Fonction sommatoire de la fonction de Möbius 1. Majorations expérimentales

François Dress

TABLE DES MATIÈRES

On établit la majoration $\lvert M(x)\rvert \le 0.570591\sqrt{x}$, valable pour x appartenant à l'intervalle $\left[33, 10^{12}\right]$, et on décrit les algorithmes qui ont permis de l'obtenir. On étudie également un algorithme de calcul de valeurs isolées de $M(x)$, dont on montre que la complexité est en $O(x^{3/4}\log^{1/2} x)$ pour le temps de calcul, et en $O(x^{1/2})$ pour l'encombrement mémoire.

We establish the upper bound $\lvert M(x)\rvert \le 0.570591\sqrt{x}$, valid for x in the interval $\left[33, 10^{12}\right]$, and we describe the algorithms used to obtain it. We also study an algorithm for the computation of isolated values of $M(x)$, proving that its time and space complexities are $O(x^{3/4}\log^{1/2} x)$ and $O(x^{1/2})$, respectively.

1. INTRODUCTION

On considère la fonction μ de Möbius et on pose $M(x) = \sum_{n\le x}\mu(n)$. (On rappelle que $\mu(n)$ prend la valeur 1 ou -1 dès que n est le produit d'un nombre pair ou impair, respectivement, de nombres premiers différents, et $\mu(n) = 0$ si n a des facteurs carrés.)

Les deux faits majeurs concernant le comportement asymptotique de la fonction sommatoire sont l'observation expérimentale que $M(x)$ est « très petite » (comparable à $\sqrt{x}$ pour les valeurs de x pour lesquelles on peut la calculer numériquement), et le théorème $M(x) = o(x)$, qui est équivalent au théorème des nombres premiers.

Depuis l'introduction de cette fonction par Möbius [1832], beaucoup de conjectures ont été faites sur le comportement du quotient $\lvert M(x)\rvert/\sqrt{x}$. On sait aujourd'hui que $\liminf -M(x)/\sqrt{x} < -1.009$ et que $\limsup M(x)/\sqrt{x} > 1.06$ (voir [Odlyzko et te Riele 1985] – article qui contient en outre une

1058-6458/96 $0.50 per page

bibliographie très complète). Ce résultat a été rendu effectif par Pintz [1987] qui a montré que $\left|M(x)\right|$ dépasse $\sqrt{x}$ pour un x inférieur à $\exp(3.21\,10^{64})$.

La situation pour les «petites» valeurs de x n'est pas très avancée. Il a fallu attendre [Neubauer 1963] pour localiser un dépassement de $\frac{1}{2}\sqrt{x}$ au-delà de $x = 201$ (à savoir $M(x) = 47\,465 \approx 0.5388\sqrt{x}$ pour $x = 7.76\,10^9$). Ultérieurement Cohen et Dress [1979] ont donné la valeur exacte 7 725 038 629 de l'abscisse du premier dépassement de $\frac{1}{2}\sqrt{x}$ après 201. On verra dans cet article que les dépassements en valeur absolue de $\frac{1}{2}\sqrt{x}$ sont expérimentalement assez fréquents et que la seule chose surprenante est que, «par hasard», il n'y en ait pas entre 201 et 7 725 038 629.

Tous les spécialistes conjecturent aujourd'hui que $\left|M(x)\right|/\sqrt{x}$ tend vers l'infini avec x (la conjecture de Riemann implique $M(x) = O(\sqrt{x \log x})$ et ne permet pas de se prononcer sur la propriété $M(x) = O(\sqrt{x})$).

Les majorations expérimentales de la forme

$$\left|M(x)\right| \le c\sqrt{x},$$

qui sont l'objet de ce travail, répondent à trois motivations au moins :

- étendre l'étude expérimentale du comportement de $M(x)/\sqrt{x}$;
- fournir un « raccord » à une majoration explicite du type $\left|M(x)\right| \le \varepsilon x$, démontrée par des méthodes théoriques mais valide seulement pour de très grandes valeurs de x [Dress et El Marraki 1993] ;
- fournir ensuite, grâce à la majoration précédente en εx, un point de départ dans une méthode par itérations qui permet d'améliorer la constante c dans les majorations explicites du type $|M(x)| \le cx \log^{-\alpha} x$ (cf. [El Marraki 1991]).

