

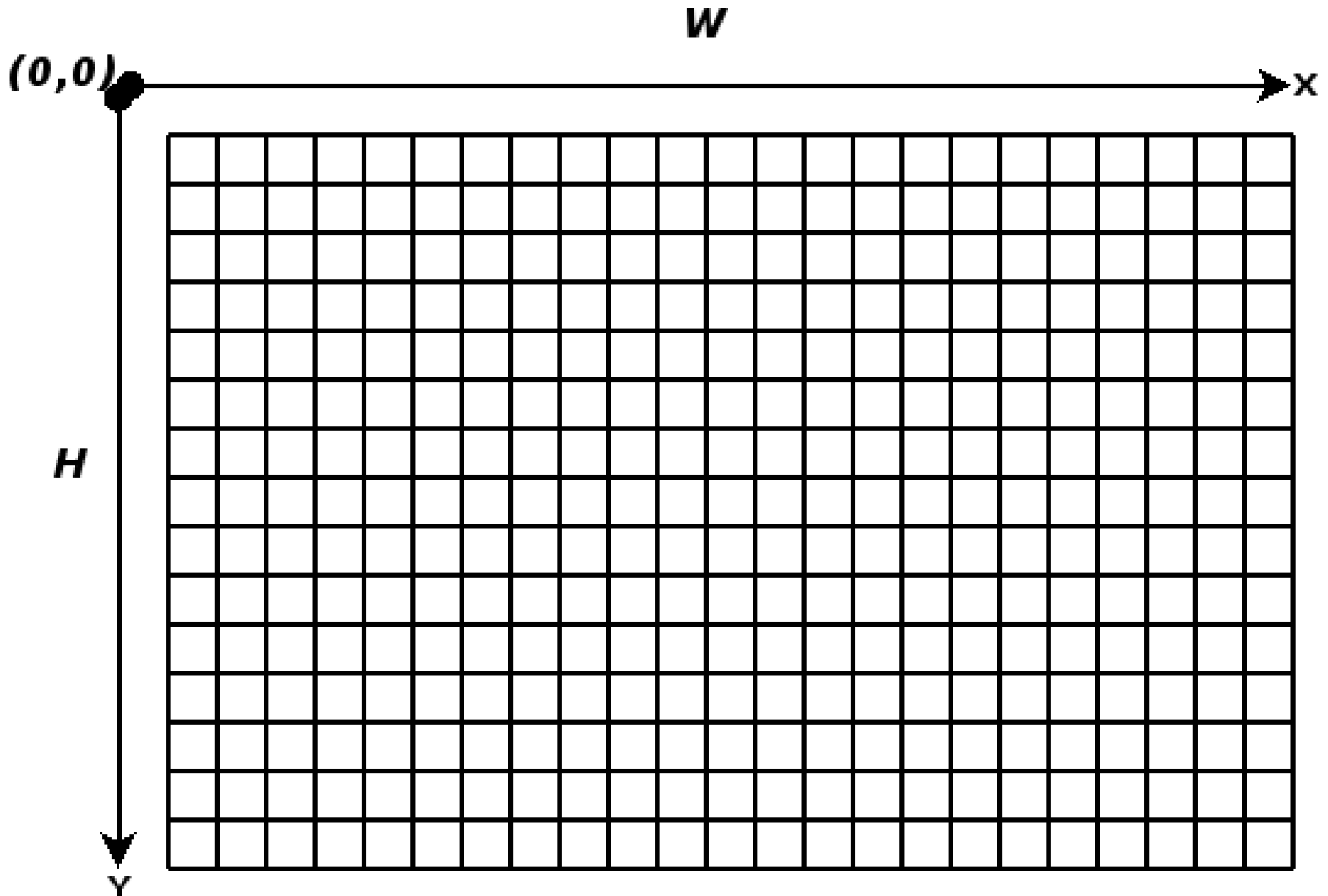
Líneas y curvas

Introducción a la Computación Gráfica

