Algorithme de minoration de la distance d'un segment de droite à $\mathbb{Z}^2$: le retour

Vincent Lefèvre

Loria / INRIA Lorraine

Journées au vert

22–23 juin 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).

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

$\longrightarrow$ Nouveaux timings (comparaisons avec la variante et avec SLZ).

Rappel du problème

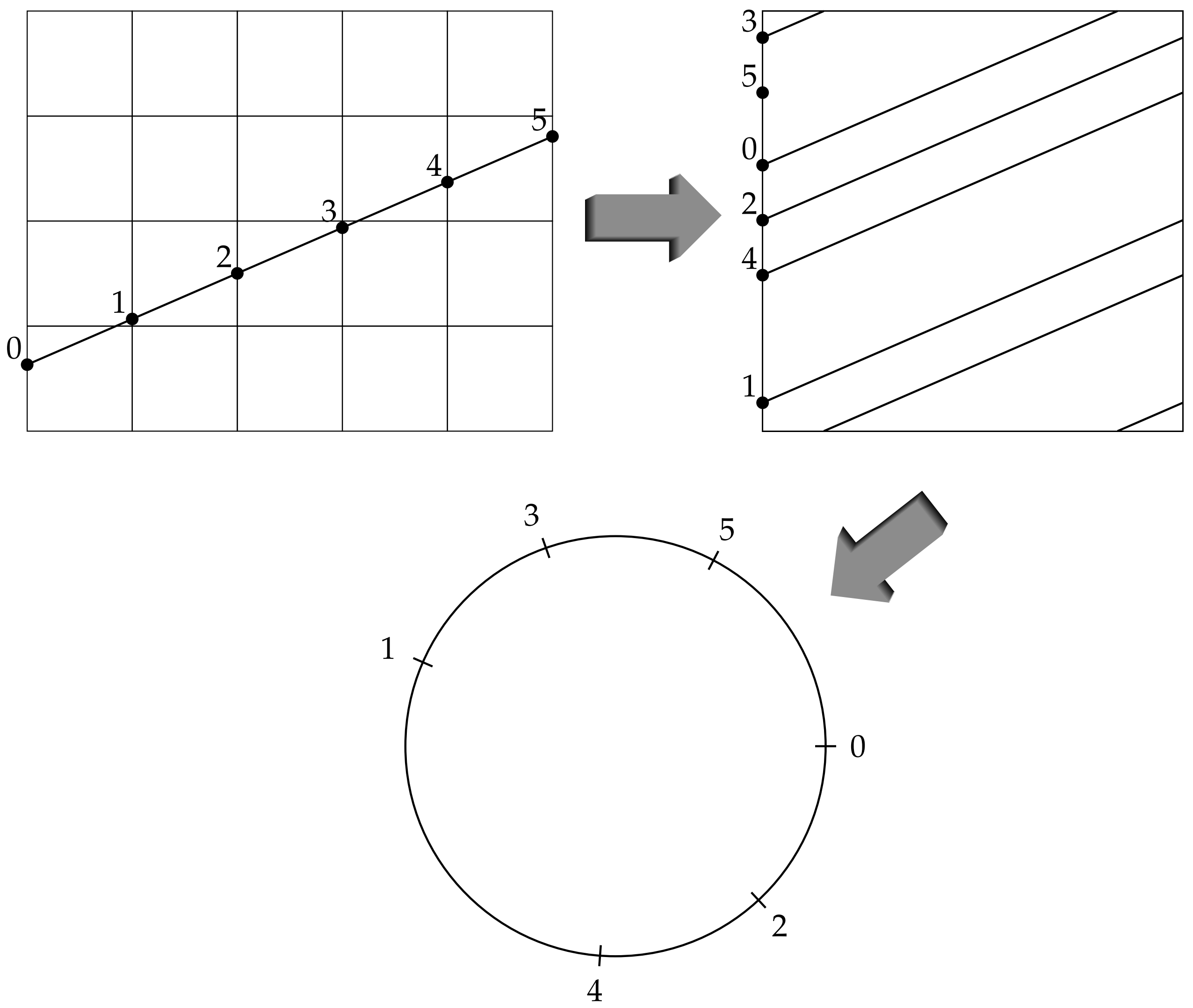
But : test exhaustif des fonctions élémentaires en précision fixée (e.g. double précision), i.e. « trouver les nombres machine x tels que $f(x)$ est très proche d'un nombre machine ».

Dans chaque intervalle :

– f approchée par un polynôme de degré $1 \to$ segment $y = b - ax$.

– Multiplication des coordonnées par des puissances de 2 $\to$ 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.
- Option old `-w` : variante de l'algorithme de minoration (remplace l'algorithme d'origine). N'est plus disponible.

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

Pour chaque test, les résultats (pires cas potentiels) obtenus par application des différentes options ont été comparés entre eux.