Les dernières majorations sont intéressantes car, si le théorème $M(x) = o(x)$ est bien entendu susceptible de tous les raffinements valables pour le théorème des nombres premiers – on a notamment $M(x) = O(xe^{-c\sqrt{\log x}})$ – on n'a pas pu rendre ces améliorations effectives pour M comme Rosser et Schoenfeld [1962, 1975] l'avaient fait pour les fonctions π ou Ψ.

2. RÉSULTATS NUMÉRIQUES

Théorème 1. *Soient d'une part les intervalles* $I_n = [a_n, b_n{+}1[$, *pour* $n = 1, 2, \ldots, 8$, *dont les paramètres* a_n *et* b_n *sont définis par le tableau* 1, *et d'autre part l'ensemble* $J = [15, 10^{12}] - \bigcup_{n=1}^{8} I_n$ *de nombres réels. Alors* :

a) *le quotient* $\left|M(x)\right|/\sqrt{x}$ *est supérieur à* 0.5 *pour* $x = a_n$ *et* $x = b_n$, *le maximum sur* I_n *étant atteint pour* $x = c_n$, *avec la valeur indiquée dans le tableau* 1 ;
b) *le quotient* $\left|M(x)\right|/\sqrt{x}$ *est inférieur à* 0.5 *sur l'ensemble* J.

Corollaire. $\left|M(x)\right|/\sqrt{x}$ *est borné par* 0.5705909 *sur tout l'intervalle* $[33, 10^{12}]$.

La démonstration du du théorème est une vérification par ordinateur qui utilise conjointement deux

n	a_n	b_n	c_n	$M(c_n)$	$M(c_n)/\sqrt{c_n}$
1	19	20	19	−3	−0.688247
2	30	33	31	−4	−0.718421
3	114	114	114	−6	−0.561951
4	199	200	199	−8	−0.567105
5	7 725 038 629	7 806 709 032	7 766 842 813	50 286	0.570591
6	108 798 687 923	108 937 938 658	108 924 543 546	170 358	0.516178
7	109 919 609 670	110 087 991 364	110 065 312 957	166 756	0.502639
8	330 486 258 610	330 536 978 172	330 508 686 218	−294 816	−0.512814

TABLEAU 1. Données pour le théorème 1.

algorithmes : l'un pour le calcul rapide d'une valeur isolée de $M(x)$, et l'autre pour le calcul des valeurs de $M(x)$ sur un intervalle de grande longueur. L'algorithme de calcul d'une valeur isolée répond à deux exigences : disposer d'une valeur de départ pour effectuer le calcul sur un grand intervalle, et fournir à l'extrémité de cet intervalle un contrôle indépendant qui valide les calculs effectués.

3. ALGORITHME DE CALCUL D'UNE VALEUR DE M(x)

Les algorithmes antérieurement utilisés exploitaient la formule

$$\sum_{n\le x} M\Big(\frac{x}{n}\Big) = 1. \tag{3.1}$$

Lehman [1960] a utilisé la formule

$$\sum_{n\le x} L\Big(\frac{x}{n}\Big) = [\sqrt{x}]$$

pour calculer la fonction sommatoire de la fonction de Liouville, en découpant la somme de gauche en trois parties. Il propose d'utiliser le même procédé pour la fonction de Möbius, ce qui donnerait pour $M(x)$, tous réarrangements effectués :

$$M\Big(\frac{x}{w}\Big) - \sum_{m\le \frac{x}{w}} \mu(m)\bigg(\sum_{k<v} \mu(k)\Big(\Big[\frac{x}{mk}\Big] - \Big[\frac{x}{mv}\Big]\Big)\bigg) - \sum_{\frac{x}{w}\le l\le \frac{x}{v}} M\Big(\frac{x}{l}\Big)\xi(l),$$

où les paramètres v et w vérifient $v < w < x$, et où la fonction ξ est définie par

$$\xi(l) = \sum_{\substack{m|l\\ m\le \frac{x}{w}}} \mu(m).$$

Lehman propose de prendre v de l'ordre de $x^{1/3}$ et w de l'ordre de $x^{2/3}$.

