

Tracé de lignes entre deux points

Nicolas FLASQUE

nicolas.flasque@efrei.fr

Rappel de la situation

Une ligne (line) est définie par deux points A et B , de coordonnées (x_A, y_A) et (x_B, y_B) . Ces coordonnées sont des coordonnées entières.

On souhaite trouver **quels pixels** composent la ligne à dessiner entre A et B .

Les pixels ont donc des coordonnées entières : ce sont des points.

Une ligne est donc un ensemble de points.

La ligne entre A et B étant la même que celle entre B et A, on choisit donc que $x_A \leq x_B$

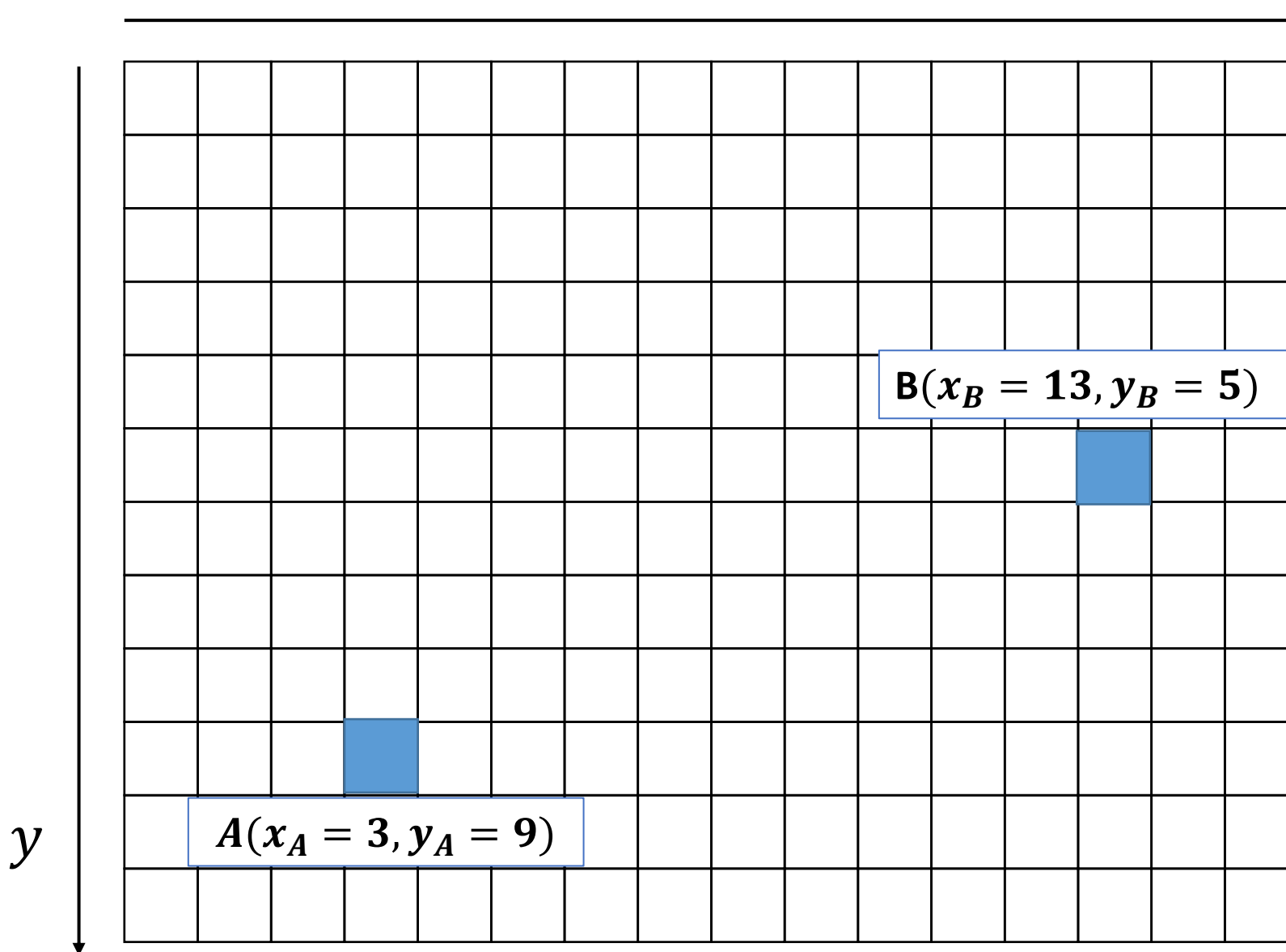
On calcule $dx = x_B - x_A$ et $dy = y_B - y_A$. On a $dx \geq 0$

On va chercher à savoir combien de segments (imaginer cela comme des marches d'escalier) droits on doit tracer pour matérialiser la ligne de A à B :

On calcule $dmin = \min(dx, |dy|)$ et $dmax = \max(dx, |dy|)$

$dmin$ est la plus petite différence entre les coordonnées, $dmax$ est la plus grande différence entre les coordonnées.

