1
y ← y1

F ← 0
dFM ← dx − 2 dy
dFcas1 ← 2 dx − 2 dy
dFcas2 ← −2 dy
```

```
// Boucle principale
tant_que x ≠ x2 faire
    DessinerPixel(x,y)
    si F + dFM < 0 alors
        y ← y + 1
        F ← F + dFcas1
    sinon
        F ← F + dFcas2
    fin_si
    x ← x − 1
fin_tant_que
DessinerPixel(x,y)
```

Segment de droite

Algo. de Bresenham

Cas $x_1 \leq x_2$, $y_1 \leq y_2$, $dx = |x2 - x1| \geq dy = |y2 - y1|$

```
// Initialisation
dx ← |x2 − x1|
dy ← |y2 − y1|
x ← x1
y ← y1

F ← 0
dFM ← dx − 2 dy
dFcas1 ← 2 dx − 2 dy
dFcas2 ← −2 dy

// Boucle principale
tant_que x ≠ x2 faire
    DessinerPixel(x,y)
    si F + dFM < 0 alors
        y ← y + 1
        F ← F + dFcas1
    sinon
        F ← F + dFcas2
    fin_si
    x ← x + 1
fin_tant_que
DessinerPixel(x,y)
```

Segment de droite

Algo. de Bresenham