→ Détection d'un problème général avec les *pipes* sur le cluster ble.ens-lyon.fr (3 machines quadri-pro Xeon), avec erreurs non détectées au niveau de la génération de code. Exemple...

```
--- tst3.c.1 Mon Jun 21 12:34:25 2004
+++ tst3.c.2 Mon Jun 21 12:34:26 2004
[...]
@@ -129,7 +129,7 @@
                     0x0000000000000000 };

 uint64_t a2_0[] = { 0xEF7FE0B163D4BC53, 0x00000000000007AF };
-uint64_t a2_1[] = { 0x961C1FBE00000000, 0x000000155046A143,
+uint64_t a2_1[] = { 0x0000000000000000, 0x0000000000000000,
                     0x0000000000000000 };
 uint64_t a2_2[] = { 0x3B18024F2F8F3E84, 0x0000000000000000,
                     0x0000000000000000 };
```

Fonctions exp et 2^x, exposant 0

	$\exp x$				2^x			
`-c`	—	`-l=3`	`-w`	old `-w`	—	`-l=3`	`-w`	old `-w`
0	42.30	35.46	35.26	(39.22)	37.83	32.95	32.82	(49.24)
1	26.32	19.27	19.09	(18.40)	23.83	18.72	18.67	(20.45)
2	24.29	17.09	17.04	(16.88)	22.44	17.20	17.28	(19.07)
3	24.09	16.82	16.85	(16.67)	22.21	16.96	17.04	(18.79)
4	24.25	16.83	16.96	(16.60)	22.54	17.28	17.35	(18.78)
5	24.47	17.29	17.29	(16.76)	23.23	18.03	18.08	(19.04)
—	21.54	14.23	14.26	(15.38)	21.68	16.42	16.52	(18.36)

Fonctions sin et cos, exposant 0

	$\sin x$				$\cos x$			
`-c`	—	`-l=3`	`-w`	old `-w`	—	`-l=3`	`-w`	old `-w`
0	40.24	31.72	31.67	(42.88)	39.08	33.52	33.51	(36.04)
1	28.28	19.52	19.49	(19.58)	25.87	20.10	20.18	(19.61)
2	26.62	17.79	17.78	(18.07)	23.59	17.78	17.92	(17.87)
3	26.41	17.54	17.55	(17.72)	22.76	16.93	17.08	(17.11)
4	26.88	18.03	18.07	(17.50)	22.84	17.02	17.16	(17.03)
5	27.15	18.36	18.32	(17.55)	23.15	17.29	17.47	(17.24)
—	23.71	14.74	14.85	(16.11)	19.99	14.12	14.30	(15.20)

Fonctions exp et 2^x, exposant -6

	exp x				2^x			
`-c`	–	`-l=3`	`-w`	old `-w`	–	`-l=3`	`-w`	old `-w`
0	18.29	18.15	18.09	(59.08)	21.42	21.31	21.27	(81.95)
1	12.54	12.52	12.51	(18.05)	13.27	13.18	13.16	(22.15)
2	12.25	12.10	12.00	(17.18)	12.99	12.84	12.79	(21.14)
3	12.10	11.95	11.86	(17.07)	12.84	12.91	12.68	(21.26)
4	12.32	12.23	12.17	(17.22)	13.20	13.37	12.94	(21.57)
5	14.41	14.31	14.16	(17.65)	14.67	14.56	14.54	(22.34)
–	22.13	21.94	21.97	(26.25)	17.62	17.40	17.44	(21.31)

Fonctions sin et cos, exposant −6

	$\sin x$				$\cos x$			
`-c`	—	`-l=3`	`-w`	old `-w`	—	`-l=3`	`-w`	old `-w`
0	15.74	15.56	15.59	(16.21)	15.61	15.43	15.44	(19.10)
1	10.22	10.06	10.10	(9.79)	10.72	10.57	10.58	(10.74)
2	9.67	9.55	9.57	(9.52)	10.43	10.30	10.29	(10.66)
3	9.45	9.25	9.26	(9.33)	10.12	9.99	10.04	(10.58)
4	9.34	9.21	9.22	(9.25)	10.23	10.10	10.11	(10.29)
5	9.34	9.16	9.20	(9.30)	10.50	10.30	10.33	(10.72)
—	314.8	314.3	314.6	(369.9)	161.3	161.1	161.1	(188.6)

Fonction exp, exposant 2

-c	—	-l=3	-w	old -w
0	43.55	11.39	9.63	(11.00)
1	40.00	6.36	5.43	(5.28)
2	39.46	5.57	4.82	(4.73)
3	39.37	5.40	4.73	(4.61)
4	39.40	5.48	4.76	(4.63)
5	39.47	5.61	4.86	(4.71)
—	38.82	4.56	4.11	(4.26)

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

	intervalle 50616				intervalle 50624			
`-c`	—	`-l=3`	`-w`	old `-w`	—	`-l=3`	`-w`	old `-w`
0	1.79	1.12	1.15	(1.15)	1.67	1.06	1.03	(1.03)
1	1.44	0.81	0.78	(0.79)	1.37	0.77	0.77	(0.73)
2	1.40	0.77	0.78	(0.76)	1.36	0.76	0.73	(0.74)
3	1.40	0.77	0.78	(0.77)	1.35	0.72	0.72	(0.70)
4	1.43	0.77	0.77	(0.73)	1.35	0.69	0.72	(0.72)
5	1.39	0.76	0.76	(0.72)	1.35	0.73	0.70	(0.68)
—	20.63	20.70	20.15	(24.53)	40.42	40.54	40.19	(48.72)

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

	intervalle 50632				intervalle 50640			
`-c`	—	`-l=3`	`-w`	old `-w`	—	`-l=3`	`-w`	old `-w`
0	1.15	0.59	7.72	(1653)	1.24	0.87	4.70	(708)
1	1.09	0.56	1.75	(279)	1.05	0.70	1.35	(120)
2	1.11	0.57	1.66	(259)	1.06	0.70	1.29	(112)
3	1.10	0.58	1.69	(259)	1.04	0.66	1.31	(111)
4	1.09	0.53	1.66	(259)	1.01	0.69	1.26	(111)
5	1.04	0.55	1.68	(259)	1.03	0.64	1.26	(111)
—	230	230	230	(323)	102	103	103	(137)

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).