On obtient ainsi une complexité de calcul apparemment minimale, en $O(x^{2/3})$ opérations, mais il y a une difficulté insurmontable : il faut au préalable établir une table des valeurs de la fonction ξ, puis la stocker. Dans l'hypothèse, qui est celle de Lehman, où x/w reste constant et où le calcul de $M(x)$ est effectué pour de nombreuses valeurs de x, le calcul des valeurs de $\xi(l)$ n'est effectué qu'une seule fois, et le fait que sa complexité soit supérieure à $O(x^{2/3})$ est sans incidence réelle. Par contre le stockage de la table, de taille $O(x^{2/3})$, reste insurmontable ; Lehman travaillait avec x de l'ordre de 10^9 et $x/w = 1\,000$, tandis que nous souhaitons travailler avec x au-delà de 10^{12}.

On pourrait éliminer l'utilisation de la fonction ξ en prenant $v = w$, mais alors la formule de Lehman se réduit à celle obtenue par la méthode de l'hyperbole, que nous exposerons un peu plus loin.

Une autre solution avait été adoptée par Neubauer [1963]. La formule (3.1) permet en effet d'écrire, A étant un entier fixé assez petit,

$$M(x) = -\sum_{\substack{n\le x\\ (n,A)=1}} M\Big(\frac{x}{n}\Big).$$

Si on prend par exemple $A = 210$, on peut ainsi calculer $M(x)$ en réutilisant des valeurs de $M(y)$ avec $y \le x/10$. Divers raffinements permettent ensuite de restreindre considérablement la zone des valeurs de $M(y)$ à connaître, et Neubauer avait ainsi calculé des valeurs de $M(x)$ jusqu'à 10^{10}. Mais les calculs s'alourdissent très vite et les formules spécifiques établies par Neubauer ne paraissent pas susceptibles de généralisation raisonnable.

L'algorithme que nous décrivons dans cet article repose sur l'utilisation de la méthode de l'hyperbole, avec ensuite un calcul récursif de type « quadtree ». Nous allons tout d'abord exprimer la formule de l'hyperbole sous la forme la plus générale.

Proposition. *On considère deux fonctions arithmétiques f et g, et une fonction K à valeurs réelles ou complexes et définie pour $x \ge 1$. On pose*

$$F(x) = \sum_{n\le x} f(n)K\Big(\frac{x}{n}\Big),$$

$$G(x) = \sum_{n\le x} g(n)K\Big(\frac{x}{n}\Big),$$

$$H(x) = \sum_{n \le x} (f * g)(n) K\Big(\frac{x}{n}\Big).$$

Alors on a la relation suivante, valable pour tous $x, A, B \ge 1$:

$$\begin{aligned} H(x) = & \sum_{m \le A} f(m) G\Big(\frac{x}{m}\Big) + \sum_{n \le B} g(n) F\Big(\frac{x}{n}\Big) \\ & - \sum_{\substack{m \le A \\ n \le B}} f(m) g(n) K\Big(\frac{x}{mn}\Big) \\ & + \sum_{\substack{m > A \\ n > B}} f(m) g(n) K\Big(\frac{x}{mn}\Big). \end{aligned} \tag{3.2}$$

Démonstration. On écrit

$$H(x) = \sum_{\substack{m \le x \\ n \le x}} f(m) g(n) K\Big(\frac{x}{mn}\Big)$$

(avec la convention $K(x/mn) = 0$ si $x < mn$), puis on décompose la somme en quatre parties :

$$\sum_{\substack{m \le x \\ n \le x}} = \sum_{\substack{m \le A \\ n \le x}} + \sum_{\substack{m \le x \\ n \le B}} - \sum_{\substack{m \le A \\ n \le B}} + \sum_{\substack{m > A \\ n > B}} .$$

Les deux premiers termes se réexpriment avec les produits de convolution G et F, les troisième et quatrième termes sont exactement ceux de la relation (3.2). □

Le plus fréquemment, on utilise la relation (3.2) avec $AB = x$, de façon à faire disparaître le quatrième terme. On retrouve ainsi de nombreuses formules, et notamment celle utilisée par Schoenfeld [1969] pour fabriquer des majorations itérées de $|M(x)|$, et reprise par El Marraki [1991] :

$$\begin{aligned} -\sum_{n \le x} \mu(n) \log n - 1 = & \sum_{m \le y} (\Lambda(m) - 1) M\Big(\frac{x}{m}\Big) \\ & + \sum_{n \le \frac{x}{y}} \mu(n) \Big(\Psi\Big(\frac{x}{n}\Big) - \Big[\frac{x}{n}\Big]\Big) \\ & - \sum_{\substack{m \le y \\ n \le \frac{x}{y}}} (\Lambda(m) - 1) \mu\Big(\frac{x}{n}\Big), \end{aligned}$$