Leonardo Flórez-Valencia

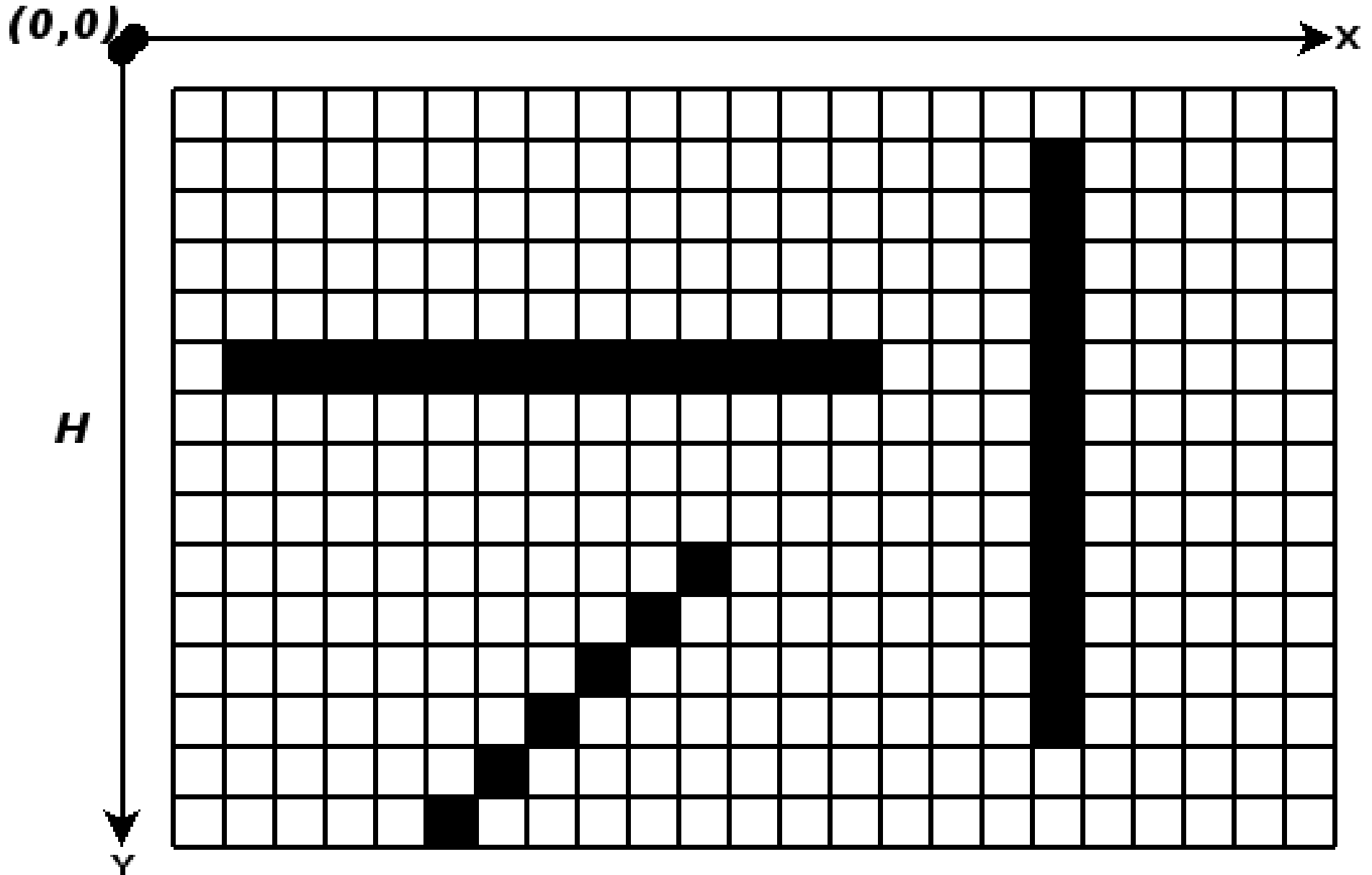
Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas

Una pantalla 2D cualquiera

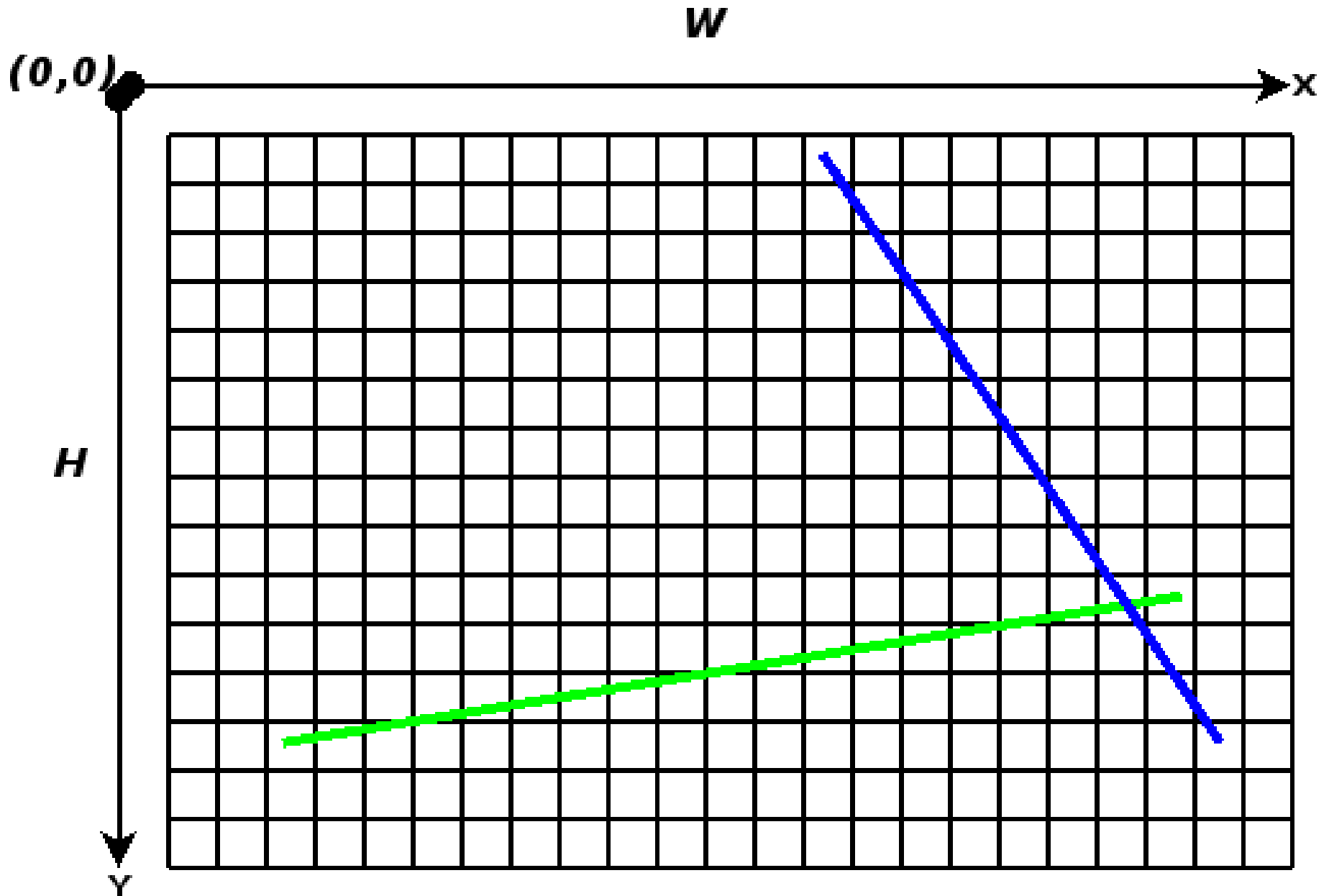


Una pantalla 2D cualquiera

W



Una pantalla 2D cualquiera



Problema

- Encender los *mejores píxeles* (i.e. lugares **discretos**) asociados a una línea **continua**.
- Entradas:
 - Ventana de visualización: W y H (se puede suponer que es lo suficientemente grande).
 - Segmento de línea continuo:

$$\mathbf{p}_0 = (x_0, y_0) \text{ y } \mathbf{p}_1 = (x_1, y_1)$$

- Salidas:

$$R = [\mathbf{q}_i : 0 \leq i \leq n \wedge \mathbf{q}_i \in \mathbb{Z}^2 \wedge \mathbf{q}_i \simeq \overline{\mathbf{p}_0 \mathbf{p}_1} \forall i \in [1, n-1]]$$