Cas $x_1 \leq x_2$, $y_1 \leq y_2$, $dx = |x2 - x1| \leq dy = |y2 - y1|$

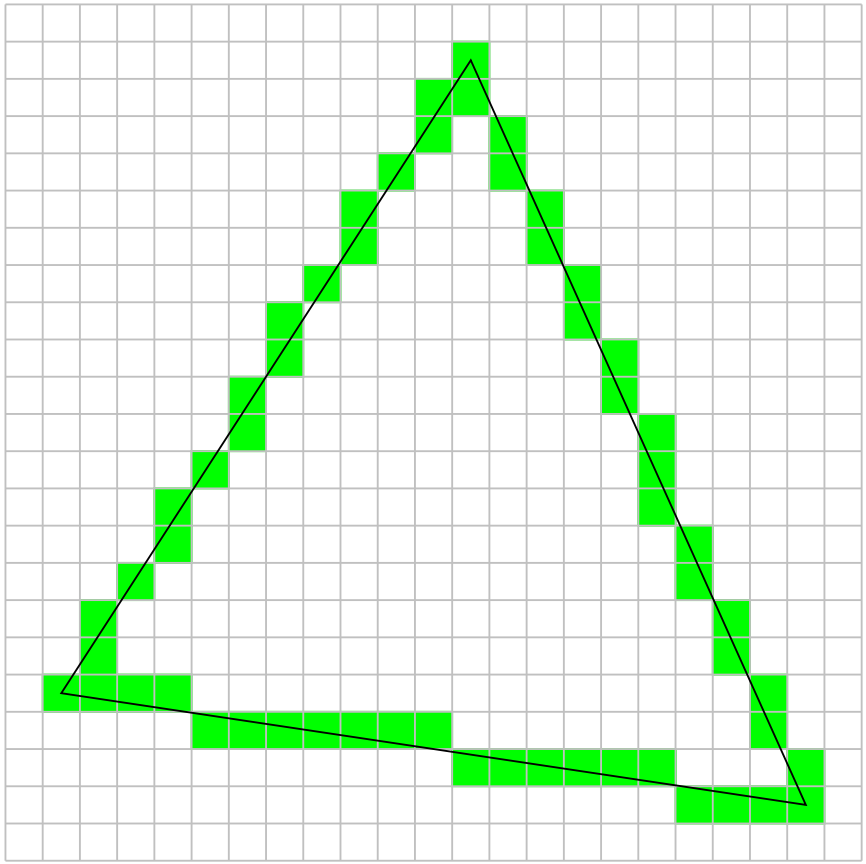
```
// Initialisation
dx ← |x2 − x1|
dy ← |y2 − y1|
x ← x1
y ← y1

F ← 0
dFM ← dy − 2 dx
dFcas1 ← 2 dy − 2 dx
dFcas2 ← −2 dx
```

```
// Boucle principale
tant_que y ≠ y2 faire