ce dernier terme pouvant être remplacé par

$$-(\Psi(y) - [y]) M\Big(\frac{x}{y}\Big).$$

Si l'on met dans la proposition $f = g = \mu$, $K \equiv 1$ et $A = B = \sqrt{x}$, on obtient la formule que nous avons utilisée pour calculer $M(x)$, et dont allons maintenant étudier en détail la complexité :

$$M(x) = 2M(\sqrt{x}) - \sum_{\substack{m \le \sqrt{x} \\ n \le \sqrt{x}}} \mu(m) \mu(n) \Big[\frac{x}{mn}\Big]. \tag{3.3}$$

Le calcul « brutal » de la somme dans cette expression est clairement en $O(x)$, mais on peut faire la remarque suivante : soient $m1 < m2 < \sqrt{x}$ et $n1 < n2 < \sqrt{x}$, et supposons que

$$\Big[\frac{x}{m_1 n_1}\Big] = \Big[\frac{x}{m_2 n_2}\Big] = q;$$

alors la somme partielle $\sum \mu(m)\mu(n)[x/mn]$ pour $m_1 \le m \le m_2$ et $n_1 \le n \le n_2$ est égale à

$$(M(m_2) - M(m_1 - 1))(M(n_2) - M(n_1 - 1))q,$$

et se calcule donc en $O(1)$ opérations au lieu de $O((m_2 - m_1)(n_2 - n_1))$. On notera que l'ordre de grandeur de l'encombrement mémoire reste toujours en $O(x^{1/2})$ (tables de μ et de M, au lieu de tables de μ seulement).

On dispose donc d'une possibilité d'accélération du calcul : découper le carré $]0, \sqrt{x}] \times]0, \sqrt{x}]$ en une réunion de rectangles $]m_1, m_2] \times]n_1, n_2]$ sur lesquels $[x/mn]$ reste constant.

Il est très facile d'écrire un algorithme récursif construit sur ce principe en découpant le carré initial, puis chaque carré successif, en quatre carrés jusqu'à obtention de carrés sur lesquels $[x/mn]$ reste constant (en fait, sauf si $\sqrt{x}$ est une puissance de 2, certains carrés sont des rectangles).

Ce principe est connu en informatique sous le nom de « quadtrees » : voir [Klinger 1971 ; Aho et al. 1974 ; Finkel et Bentley 1974]. Cette dénomination se réfère aussi bien au schéma de décomposition récursive « divide and conquer » qu'au type de structure hiérarchique de données – étant entendu que souvent, et c'est le cas ici, les deux points de

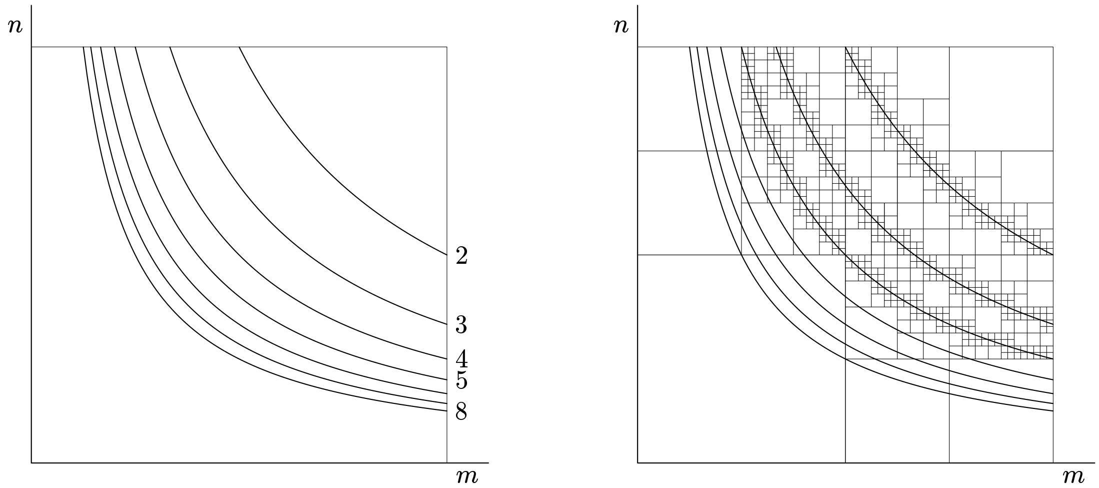

FIGURE 1. Représentation du carré $]0,\sqrt{x}] \times]0,\sqrt{x}]$ avec les courbes de niveau $[x/mn] = 2,\ldots,8$, puis avec un début de partition en quadtrees.

vue sont associés. Le domaine principal d'utilisation des quadtrees est l'imagerie informatique.

