Tests exhaustifs des fonctions élémentaires

(recherche de pires cas pour l'arrondi exact)

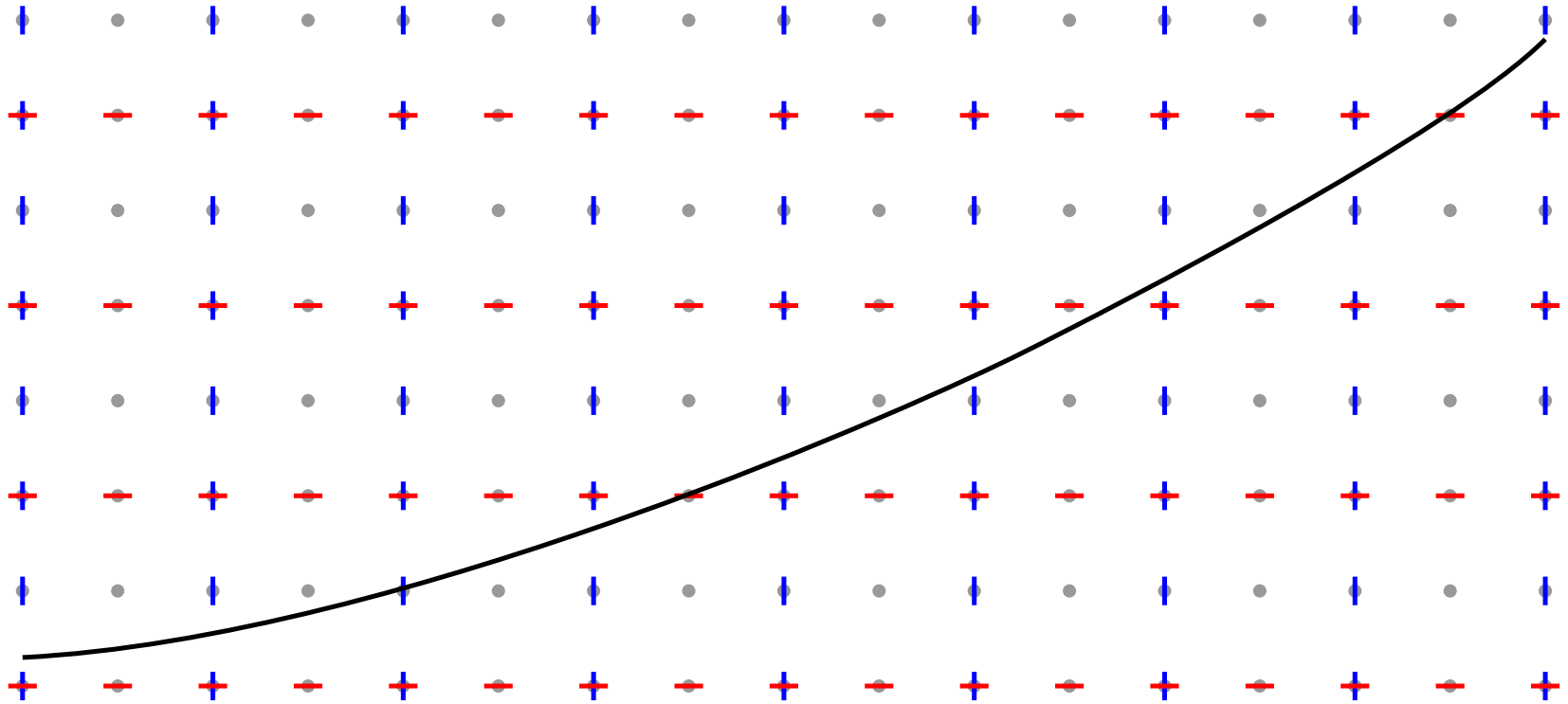
Vincent LEFÈVRE

Loria, INRIA Lorraine

Journées Arinews

Novembre 2002

Rappel du problème de recherche de pires cas



Real Small Value Problem

Pire cas : la courbe est très proche d'un point d'une grille régulière.

- Après multiplication de l'argument x et de l'image $f(x)$ par une puissance de 2 $\rightarrow$ cas où la courbe est très proche d'un point à coordonnées entières.
- La fonction est approchée par un polynôme dans le domaine considéré.

