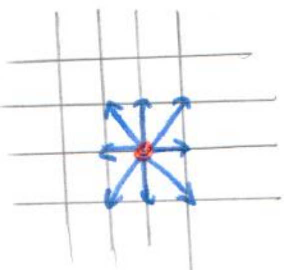
Tracé de droites discrètes II

4 Chercher ailleurs et autrement

Dans un article, Xiaolin Wu fait les observations suivantes :



1. Pour tracer une droite, il y a 8 mouvements possibles mais sur une droite donnée, on en utilisera que deux.

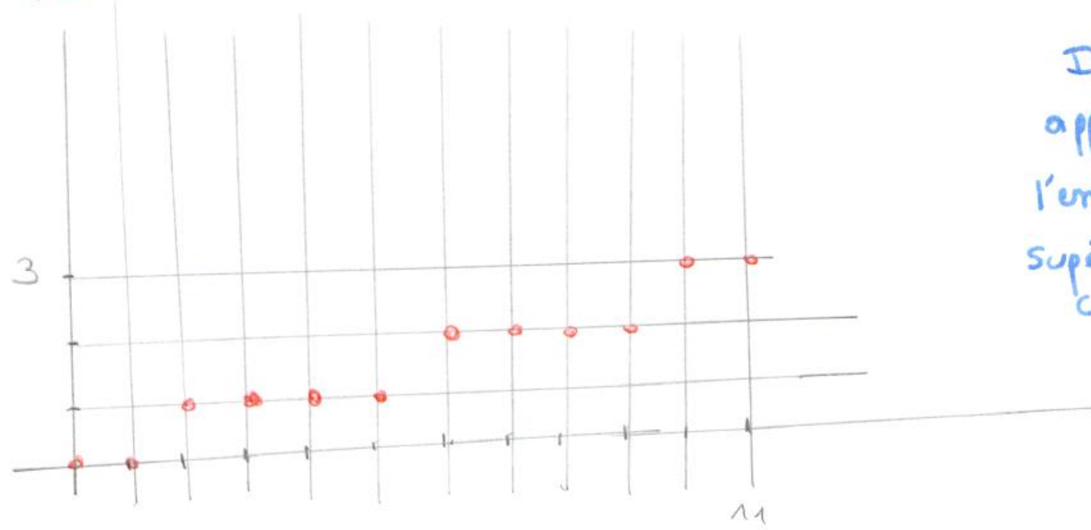
   Ex : Dans le premier octant, mouvement horizontal et oblique.

2. Sur ces deux mouvements, l'un apparaît toujours de façon solitaire et l'autre de façon groupée

Dans la partie basse :
Solitaire ⤳ Oblique
Plage ⤳ horizontal

Dans la partie haute
Solitaire ⤳ Horizontal
Plage ⤳ Oblique

Ex : Droite $\mathcal{D}(11,3)$ $dx = 11$, $dy = 3$



Droite avec approximation à l'entier le plus proche, supérieur dans les cas limites

Cette droite est bien dans la partie basse car $11 > 2 \times 3$ (2)

Rappels : Partie haute : $0 < dy < dx < 2dy$

Partie basse : $0 < \cancel{dy} < 2dy < dx$

Le mouvement horizontal est bien groupé !

(3) Le mouvement solitaire est bien réparti ; les plages ont toute la même longueur à 1 près (hormis éventuellement les plages extremales)

■ Mot de trace d'une droite

Dans le premier octant, on a soit

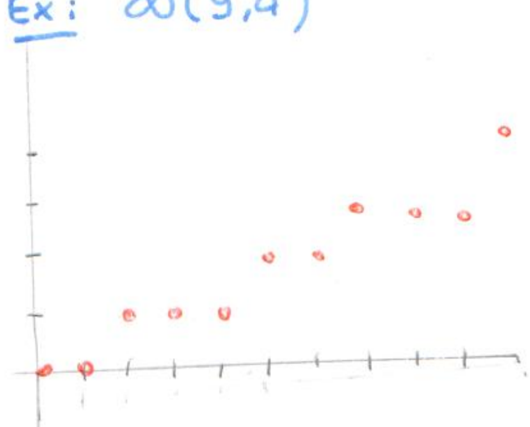
- mouvement horizontal $\rightsquigarrow 0$
- mouvement oblique $\rightsquigarrow 1$

On peut donc représenter le chemin suivi par un mot formé de 0 et de 1 qui indiquent le mouvement.

