Retour sur l'algorithme de minoration de la distance d'un segment de droite à $\mathbb{Z}^2$

Vincent LEFÈVRE

Loria / INRIA Lorraine

Journées Arinews

25–26 mai 2004

Introduction / plan

Preuve actuelle de mon algorithme de minoration de la distance d'un segment de droite à $\mathbb{Z}^2$: très difficile à lire (transformations de suites de quadruplets, décrivant les configurations successives, au fur et à mesure qu'on ajoute des points).

→ Tentative de présentation d'une preuve plus géométrique et intuitive, ainsi que d'une variante de l'algorithme.

→ Nouveaux timings (comparaisons avec la variante et avec SLZ).

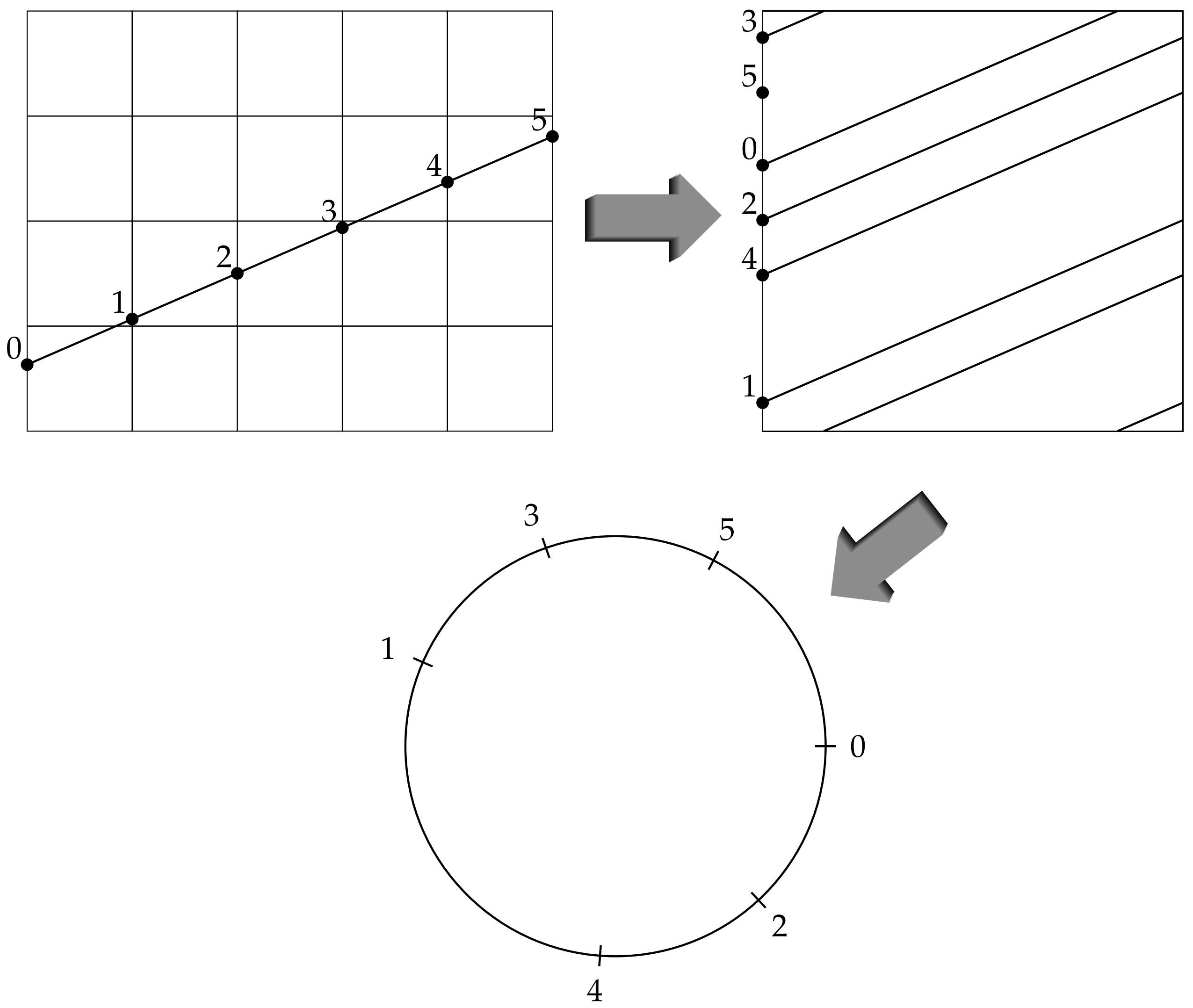
Rappel du problème

But : test exhaustif des fonctions élémentaires en précision fixée (e.g. double précision), i.e. « trouver les nombres machine x tels que $f(x)$ est très proche d'un nombre machine ».

Dans chaque intervalle :

- f approchée par un polynôme de degré $1 \longrightarrow$ segment $y = b - ax$.
- Multiplication des coordonnées par des puissances de 2 $\longrightarrow$ grille $= \mathbb{Z}^2$.

On s'intéresse aux n tels que $\{b - n.a\} < d_0$, où $\{x\}$ désigne la partie fractionnaire positive de x, a, b et d_0 sont des réels et $n \in [\![0, N-1]\!]$.


0
1
2
3
4
5
3
5
0
2
4
1
3
5
1
0
2
4

Si a est rationnel, en considérant les points successifs $0.a$, $1.a$, $2.a$, etc. (modulo 1), on retombera sur le point initial 0.

$\rightarrow$ Cela complique l'analyse théorique.

$\rightarrow$ On suppose donc a irrationnel, ou si on veut, un rationnel $+$ un irrationnel arbitrairement petit.

Pour l'implémentation, a sera rationnel. $\rightarrow$ Faire attention au cas où on retombe sur le point initial (seul cas de discontinuité).

Notations / hypothèse de récurrence

- Intervalles x_0, x_1, ..., x_{u-1} de longueur x, où x_0 est l'intervalle le plus à gauche et $x_r = x_0 + r.a$ (translation de $r.a$ modulo 1).