Primitiva 2D: línea

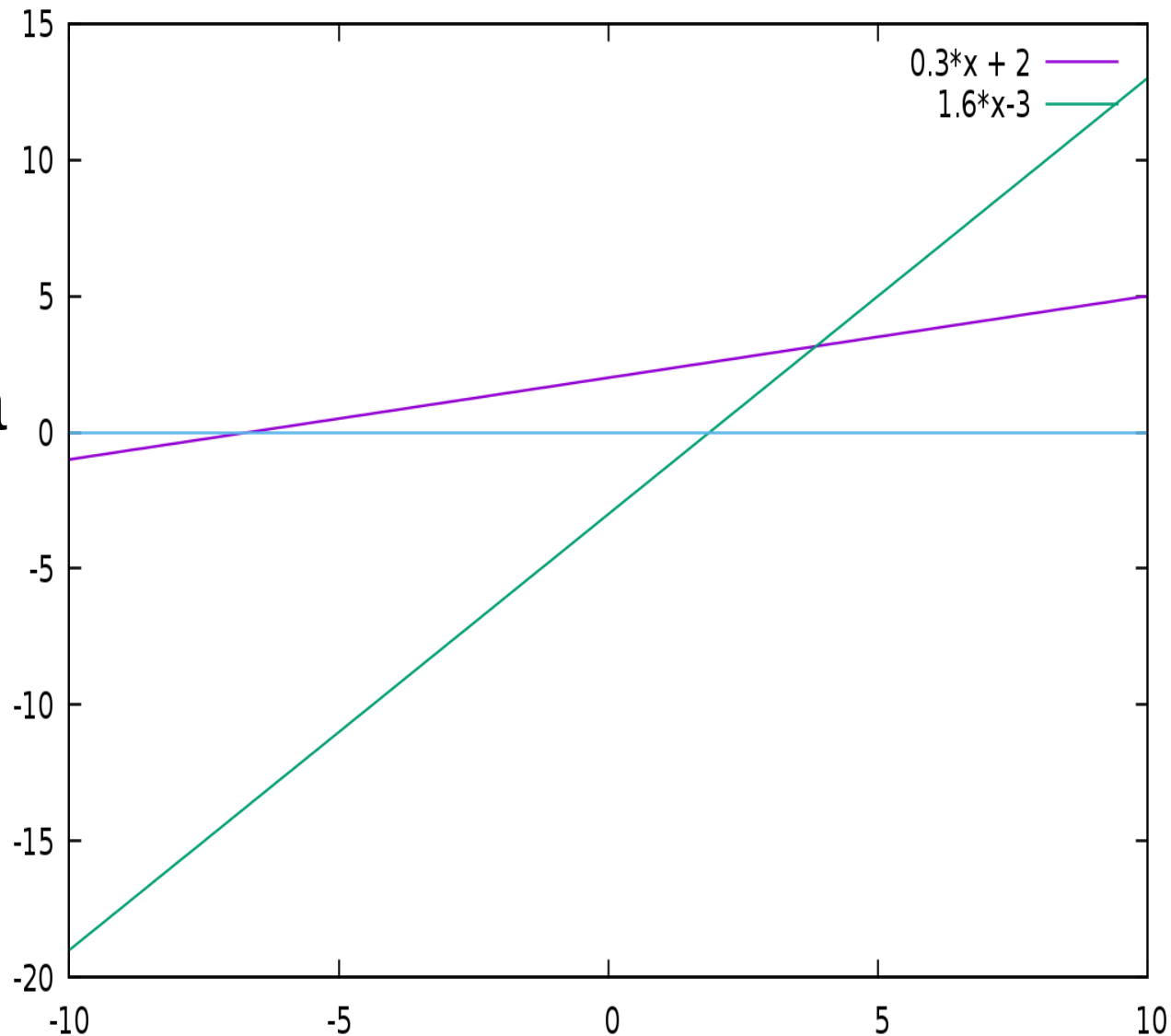
- Línea continua

$$y = mx + b$$

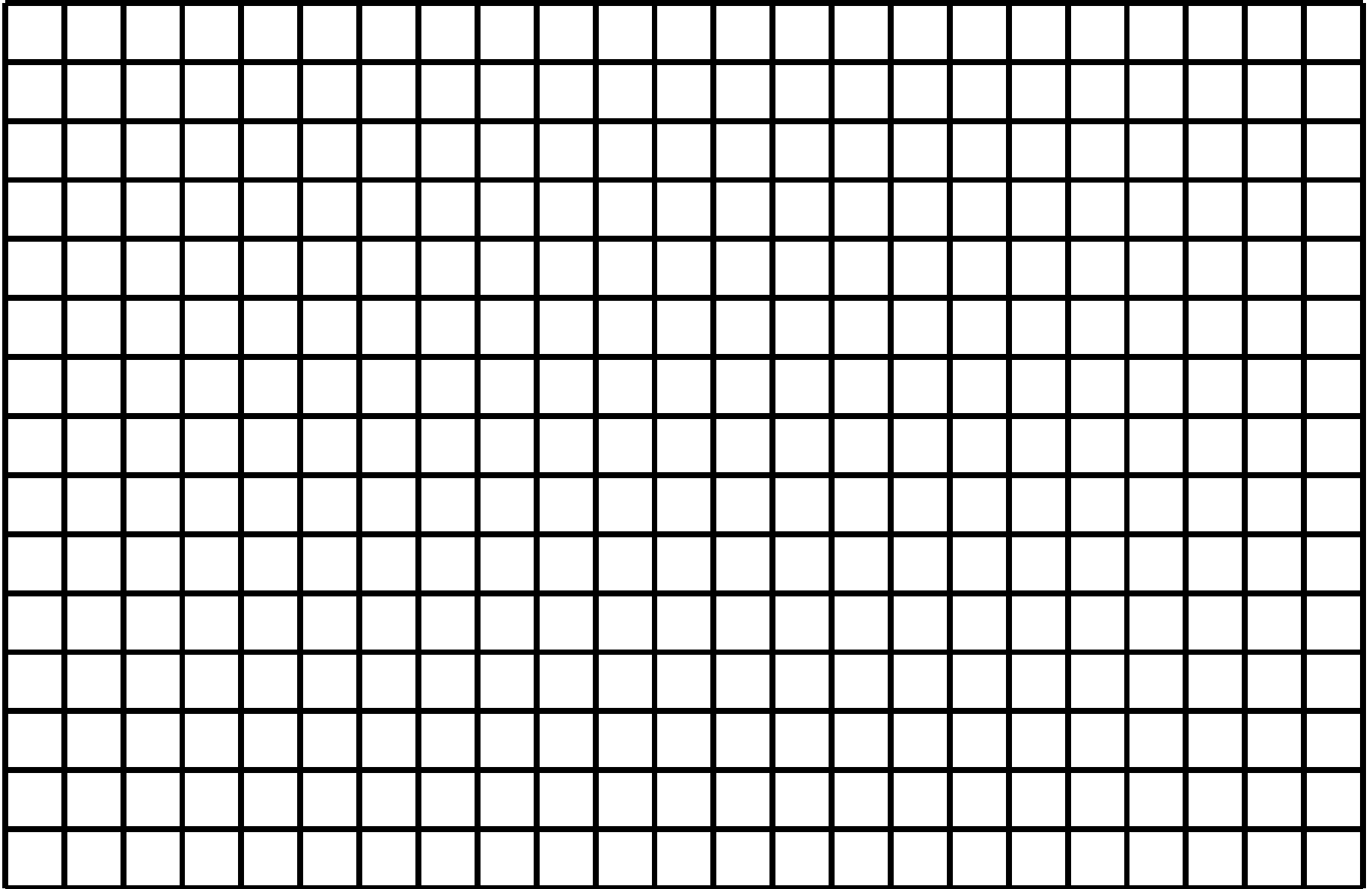
- Segmento de línea

$$m = \frac{y_1 - y_0}{x_1 - x_0}$$

$$b = y_0 - mx_0$$



Una pantalla 2D cualquiera



Digital differential analyzer (DDA), cuadrante 0

procedure DDA_0(x0, y0, x1, y1)

dx \leftarrow x1 - x0

dy \leftarrow y1 - y0

if |dy| < |dx| **then**

 step \leftarrow |dx|

else

 step \leftarrow |dy|

end if

dx \leftarrow dx / step

dy \leftarrow dy / step

...

procedure DDA_0(x0, y0, x1, y1)

...

x \leftarrow x0

y \leftarrow x1

for i \leftarrow 1 to step do

R \leftarrow R U (x,y)