    DessinerPixel(x,y)
    si F + dFM < 0 alors
        x ← x + 1
        F ← F + dFcas1
    sinon
        F ← F + dFcas2
    fin_si
    y ← y + 1
fin_tant_que
DessinerPixel(x,y)
```

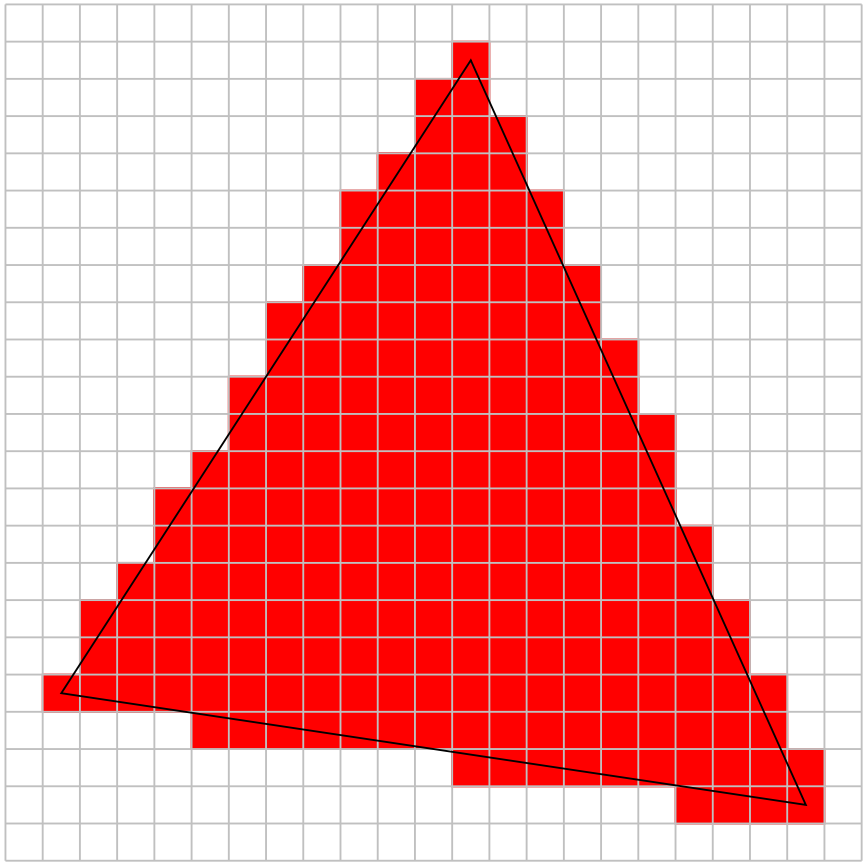
Triangle

Triangle



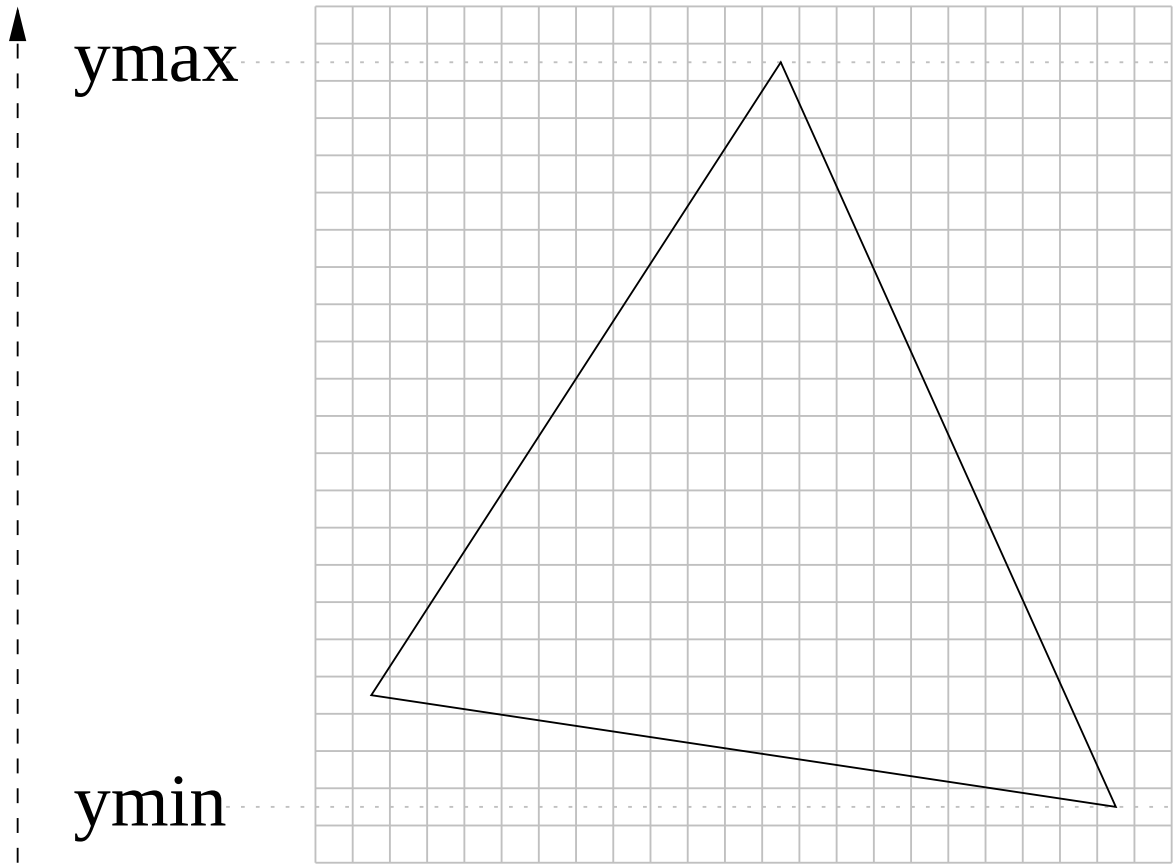