Real Small Value Problem (Real SValP) : étant donnés deux entiers positifs M et T, et un polynôme $P \in \mathbb{R}[X]$, trouver tous les entiers $|t| < T$ tels que :

$$|P(t) \bmod 1| < \frac{1}{M}.$$

Historique

- Mon algorithme : on approche la fonction par des polynômes de degré 1 ($\rightarrow$ segments de droite), puis extension de l'algorithme d'Euclide (plusieurs variantes).
  Avantage : simple à implémenter, opérations de base.
  Inconvénient : degré 1 $\rightarrow$ petits intervalles seulement.
- Extension aux polynômes de degrés supérieurs ?
- Une idée : décorréler les monômes d'un polynôme de degré 2 et se baser sur de la réduction de réseaux (algo LLL).
  $\rightarrow$ Ne fonctionne pas (AMHA).
- Utilisation des travaux de Coppersmith par Damien Stehlé et Paul Zimmermann.

Lien avec les travaux de Coppersmith

Technique de Coppersmith utilisée pour calculer les petites racines d'un polynôme multivarié modulo un entier N. Applications en cryptographie.

- Basée sur LLL, mais technique complètement différente de l'idée de décorréler les monômes.
- Très proche de notre problème (trouver les $|t| < T$ tels que $|P(t) \bmod 1|$ est petit). Transformer notre problème (valeurs réelles) en un problème sur les entiers : multiplication par un certain entier et arrondi + majorations d'erreur.

Integer Small Value Problem

Soit $P \in \mathbb{Z}[X]$ de degré d. Quels sont les petits entiers x tels que $P(x)$ prenne de petites valeurs modulo un entier fixé N ?

De manière équivalente : trouver les petites racines de

$$Q(x, y) = P(x) + y \pmod{N}.$$

Réduction de réseaux et LLL

Réseau : sous-groupe discret de $\mathbb{R}^n$.

$$L = \left\{ \sum_{i=1}^{\ell} n_i \mathbf{b}_i \mid n_i \in \mathbb{Z} \right\}$$

où les $\mathbf{b}_i$ sont linéairement indépendants.

Soit une base $\{\mathbf{b}_1, ..., \mathbf{b}_\ell\}$ d'un réseau $L \subset \mathbb{Z}^n$. L'algorithme LLL fournit en temps polynomial en ℓ et en la taille (en bits) des $\mathbf{b}_i$ une base $\{\mathbf{v}_1, ..., \mathbf{v}_\ell\}$ telle que :

1. $||\mathbf{v}_1|| \leqslant 2^{\frac{\ell}{2}} \det(L)^{\frac{1}{\ell}}$;
2. $||\mathbf{v}_2|| \leqslant 2^{\frac{\ell}{2}} \det(L)^{\frac{1}{\ell-1}}$.

Technique de Coppersmith

Soit α un entier positif. On considère la famille de polynômes

$$Q_{i,j}(x,y) = x^i Q^j(x,y) N^{\alpha-j}$$

où $0 \leqslant i + dj \leqslant d\alpha$.

Si (x_0, y_0) est une racine de Q modulo N, alors c'est une racine de chaque $Q_{i,j}$ modulo N^α.

LLL $\rightarrow$ deux polynômes $v_1(x,y)$ et $v_2(x,y)$ combinaisons linéaires des $Q_{i,j}$ qui prennent de petites valeurs ($< N^\alpha$) pour $|x| \leqslant X$ et $|y| \leqslant Y$. Asymptotiquement ($\alpha \rightarrow \infty$), on obtient $X^d Y = O(N)$.

$\rightarrow (x_0, y_0)$ racine de v_1 et v_2 modulo N^α, donc sur $\mathbb{Z}$, trouvée en regardant les racines entières de $\mathrm{Res}_y(v_1, v_2) \in \mathbb{Z}[x]$.

Algorithme SLZ

Entrée : fonction f (etc.), entiers positifs T, M, d, α.