Notre exemple



Dans un repère orienté, où y augmente de haut en bas de l'image :

$$dx = x_B - x_A = 13 - 3 = 10$$

$$dy = y_B - y_A = 5 - 9 = -4$$

$$dmin = \min(dx, |dy|) = \min(10, 4) = 4$$

$$dmax = \max(dx, |dy|) = \max(10, 4) = 10$$

Le nombre de segments à 'dessiner' est alors $(dmin + 1)$

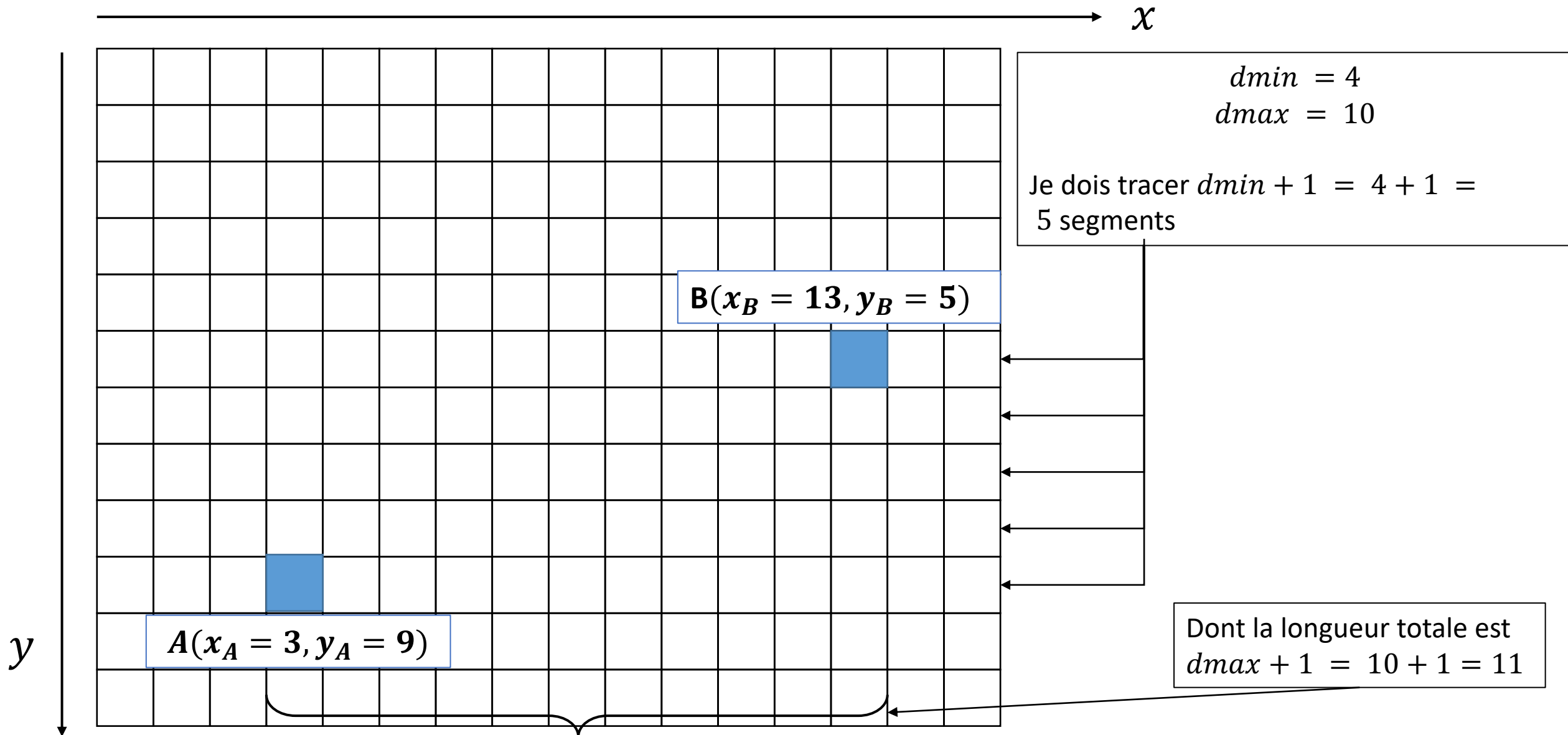
Par exemple, si $dmin$ vaut 0, alors il faut tracer un seul segment qui relie les deux points, qui sont alignés horizontalement ou verticalement.