Pour reconstituer la droite à partir du mot : x augmente à chaque fois et y augmente du bit correspondant.

Ce mot, noté $w(dx, dy)$, est appelé mot de trace de la droite.

Ex : $\mathcal{D}(9,4)$



Mot de trace :

0100101001

Remarques : Soit w le mot de trace d'une droite $\mathcal{D}(dx, dy)$

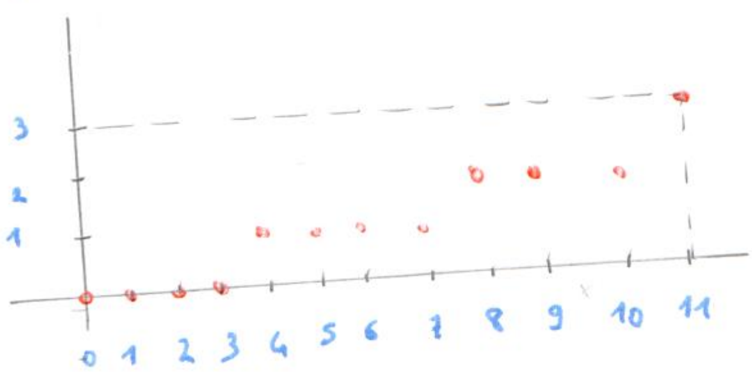
Longueur : $|w| = dx$

Nombre de 1 : $|w|_1 = dy$

Nombre de 0 : $|w_0| = dx - dy$

Remarque : Les différents choix pour tracer la droite (partie entière, partie supérieure, meilleur approximation) donnent des mots de trace différents. Pour avoir des plages bien visibles et un mot de trace qui se termine par un 1, il faut prendre l'approximation à la partie entière.

Ex : $\mathcal{D}(11, 3)$ avec partie entière



Mot de trace
$w = 00010001001$

Nombre de plages : dy (le nombre de suites de 0 avant d'arriver au dernier 1)

Longueur d'une plage : $\underbrace{\left\lfloor \dfrac{dx - dy}{dy} \right\rfloor}_{\text{plage courte}}$ ou $\underbrace{\left\lfloor \dfrac{dx - dy}{dy} \right\rfloor + 1}_{\text{plage longue}}$

Question : Combien de plages composent le mot de trace ?

Si on suppose qu'il y a seulement des plages courtes dans le mot, on remplit

$$dy \times \left\lfloor \frac{dx - dy}{dy} \right\rfloor$$

fois 0 dans le mot.

quotient de la division euclidienne de $dx - dy$ par dy.

Puisqu'on a au total $dx - dy$ zéros, il reste comme 0 à remplir :

$$dx - dy \quad - dy \times \left\lfloor \frac{dx - dy}{dy} \right\rfloor$$

Remarque : Pour a et b des nombres entiers, on a

$$a - b \times \left\lfloor \frac{a}{b} \right\rfloor = a \% b$$

reste dans la division euclidienne de a par b

quotient de a par b

Donc il reste comme 0 à remplir :

$$(dx - dy) \% dy = dx \% dy$$

$a = dx - dy$
$b = dy$

Conclusion : Nombre de plages longues $= dx \% dy$

Nombre de plages courtes $= dy - dx \% dy$

(La somme des deux doit donner dy qui est le nombre de plages)

Exemple : Droite $\mathcal{D}(11, 3)$

$11 \% 3 = 2$

$3 - 11 \% 3 = 1$

$\rightsquigarrow$ 1 plage courte Longueur 2

2 plages longues Longueur 3

A partir de ces observations, nous allons voir 3 algorithmes qui permettent de tracer une droite discrète à partir d'opérations sur les mots de trace: (5)

1. Dulucq / Bourdin
2. Green / Pitteway et Castle / Pitteway
3. Berstel

1. Dulucq / Bourdin

1_ Dans la partie basse de l'octant ($dx > 2dy$), on crée une fonction

$$\varphi_m : \{\text{Mots sur } 0,1\} \to \{\text{Mots sur } 0,1\}$$

$$0 \mapsto 0^{m+1}1$$

$$1 \mapsto 0^{m}1$$

où $0^{m+1}1 = \underbrace{00\ldots0}_{m+1 \text{ fois}}1$

Puis on montre alors que

$$\omega(dx, dy) = \varphi_m(\omega(p,q)) \text{ où } \begin{cases} p = dy \\ q = dy - dx \% dy \\ m = \left\lfloor \dfrac{dx - dy}{dy} \right\rfloor \end{cases}$$