Tracé du bord du triangle

Triangle



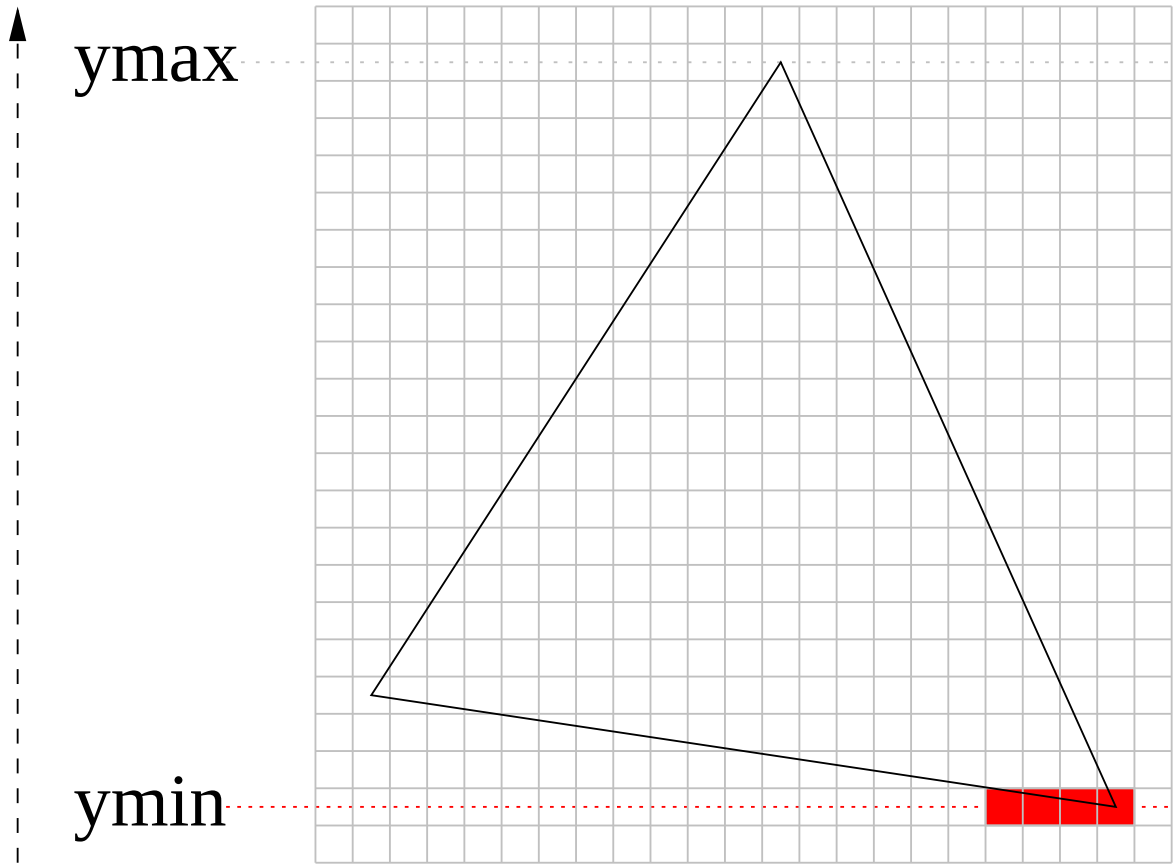
Remplissage du triangle

Triangle



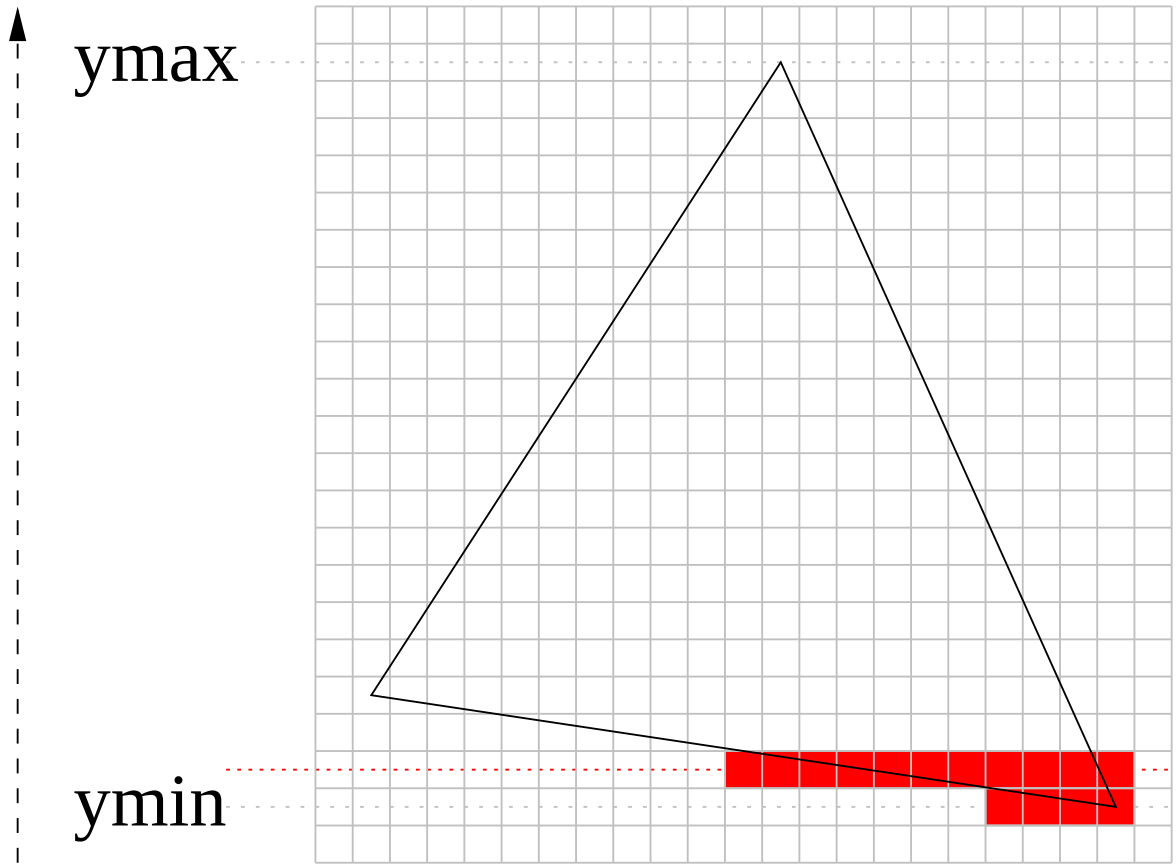
Remplissage par balayage horizontal

Triangle



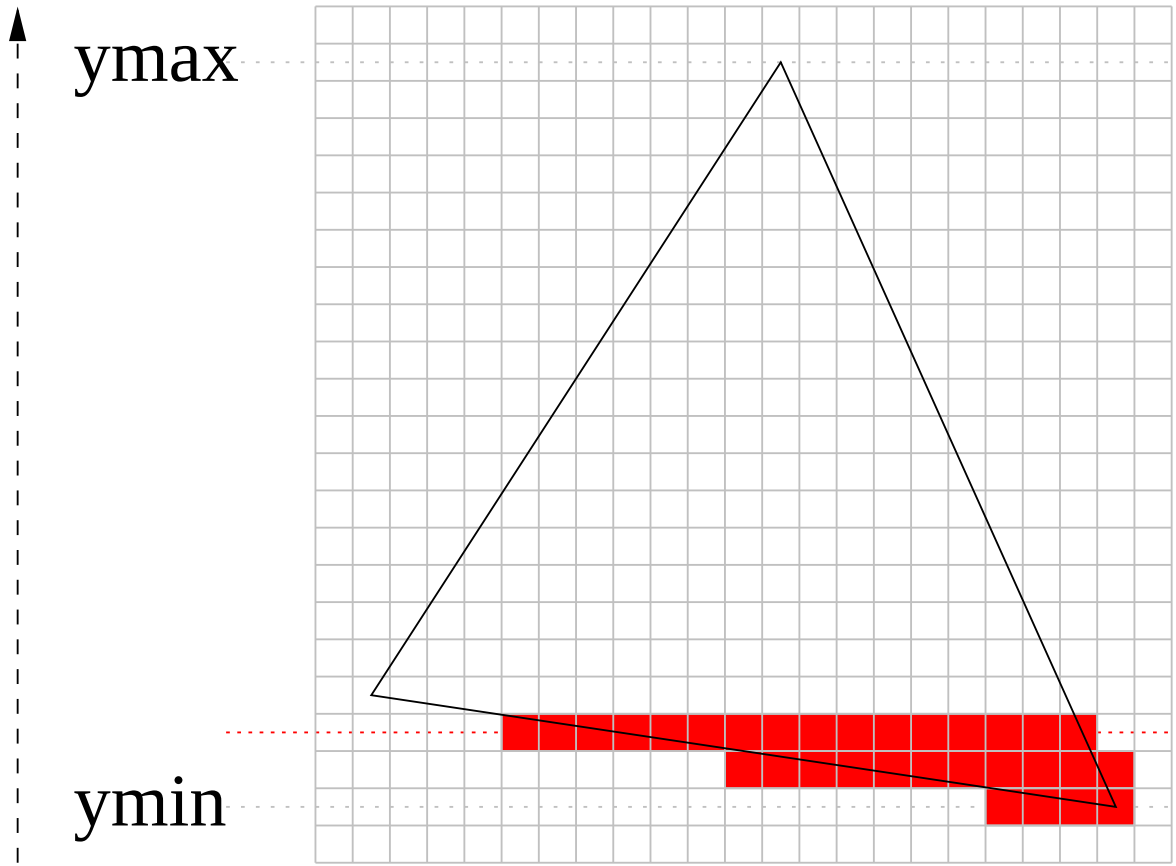
Remplissage par balayage horizontal