On utilise une variable entière à cet effet : $nb_segs = (dmin + 1)$

Combien de points (pixels) dans chaque segment ?

L'idée est de répartir de manière équilibrée, le nombre de points : il y en a au moins : $(dmax + 1) / (dmin + 1)$

Notre exemple



Première partie : calculons la taille 'de base' du segment :

Elle nous est donnée par : $(d_{max} + 1)/(d_{min} + 1)$

d_{max} et d_{min} étant des entiers, en C, c'est une division entière, donc on obtient un entier, qui est la taille 'de base' de chaque segment.

Sur notre exemple, on calcule $11 / 5$: la taille totale divisée par le nombre de segments : $11 / 5 = 2$ (division entière)

On aura donc 5 segments de taille 2 (*pixels*) comme base de tracé de notre ligne

on crée donc un tableau *segments* comportant *nb_segs* éléments, et on initialise tous ses éléments avec la taille de de base : ici, un tableau avec 5 éléments, chaque élément valant 2

On cherche ensuite à répartir les pixels manquants sur les segments :

Il nous reste donc **restants** = $(dmax + 1) \% (dmin + 1)$ pixels à répartir

Sur notre exemple : **restants** = $11 \% 5 = 1$ pixel à répartir. Quel segment va accueillir ce pixel ?

Pour ce faire, on calcule le cumul des restants : on crée un tableau qui indiquera combien de pixels on doit ajouter à chaque segment (ce tableau contiendra des 0 et des 1)

Pixels à répartir (suite) : Ce code permet de calculer le nombre de pixels restants et met à jour le tableau des segments.

On suppose que l'on a le tableau `segments`

```
int *cumuls = (int *)malloc(nb_segs*sizeof(int));
cumuls[0]=0;
for (int i = 1; i < nb_segs;i++)
{
    cumuls[i] = ((i*restants)%(dmin+1) < ((i-1)*restants)%(dmin+1));
    segments[i] = segments[i]+cumuls[i];
}
```

On connaît maintenant les segments reliant A à B et leur taille.

Pour le tracé : on part des coordonnées de A et on trace segment par segment : il faut savoir s'ils sont horizontaux ou verticaux

Il faut savoir si on trace 'vers le haut' ou 'vers le bas'

Si $dy < 0$

on trace vers le bas

Si $dx > |dy|$

les segments sont horizontaux (on les parcourt en augmentant x)

sinon

les segments sont verticaux (on les parcourt en diminuant y)

Sinon

on trace vers le haut

Si $dy < 0$ // on trace vers le bas

Si $dx > |dy|$

les segments sont horizontaux (on les parcourt en augmentant x)
à chaque changement de segment, on diminue y

Sinon

les segments sont verticaux (on les parcourt en diminuant y)
à chaque changement de segment, on augmente x

Sinon // on trace vers le haut

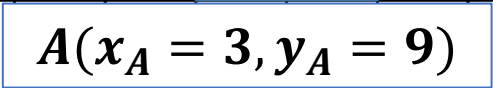
Si $dx > dy$

les segments sont horizontaux (on les parcourt en augmentant x)
à chaque changement de segment, on augmente y

Sinon

les segments sont verticaux (on les parcourt en augmentant y)
à chaque changement de segment, on augmente x

A horizontal line with an arrow pointing to the right, labeled with the variable x .


$$\mathbf{B}(x_B = 13, y_B = 5)$$

Segments : [2 ,2 ,2 ,2 ,3]

$$dx = 10, dy = -4$$

$dx > |dy|$ donc segments horizontaux

Enfin, pour tracer les segments :

On utilise une double boucle :

Le point de départ est A

Pour i de 0 à nb_segs-1

 pour j de 0 à segments[i]

 Ajouter au tableau des pixels les coordonnées des pixels (donc augmenter ou diminuer x ou y) en fonction des situation décrites dans le slide 11.

 passer au segment suivant (donc augmenter ou diminuer x ou y)