Exemple: $\varphi_2(010) = \text{Concaténation}(\varphi_2(0), \varphi_2(1), \varphi_2(0))$

$= 0001\ 001\ 0001$

2 - Dans la partie haute de l'octant, on définit de même

$$\varphi_m : \{\text{mots sur } 0,1\} \longrightarrow \{\text{mots sur } 0,1\}$$
$$0 \longmapsto 01^m$$
$$1 \longmapsto 01^{m+1}$$

et on montre que

$$\omega(dx, dy) = \varphi_m(\omega(p,q)) \quad \text{où} \quad \begin{cases} m = \left\lfloor \dfrac{dy}{dx-dy} \right\rfloor \\ p = dx - dy \\ q = dy - dy \% (dx-dy) \end{cases}$$

Exemple : $\mathcal{D}(11,3)$ dans la partie basse de l'octant

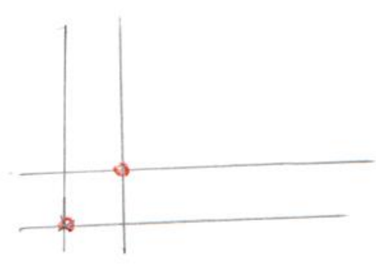
$p = 3$ $\quad\quad m = \left\lfloor \dfrac{11-3}{3} \right\rfloor = 2$

$q = 3 - 2 = 1$

⤳ On doit donc calculer $\omega(\underset{d'x}{\underset{\shortparallel}{3}}, \underset{d'y}{\underset{\shortparallel}{1}})$!

$p' = 1$, $q' = 1$, $m' = \left\lfloor \dfrac{3-1}{1} \right\rfloor = 2$

⤳ On calcule $\omega(1,1) = 1$



$\downarrow \varphi_2$

$$\varphi_2(\omega(1,1)) = \varphi_2(1) = 001$$

$\downarrow \varphi_2$

$$\varphi_2(\omega(3,1)) = \varphi_2(001) = 0001 0001 001$$

Mot de trace pour la droite $\mathcal{D}(11,3)$ avec l'approximation partie entière.

(2) Green / Pitteway   Même idée : on va concaténer des mots au fur et à mesure d'opérations sur dx, dy

Algo dans la partie basse :

```
void droite-gp(int dx, int dy){
    s= "0"
    t= "0"
    dx -= dy
    while (dx!=dy){
        if (dx>dy){
            dx -= dy;
            t = s.t;
            t = s.¬t
        }
        else{
            dy -= dx;
            s = s.t
            s = t.¬s
        }
    }
    Return (s.t)^dx
           (s.¬t)^dx
}
```

(Concaténation de chaînes)

La version de base donne le mot de trace pour l'approximation à la partie entière. La version verte permet d'obtenir la version avec la meilleure approx. L'opérateur ¬ renverse un mot en son miroir : si $w = w_1 \dots w_m$,

$$\neg w = w_m \dots w_1$$

<u>Exemple</u> : • Version partie entière pour $\mathcal{D}(0,1)$

dx	11	8	5	2		1
dy	3				1	
s	0				0001	
t	1	01	001			0001001

Puis on renvoie

$$(s.t)^{dx} = s.t = 00010001001$$

• Version meilleure approximation

dx	11	8	5	2		1
dy	3				1	
s	0				0100	
t	1	01	010			0100010

Puis on renvoie $(s.\neg t)^{dx} = s.\neg t$
$= 01000100010$

On retrouve que les mots de trace sont différents selon l'approx.