1. $P(t)$: polynôme de degré d approchant la fonction f ;
   $n = \frac{(\alpha+1)(d\alpha+2)}{2}$.
2. Calculer ε : borne sur l'erreur de l'approximation pour $|t| \leqslant T$.
3. $M' = \lfloor \frac{1/2}{1/M+\varepsilon} \rfloor$, $C = (d+1)M'$ et $P'(x) = \frac{1}{C} \lfloor CP(Tx) \rceil$.
4. $\{e_1, ..., e_n\} \leftarrow \{x^i y^j\}$ pour $0 \leqslant i + dj \leqslant d\alpha$.
5. $\{g_1, ..., g_n\} \leftarrow \{C^\alpha (Tx)^i (P'(x) + \frac{y}{M'})^j\}$ pour $0 \leqslant i + dj \leqslant d\alpha$.
6. Matrice L contenant les coefficients des monômes e_k dans g_ℓ.
7. $V \leftarrow C^{-\alpha} \mathrm{LatticeReduce}(L)$.

8. v_1 et v_2 : les deux plus petits vecteurs de V ;
   p_1 et p_2 : polynômes correspondants.
9. Si $\exists\, x, y \in [-1, 1]$ avec $|p_1(x, y)| \geqslant 1$ ou $|p_2(x, y)| \geqslant 1$, alors retourner `FAIL`.
10. $p(t) \leftarrow \mathrm{Res}_y(p_1(t/T, y), p_2(t/T, y))$.
    Si $p = 0$, alors retourner `FAIL` (ne devrait jamais arriver).
11. Pour chaque racine $t_0 \in [-T, T]$ de p, vérifier si c'est un pire cas.

Borne de Coppersmith : pour éviter d'obtenir `FAIL` à l'étape 9
$\rightarrow T = O(M^{1/d})$.

Erreur due à l'approximation $\rightarrow MT^{d+1} = O(N^d)$,
où $N = 2^n$ et n est la précision cible.

Complexité

Rappel des bornes : $T \ll M^{\frac{1}{d}}$ et $T^{d+1} \ll N^d/M$.

N étant fixée, la plus grande borne pour T s'obtient pour $M \sim N^{\frac{d^2}{2d+1}}$, ce qui donne $T \ll N^{\frac{d}{2d+1}}$.

$d = 1 \rightarrow T \ll N^{1/3}$; complexité $O(N^{2/3+\varepsilon})$, comme mon algo.

$d = 2 \rightarrow T \ll N^{2/5}$; complexité $O(N^{3/5+\varepsilon})$.

Théoriquement, plus d est grand, plus on peut prendre de gros intervalles, mais cela demande d'augmenter aussi M (OK pour trouver des bornes non optimales sur la précision intermédiaire).

Et en pratique ?

Quelques nouveaux pires cas...

N	t_0	$N2^{-1/2+t_0/N} \bmod 1$
2^{64}	586071771766963	$0.1\,1^{47}001111\ldots$
2^{64}	594068190588573	$0.0\,0^{48}100010\ldots$
2^{64}	891586182147388	$0.1\,1^{50}001000\ldots$
2^{64}	901438488920214 7	$0.0\,1^{53}010011\ldots$
2^{64}	960286602385263 1	$0.0\,0^{54}111001\ldots$
2^{113}	11193749220728654 95	$0.0\,1^{63}000000\ldots$
2^{113}	89239603723066500 64	$0.0\,0^{64}101011\ldots$
2^{113}	43616445401128570224	$0.0\,1^{65}011110\ldots$
2^{113}	53608038600996804036	$0.0\,1^{67}000001\ldots$