Triangle



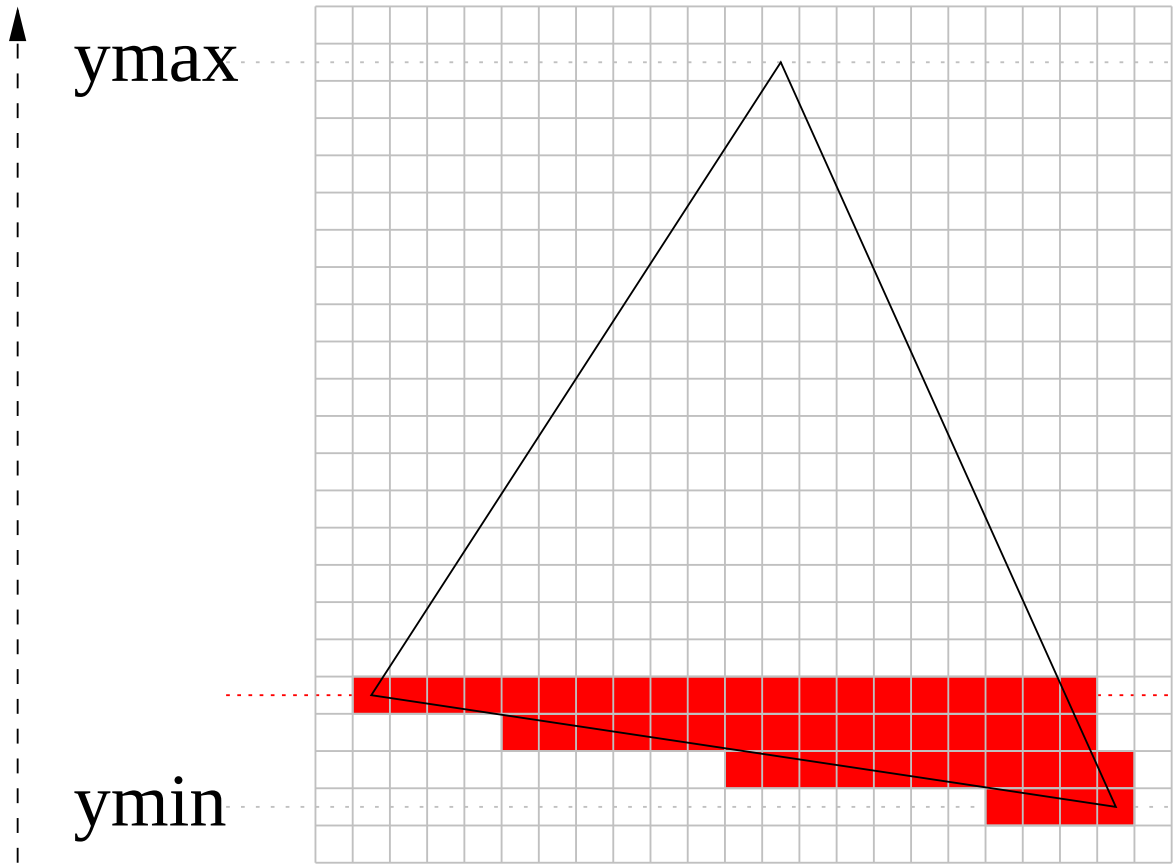
Remplissage par balayage horizontal

Triangle



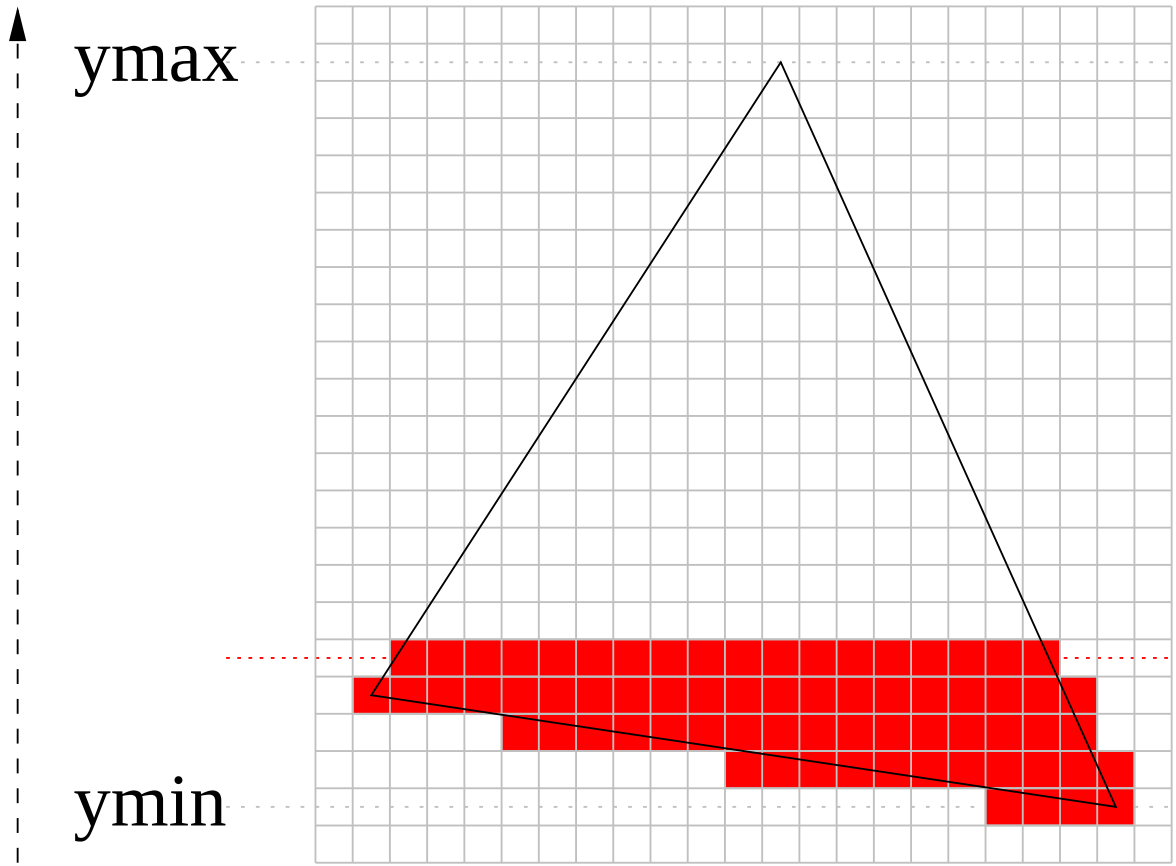
Remplissage par balayage horizontal

Triangle



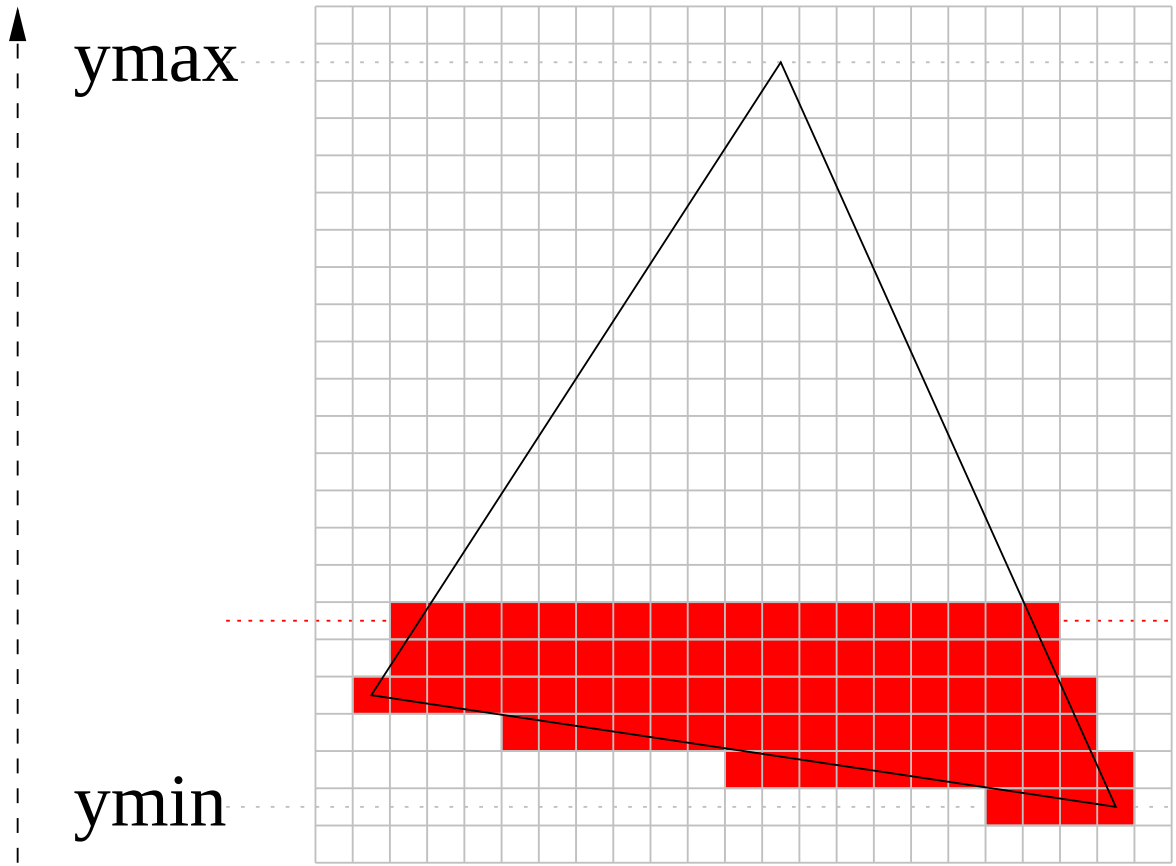
Remplissage par balayage horizontal