Pour $\mathcal{D}(11,3)$:

Partie entière		$w = 00010001001$
Meilleure		$w' = 01000100010$
Partie supérieure		$w'' = 10010001000$

Jean Berstel a démontré que tous ces mots sont conjugués:

- il existe un mot f tel que

$$w' f = f w$$

- il existe un mot f' tel que

$$w'' f' = f' w$$

Propriété: Si $|f| = \ell$, alors

$$dy \times \ell + \left\lfloor \frac{dx}{2} \right\rfloor \equiv 0 \, [dx]$$

Exemple: Pour $\mathcal{D}(11, 3)$:

$$3\ell + 5 \equiv 0 \, [11] \quad \Rightarrow \quad 3\ell \equiv -6 \, [11] \equiv 5 \, [11]$$

$$\Rightarrow \ell \equiv 2 \, [11]$$

Donc $\ell = 2, 13, 24 \dots$ ⇝ on va considérer le mot de taille la plus courte possible

$$w' f = 0100010001001$$

$$\boxed{f = 01}$$

$$f w = 0100010001001$$

Moralité: on peut tout calculer pour w, et transférer quelques lettres pour obtenir w'.

Revenons sur l'algorithme de Green / Pitteway :

```
int mystere (int a, int b){
    a -= b
    while (a != b){
        if (a > b){
            a -= b;
        }
        else{
            b -= a;
        }
    }
    return b;
```

Ex :

a	22	11	7	8
b	11	3	5	6
Res	11	1	1	2

Cette fonction est en fait une version de l'algorithme d'Euclide, qui calcule pgcd (a,b) !

Cependant, l'algo de Green / Pitteway n'est pas aussi rapide qu'Euclide car la concaténation de chaînes est coûteuse.

③ Berstel On a d'abord besoin de notions sur les fractions continues. Tout nombre réel x peut s'écrire sous la forme

$$x = u_0 + \cfrac{1}{u_1 + \cfrac{1}{u_2 + \cfrac{1}{u_3 + \dots}}}$$

où la suite $(u_n)_{n \in \mathbb{N}}$ forme une infinité de coefficients, potentiellement finie pour des rationnels.

Notation : $x = [u_0, u_1, u_2, u_3, \dots]$

Remarques: • Pour un nombre rationnel $\frac{u}{v}$, le premier terme u_0 est la valeur du quotient $\left\lfloor \frac{u}{v} \right\rfloor$

• Pour le deuxième terme, si $u > v$, on a

$$\frac{v}{u} = \frac{1}{\frac{u}{v}} = \frac{1}{\left\lfloor \frac{u}{v} \right\rfloor + \frac{u \% v}{v}}$$

et on recommence ensuite avec $\frac{u\%v}{v}$...

• On peut faire des approximations du réel de départ en ne gardant que certaines valeurs.

$$x \simeq [u_0 \; u_1 \; u_2 \; \dots \; u_k]$$

Algo de Berstel: on va développer le quotient $\frac{dy}{dx}$ en fractions continues; $\frac{dy}{dx} = [u_0 \; u_1 \dots \; u_m]$

On pose:
$$w_0 = 0$$
$$w_1 = 0^{u_1 - 1} 1$$

Pour $i \leq m$,

$$w_i = \begin{cases} w_{i-1}^{u_i} \cdot w_{i-2} & \text{si } i \text{ impair} \\ w_{i-2} \cdot w_{i-1}^{u_i} & \text{si } i \text{ pair.} \end{cases}$$

Exemple: $\mathcal{D}(11, 3)$

$\frac{3}{11} = [0, 3, 1, 2]$

$$w_0 = 0$$
$$w_1 = 0^2 1 = 001$$
$$w_2 = w_0 \cdot w_1^1 = 0001$$
$$w_3 = w_2^2 \cdot w_1 = 00010001001$$

Em travaillant sur les propriétés des mots de trace, on montre

que : 1 si $w = w_0 \dots w_{dx}$,

$$w_0 = 0$$
$$w_{dx} = 1$$
$$w_i = w_{dx-i} \quad \text{pour } i > 1$$

(propriété de symétrie)

pour l'approx à la partie entière dans la partie basse de l'octant

2 $\quad w_i(u,v) = \overline{w_i}(u, u-v)$ sauf pour w_0 et w_{dx}

Négation logique

4 Le boss final

Em 1999, Boyer et (JJ) Bourdim omt présenté um algorithme qui utilise :

- la partie basse de l'octant
- du pas de deux
- du pgcd
- les plages
- les propriétés de symétrie

$\simeq$ 20 x plus rapide que Bresenham sur toutes les droites de taille <= 10 000.