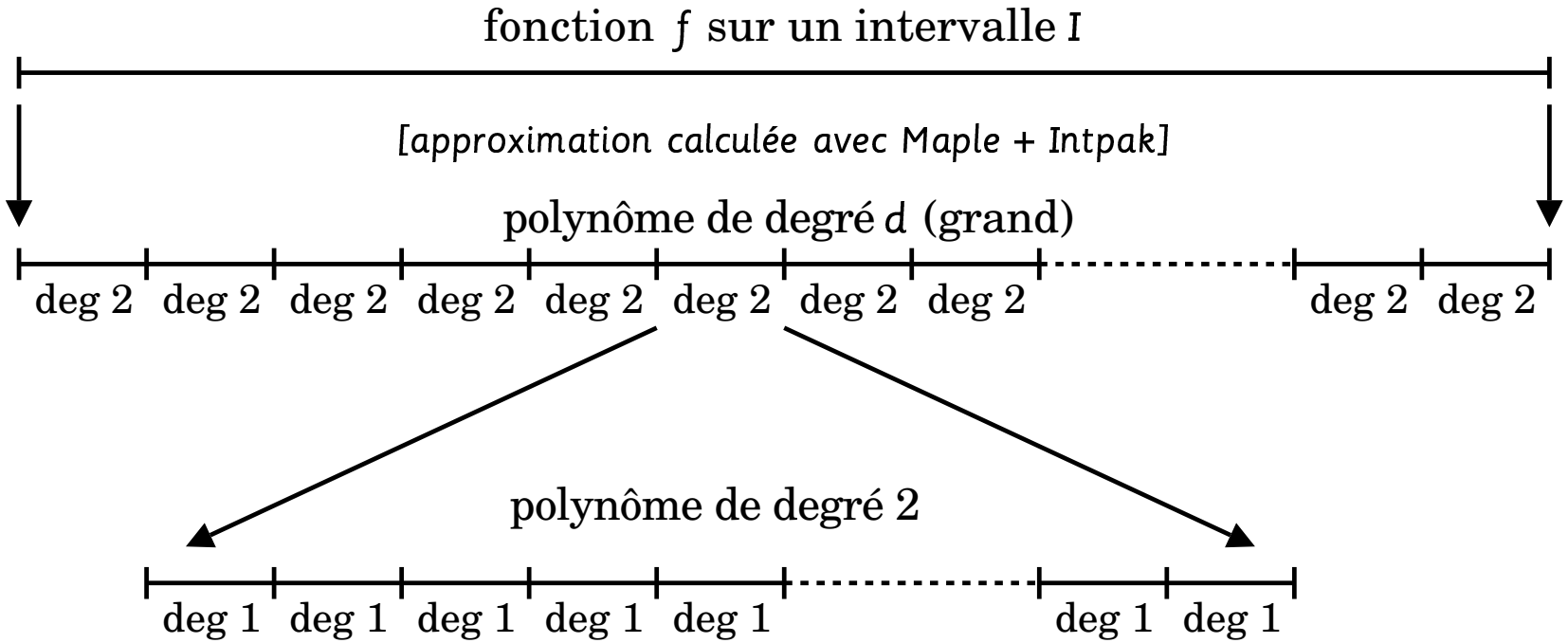
Approximations polynomiales

Problème : obtenir rapidement des approximations d'une fonction f par des polynômes de degré d' (petit) sur de nombreux (petits) intervalles.

Solution : approcher la fonction f par un polynôme de degré d (gros) sur un gros intervalle, puis faire des approximations par des polynômes de plus petits degrés, de manière hiérarchique.

Arithmétique d'intervalles pour garantir les approximations
$\rightarrow$ utiliser des logiciels comme MPFR ou Maple + intpak.

Solution actuelle



Redondance

Mais : bugs dans ces logiciels (MPFR, Maple, intpak).

- Faire des calculs redondants.
- Faire des preuves, en faisant certaines hypothèses. Par exemple : l'un des deux logiciels utilisés peut donner un résultat incorrect, mais on ne sait pas lequel.
- Détecter les bugs éventuels : si possible, obtenir un consensus.

Redondance, détermination du degré du polynôme

Exemple : déterminer le degré. Cela revient à majorer un terme d'erreur : le degré d convient ssi $E(d) < \varepsilon$ (inégalité stricte).

Mais $E(d)$ et ε calculés de manière approchée : $E'(d) \geqslant E(d)$ et $\varepsilon' \leqslant \varepsilon$. $\rightarrow$ Si $E'(d) < \varepsilon'$, alors $E(d) < \varepsilon$ et le degré d convient.

$\rightarrow$ degrés d_1 et d_2. Choisir $\max(d_1, d_2)$ est suffisant. Mais un consensus ($d_1 = d_2$) serait préférable.

On suppose que les termes d'erreur tendent vers 0 quand la précision tend vers l'infini.
$\rightarrow$ Il existe une précision pour laquelle le degré calculé est égal au degré obtenu en supposant les calculs exact (erreur $< \varepsilon - E(d_0)$).

Redondance, détermination des coefficients

Même genre de considérations.

Choisir si possible de renvoyer l'arrondi exact ?

- Pas toujours possible (milieu entre deux arrondis).
- Tenir compte des cas où la précision nécessaire serait trop grande.

→ Renvoyer une (si possible) ou deux valeurs convenables.

Si intersection vide : bug.