La figure 1 illustre le découpage en quadtrees dans le cas de notre problème.

Algorithme 1a: calcul récursif de

$$S = \sum_{\substack{m\le\sqrt{x}\\ n\le\sqrt{x}}} \mu(m)\mu(n)\Big[\frac{x}{mn}\Big].$$

procédure principale :

poser $S, m_1, n_1 \leftarrow 0$ et $m_2, n_2 \leftarrow \sqrt{x}$;
poser $S \leftarrow S + \mathrm{rect}(m_1,n_1,m_2,n_2)$; rendre S et terminer.

fonction $\mathrm{rect}(m_1,n_1,m_2,n_2)$ **:**

poser $q_1 \leftarrow \Big[\dfrac{x}{(m_1+1)(n_1+1)}\Big]$, $q_2 \leftarrow \Big[\dfrac{x}{m_2 n_2}\Big]$.
si $q_1 = q_2$: poser

$$S \leftarrow S + (M(m_2)-M(m_1))(M(n_2)-M(n_1))q_1\,;$$

rendre S et revenir.

sinon : poser $m \leftarrow \Big[\dfrac{m_1+m_2}{2}\Big]$, $n \leftarrow \Big[\dfrac{n_1+n_2}{2}\Big]$;
poser $S \leftarrow S + \mathrm{rect}(m_1,n_1,m,n)$;
poser $S \leftarrow S + \mathrm{rect}(m,n_1,m_2,n)$;
poser $S \leftarrow S + \mathrm{rect}(m_1,n,m,n_2)$;
poser $S \leftarrow S + \mathrm{rect}(m,n,m_2,n_2)$;
rendre S et revenir.

Cet algorithme possède deux défauts.

Il ne tient pas compte de la symétrie et pourrait donc être deux fois plus rapide. Pour conserver la meilleure lisibilité nous ne rectifierons pas ce défaut qui n'affecte en rien les ordres de grandeur.

Le deuxième défaut est par contre fondamental. La récursivité ne fait qu'augmenter la complication pour les petites valeurs de m ou de n et, idéalement, il faudrait limiter la procédure récursive à la région

$$m^2n^2 > (m+n)x,$$

où le gradient de $[x/mn]$ est inférieur à 1. Si on ne rectifie pas cela, les ordres de grandeur sont changés et le théorème 2 ci-dessous n'est plus valide. On peut à la fois corriger ce défaut et garder la plus grande simplicité en proposant de limiter l'application de la procédure récursive au carré $[h,\sqrt{x}] \times [h,\sqrt{x}]$. La remarque sur le gradient de $[x/mn]$ suggère une valeur de h en $O(x^{1/3})$, ordre de grandeur qui est celui des coordonnées $m = n = (2x)^{1/3}$ de l'intersection de la diagonale avec la courbe $m^2n^2 = (m+n)x$. En fait le complémentaire de ce carré introduit alors un terme en $O(x^{5/6})$ dans la complexité, et il faut prendre une valeur de h plus petite. La démonstration du théorème 2 ci-dessous montrera que la valeur optimale de h est en $x^{1/4}\log^{1/2} x$.

Algorithme 1b: calcul de

$$S = \sum_{\substack{m \le \sqrt{x} \\ n \le \sqrt{x}}} \mu(m)\mu(n)\left[\frac{x}{mn}\right]$$

avec limitation de la procédure récursive à la zone efficace.

procédure principale :

initialisation :
poser $S \leftarrow 0$ et $h \leftarrow x^{1/4}\log^{1/2} x$;
poser $m_1, n_1 \leftarrow h$ et $m_2, n_2 \leftarrow \sqrt{x}$;
partie en double boucle :
de $m = 1$ à $\sqrt{x}$
    poser $S \leftarrow S + \mu^2(m)\left[\frac{x}{m^2}\right]$;
    de $n = m+1$ à $\sqrt{x}$
        poser $S \leftarrow S + 2\mu(m)\left[\frac{x}{mn}\right]$;
partie récursive :
poser $S \leftarrow S + \mathrm{rect}(m_1, n_1, m_2, n_2)$; rendre S et terminer.

fonction $\mathrm{rect}(m_1, n_1, m_2, n_2)$ **:**
identique à celle de l'algorithme 1a.