x \leftarrow x + dx

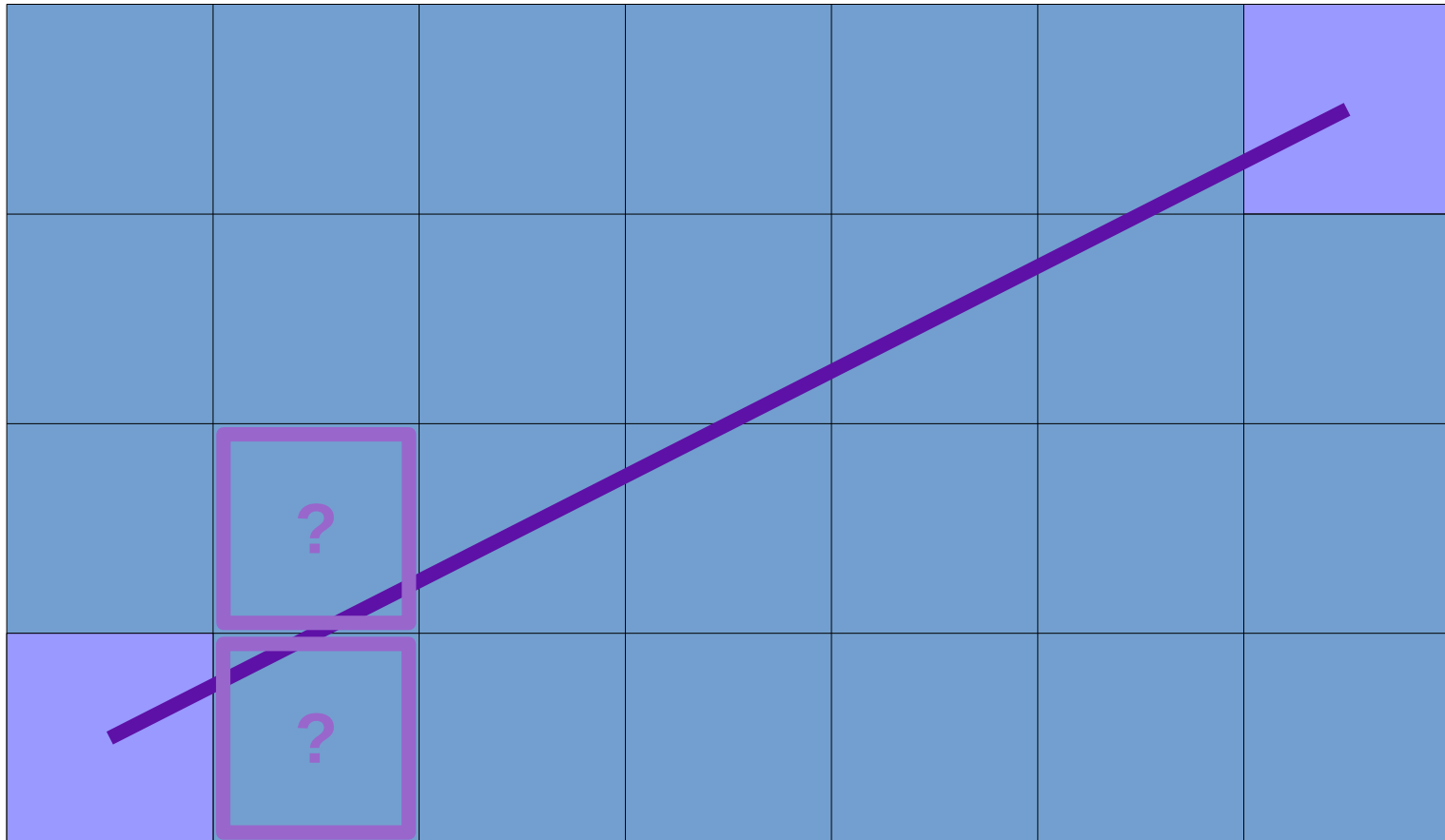
y \leftarrow y + dy

end for

return R

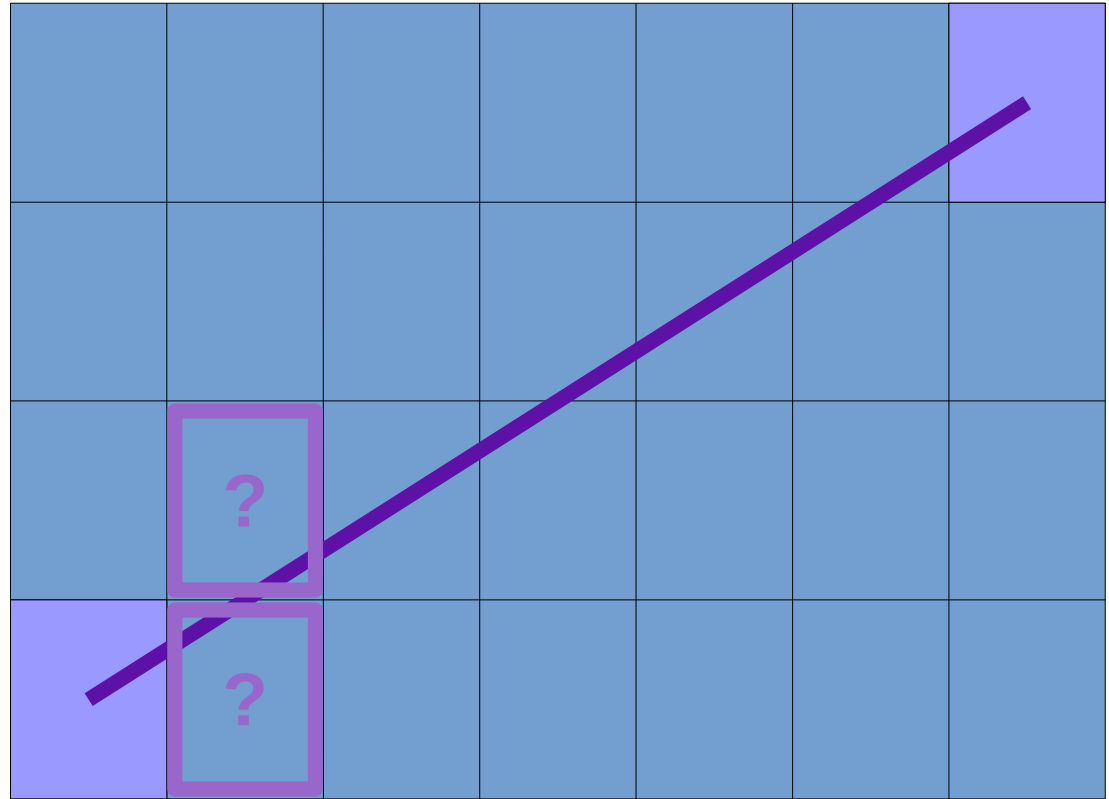
end procedure

Problema con DDA

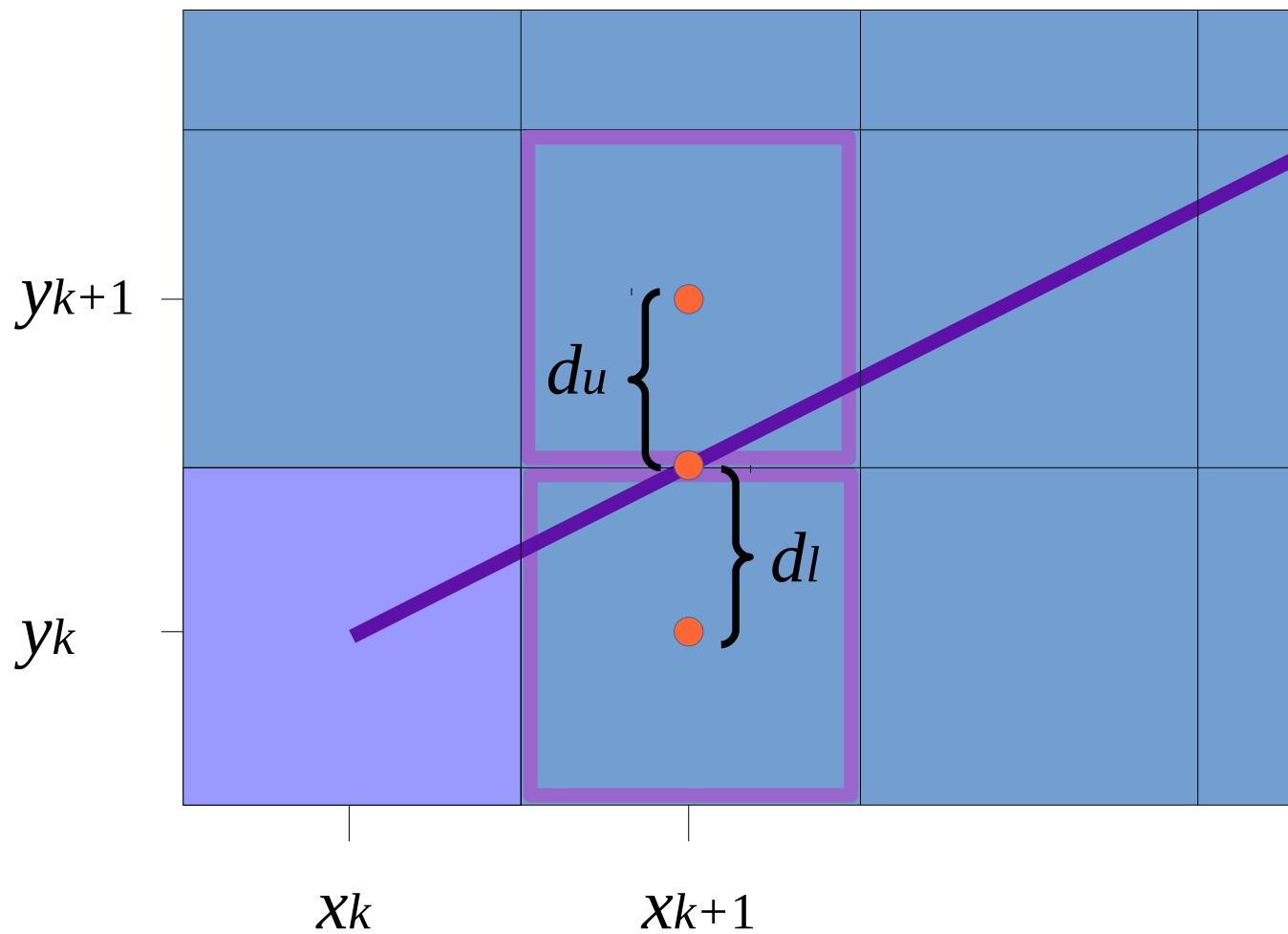


Problema con DDA

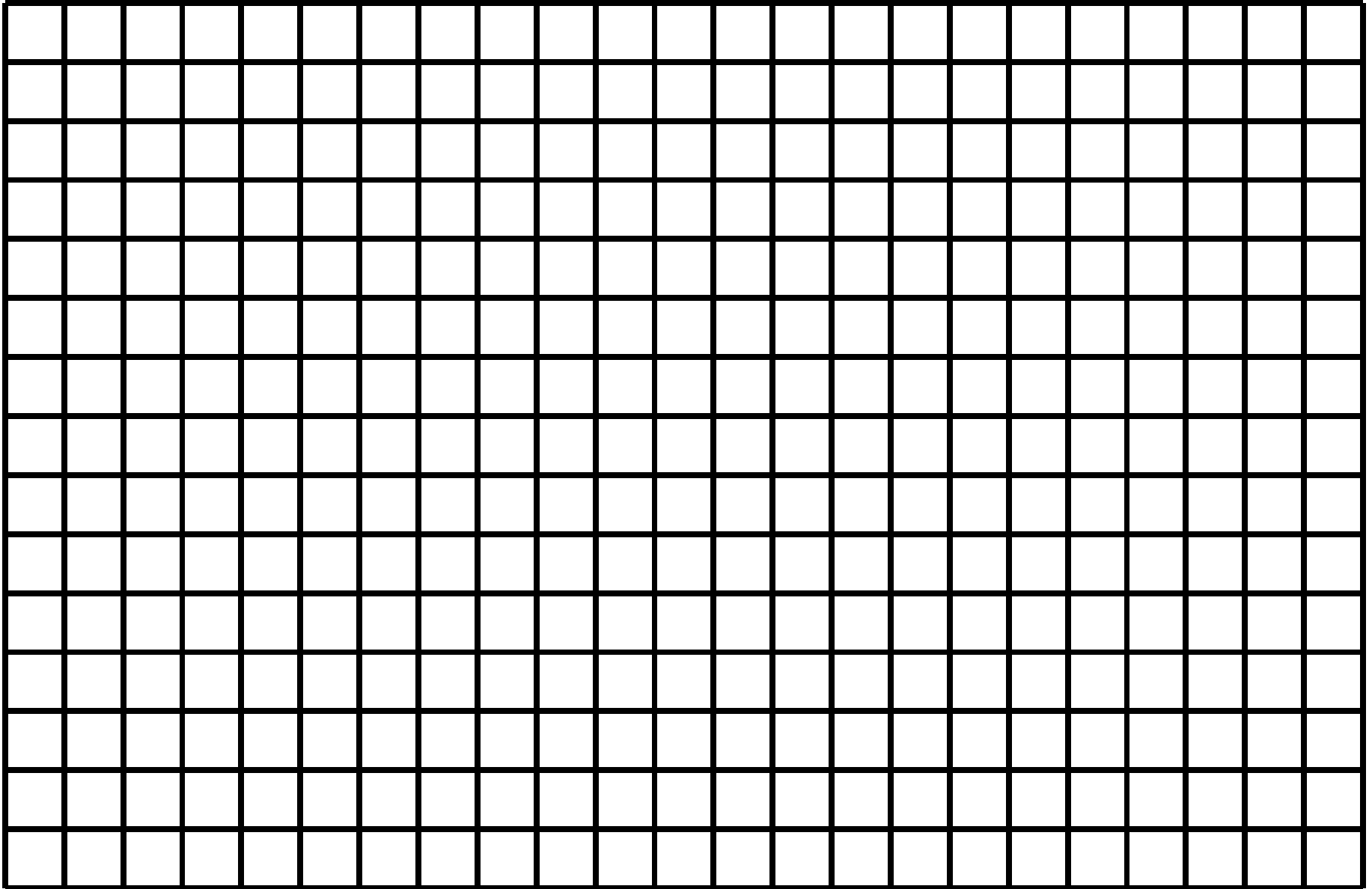
- Conociendo (x_k, y_k)
 - $\mathcal{I}(x_k+1, y_k)$?
 - $\mathcal{I}(x_k+1, y_k+1)$?



Algoritmo de Bresenham



Una pantalla 2D cualquiera



Algoritmo de Bresenham, cuadrante 0

procedure Bresenham_0(x_0 , y_0 , x_1 , y_1)

$dx \leftarrow x_1 - x_0$