- Intervalles y_0, y_1, ..., y_{v-1} de longueur y, où y_0 est l'intervalle le plus à droite et $y_r = y_0 + r.a$ (translation de $r.a$ modulo 1).
- Nombre total de points (ou d'intervalles) : $n = u + v$.

Les indices des intervalles sont appelés *rangs*. De même, le point $r.a$ (modulo 1) est appelé point de rang r.

Configuration initiale : $n = 2$, $u = v = 1$.

Passage d'une configuration à la suivante

- Puisque a est irrationnel, $n.a$ se trouve strictement entre 2 points de rangs inférieurs, dont l'un au moins, noté r, est non nul.
- Les points de rangs $r-1$ et $n-1$ (obtenus par translation) sont donc adjacents, et leur distance est soit x, soit y. Il y a la même distance entre les points r et n.
- Le point de rang n coupe donc un intervalle de longueur $h = \max(x, y)$ en deux intervalles de longueurs respectives $\ell = \min(x, y)$ et $h - \ell$.
- La longueur $h - \ell$ est nouvelle, et l'intervalle correspondant n'a donc pas d'antécédent (correspondant à l'ajout de $-a$).
- Cet intervalle a donc 0 comme point frontière.

$\rightarrow$ Le point de rang n est donc complètement déterminé.

Les autres intervalles de longueur h vont être coupés de la même manière, successivement par ordre de rangs croissants (translations de a).

Rangs des intervalles de longueur $h - \ell$: ce sont simplement les rangs des intervalles de longueur h correspondants.

Rangs des intervalles de longueur ℓ : supposons que $\ell = x$ (même raisonnement pour $\ell = y$) ; le premier intervalle de longueur x est obtenu par translation d'un ancien intervalle de longueur x (cf transparent précédent), forcément x_{u-1} (le dernier) car l'image de x_{i-1} est x_i pour tout $i < u$. Cet intervalle est donc x_u et on a bien $x_u = x_0 + u.a$. Intervalles suivants : x_{u+1}, x_{u+2}, etc.