Triangle



Remplissage par balayage horizontal

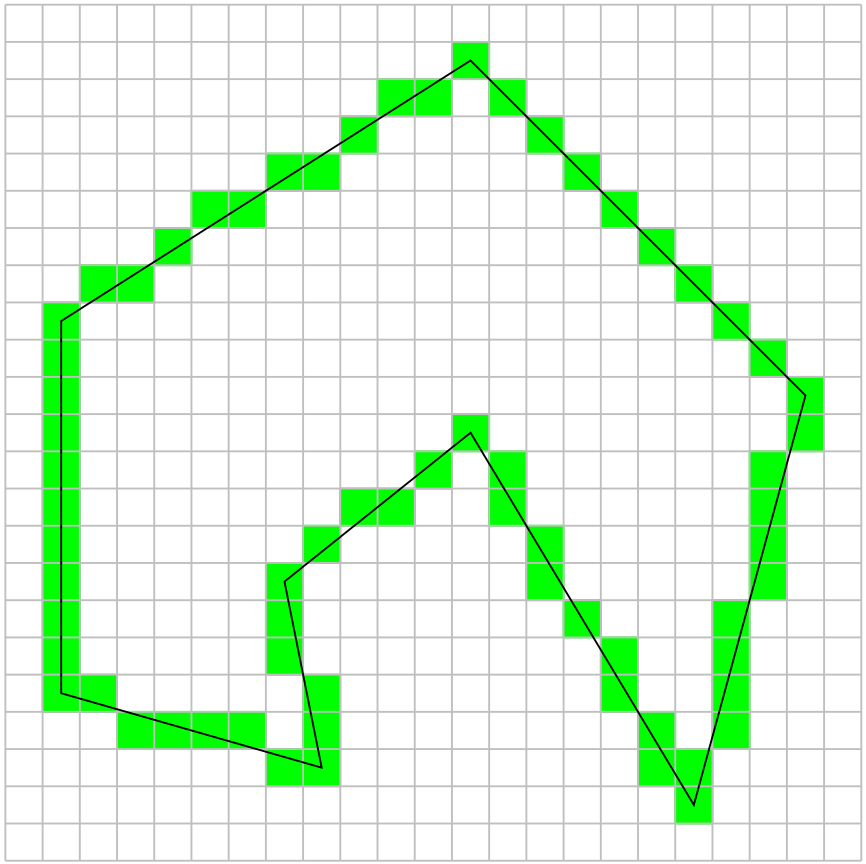
Triangle



Remplissage par balayage horizontal

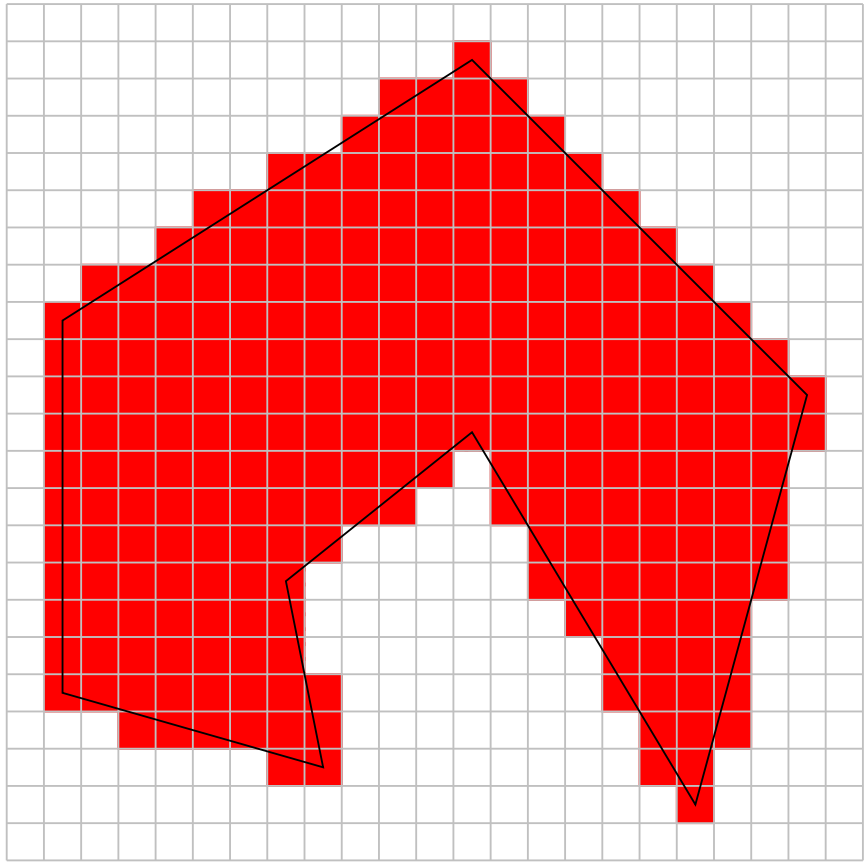
Polygone

Polygone



Tracé du bord du polygone

Polygone



Remplissage du polygone