$dy \leftarrow y_1 - y_0$

$y_i \leftarrow 1$

if $dy < 0$ **then**

$y_i \leftarrow -1$

$dy \leftarrow -dy$

end if

$D \leftarrow 2dy - dx$

$y \leftarrow y_0$

...

procedure Bresenham_0(x0, y0, x1, y1)

...

for x \leftarrow x0 **to** x1 **do**

 R \leftarrow R U (x,y)

if D > 0 **then**

 y \leftarrow y + yi

 D \leftarrow D - 2dx

end if

 D \leftarrow D + 2dy

end for

return R

end procedure

Taller dibujado de líneas

- (10%) Implementar el algoritmo DDA en C++.
- (10%) Implementar el algoritmo de Bresenham en C++.
- (30%) Crear una aplicación en C++/OpenGL que:
 - reciba por la línea de comandos los parámetros W y H (ancho y alto) de la ventana e
 - interactivamente reciba 2 clicks del ratón y trace las dos líneas correspondientes (rojo: píxeles únicos de DDA, verde: píxeles únicos de Bresenham, azul: píxeles compartidos).
- (20%) Escribir un reporte corto acerca de la comparación de los resultados de los dos algoritmos.
- (30%) Quiz próxima clase **motivado** en la pregunta: ¿cómo usar estas ideas para dibujar círculos y elipses?