Algorithmes

Algorithme de base (1997) : renvoie un minorant de $\{b - n.a\}$ (en général, $\geqslant d_0$, ce qui permet de conclure immédiatement qu'il n'y a aucun point tel que $\{b - n.a\} < d_0$).

Variante (1998) : renvoie le rang $n < N$ du premier point tel que $\{b - n.a\} < d_0$, ou bien toute valeur $\geqslant N$ s'il n'y a pas de tel point.

Seule la position de b par rapport aux autres points nous intéresse.
$\rightarrow$ Ne garder en mémoire que les informations nécessaires...

Informations en question :

- longueurs x et y, nombres d'intervalles u et v ;
- une valeur binaire indiquant si b se trouve dans un intervalle de longueur x ou dans un intervalle de longueur y ;
- le rang de cet intervalle (utile uniquement dans la variante) ;
- la distance de b à l'extrémité inférieure de l'intervalle.

Note :

- Le rang de l'extrémité inférieure d'un intervalle de longueur x est le rang de cet intervalle.
- Le rang de l'extrémité inférieure d'un intervalle de longueur y est $u +$ le rang de cet intervalle.

Algorithme (version soustractive)

Initialisation : $x = \{a\}$; $y = 1 - \{a\}$; $d = \{b\}$; $u = v = 1$; $r = 0$;

si $(d < d_0)$ **retourner** 0

Boucle infinie :

si $(d < x)$

- **tant que** $(x < y)$
  - **si** $(u + v \geqslant N)$ **fin**
  - $y = y - x;\ u = u + v;$
- **si** $(u + v \geqslant N)$ **fin**
- $x = x - y;$
- **si** $(d \geqslant x)$ $r = r + v;$
- $v = v + u;$

sinon

- $d = d - x;$
- **si** $(d < d_0)$ **retourner** $r + u$
- **tant que** $(y < x)$
  - **si** $(u + v \geqslant N)$ **fin**
  - $x = x - y;\ v = v + u;$
- **si** $(u + v \geqslant N)$ **fin**
- $y = y - x;$
- **si** $(d < x)$ $r = r + u;$
- $u = u + v;$

Timings

Notations pour les transparents suivants :

- Option `-c=k` : les soustractions sont remplacées par une division lorsque l'on doit faire au moins 2^k soustractions (– : algorithme soustractif uniquement).
- Option `-l=3` : redécoupage en $2^3 = 8$ sous-intervalles lorsque l'algorithme échoue (i.e. lorsque le minorant est trop petit).
- Option `-w` : variante de l'algorithme de minoration (remplace l'algorithme d'origine — ce n'est pas le meilleur choix !).

Chaque test est effectué 3 fois (sur marie.medicis.polytechnique.fr, AMD Opteron à 2 GHz), et on prend le temps médian.

Les résultats (pires cas potentiels) ont été comparés entre eux.

Fonctions exp et 2^x, exposant 0

	$\exp x$			2^x		
`-c`	—	`-l=3`	`-w`	—	`-l=3`	`-w`
0	42.79	35.83	39.22	38.48	33.29	49.24
1	26.68	19.34	18.40	24.27	18.94	20.45
2	24.67	17.18	16.88	22.86	17.34	19.07
3	24.52	17.01	16.67	22.63	17.11	18.79
4	24.46	17.07	16.60	23.10	17.46	18.78
5	24.81	17.47	16.76	23.52	18.26	19.04
—	21.95	14.31	15.38	22.09	16.52	18.36

Fonctions sin et cos, exposant 0

	$\sin x$			$\cos x$		
`-c`	—	`-l=3`	`-w`	—	`-l=3`	`-w`
0	40.67	32.09	42.88	39.43	33.89	36.04
1	28.61	19.77	19.58	26.28	20.50	19.61
2	27.01	17.96	18.07	23.75	18.01	17.87
3	26.73	17.75	17.72	23.23	17.10	17.11
4	27.21	18.34	17.50	23.29	17.16	17.03
5	27.56	18.58	17.55	23.26	17.49	17.24
—	24.11	14.96	16.11	20.23	14.19	15.20

Fonctions exp et 2^x, exposant -6

	$\exp x$			2^x		
`-c`	—	`-l=3`	`-w`	—	`-l=3`	`-w`
0	18.69	18.26	59.08	21.75	21.38	81.95
1	12.77	12.59	18.05	13.51	13.23	22.15
2	12.57	12.18	17.18	13.34	12.98	21.14
3	12.20	12.01	17.07	13.34	12.99	21.26
4	12.77	12.30	17.22	13.50	13.45	21.57
5	14.69	14.40	17.65	15.59	14.95	22.34
—	22.39	21.98	26.25	17.95	17.44	21.31

Fonctions sin et cos, exposant −6

	$\sin x$			$\cos x$		
`-c`	—	`-l=3`	`-w`	—	`-l=3`	`-w`
0	16.16	15.68	16.21	15.72	15.51	19.10
1	10.36	10.12	9.79	11.01	10.61	10.74
2	10.03	9.64	9.52	10.80	10.31	10.66
3	9.83	9.47	9.33	10.47	10.15	10.58
4	9.80	9.41	9.25	10.51	10.31	10.29
5	9.68	9.38	9.30	10.91	10.56	10.72
—	316.4	316.0	369.9	162.3	162.1	188.6

Fonction exp, exposant 2

`-c`	—	`-l=3`	`-w`
0	43.86	11.52	11.00
1	40.22	6.41	5.28
2	39.67	5.61	4.73
3	39.63	5.45	4.61
4	39.59	5.50	4.63
5	39.54	5.60	4.71
—	39.00	4.56	4.26

Fonction exp, $x \approx \log(4)$

	intervalle 50616			intervalle 50624		
`-c`	—	`-l=3`	`-w`	—	`-l=3`	`-w`
0	1.79	1.15	1.15	1.69	1.03	1.03
1	1.45	0.80	0.79	1.42	0.75	0.73
2	1.41	0.79	0.76	1.37	0.70	0.74
3	1.44	0.77	0.77	1.36	0.75	0.70
4	1.45	0.77	0.73	1.37	0.72	0.72
5	1.43	0.76	0.72	1.36	0.71	0.68
—	20.71	20.77	24.53	40.68	40.55	48.72

Fonction exp, $x \approx \log(4)$

	intervalle 50632			intervalle 50640		
`-c`	—	`-l=3`	`-w`	—	`-l=3`	`-w`
0	1.14	0.61	1653	1.23	0.89	707.9
1	1.13	0.54	279.2	1.10	0.72	119.9
2	1.13	0.57	259.0	1.04	0.69	111.6
3	1.13	0.55	259.4	1.03	0.67	111.4
4	1.11	0.55	259.0	1.05	0.67	111.5
5	1.12	0.55	259.3	1.02	0.65	111.1
—	231	231	323	103	103	136.9

Comparaison avec SLZ (`wclr22`), 2^{40} points

`test32f` : algo de minoration d'origine (pas la variante), avec divisions, et découpage en 8 sous-intervalles si échec (`-l=3`).

test	interv.	# bits	arrondis	lepuid	ay	marie
`test32f`	$[1/2, \ldots]$	64	D	23.8	81.8	11.5
`test32f`	$[1/2, \ldots]$	65	D & N	23.5	80.8	11.4
`test32f`	$[1, \ldots]$	64	D	26.6	86.8	13.2
`test32f`	$[1, \ldots]$	65	D & N	23.9	77.5	11.7
`wclr22`	$[-1/2, \ldots]$	64	N	26 à 28	111	12.4

D → pour les arrondis dirigés ; N → pour l'arrondi au plus près.

Machines : lepuid : Athlon ; ay : PPC G4 ; marie : Opteron (MEDICIS).