Théorème 2. *La complexité en nombre d'opérations élémentaires, et donc en temps, de l'algorithme* 1b *ci-dessus est en* $O(x^{3/4}\log^{1/2} x)$.

La partie principale de la démonstration de ce théorème consiste à sommer, sur tous les points P de la zone partitionnée en quadtrees, la fonction $f(P) = 1/\operatorname{aire} C_P$, où C_P est le carré auquel appartient le point P. Il est clair que l'on obtient ainsi le nombre total de carrés de la partition.

Cette somme sur tous les points d'une zone que l'on prendra rectangulaire est calculée en effectuant dans un premier temps des sommes le long des parallèles à la première diagonale, sommes dont l'évaluation repose sur un lemme géométrico-combinatoire que nous allons maintenant énoncer.

Définitions. Soit un rectangle du plan traversé par un ensemble E de courbes images de fonctions continues décroissantes. On suppose que ce rectangle peut être pavé par des carrés de côté L, et on note P_0 le pavage possible. On appelle n_0 l'entier défini par

$$\tfrac{1}{2} < 2^{-n_0} L \le 1,$$

et on pose $\delta = 2^{-n_0}$. Pour n entier donné, avec $0 \le n \le n_0$, on note P_n le pavage constitué par les carrés de côté $2^{-n}L$ inscrits dans les carrés de P_0. On appelle carré élémentaire tout carré de P_{n_0}, de côté δ.

On appelle partition en quadtrees du rectangle toute partition formée de carrés de $\bigcup_{n \le n_0} P_n$. Soit E' un recouvrement minimal de E par des carrés élémentaires. On appelle partition optimale respectant l'ensemble E de courbes toute partition en carrés telle que l'intersection d'un carré C quelconque avec E' soit constituée de carrés de la bordure de C (en d'autres termes, E' ne « traverse » aucun quadtree), et telle qu'aucun sous-ensemble de carrés de $\bigcup_{k<n\le n_0} P_n$ ne puisse être remplacé par un unique carré de P_k.

Remarques. Les conditions techniques dans les définitions ci-dessus, comme celles du lemme qui suit, sont adaptées à la démonstration du théorème 2. On peut imaginer de nombreuses variantes (par exemple en effectuant la première sommation le long de parallèles aux axes), mais la complication d'énoncés très généraux semble sans intérêt pratique.

On pourrait également limiter l'utilisation des quadtrees au cas où L est une puissance de 2 ; mais ce n'est pas nécessaire, et la simplification obtenue dans le lemme ci-dessous serait compensée par la nécessité de modifier l'algorithme 1b pour y introduire artificiellement une puissance de 2.

Lemme. *Soit un rectangle du plan traversé par un ensemble E de courbes images de fonctions continues décroissantes, et pavable par des carrés de côté L. Soit une partition en quadtrees optimale respectant l'ensemble E. Considérons une droite $y = x + b$, avec $b \in \mathbf{Z}$, et deux points d'intersection consécutifs A et B de cette droite avec les courbes de E, les abscisses respectives étant x_A et x_B. Soit* *enfin la fonction déjà définie $f(P) = 1/\operatorname{aire} C_P$, où C_P est le carré auquel appartient le point P.*

On a la majoration suivante :

$$\sum_{\substack{n\in\mathbf{Z}\\ x_A<n\le x_B}} f(P(n, n+b)) < 14.$$

Démonstration. Notons C_{AB} le plus petit carré qui contient les carrés élémentaires auxquels appartiennent A et B. On remarque tout d'abord que la condition de décroissance des fonctions implique que les courbes qui passent par A et B ne peuvent rencontrer que des carrés de la bordure de C_{AB}, et que l'intérieur de ce carré ne rencontre aucune autre courbe de E.

On peut maintenant raisonner comme si l'on travaillait avec une partition optimale de C_{AB}. Considérons en effet un point P de AB, et le remplacement de la partition initiale par une partition optimale de C_{AB}. Comme la limitation à C_{AB} est une contrainte supplémentaire, cela ne peut que diminuer (éventuellement) la taille du carré auquel appartient P (cas du carré C dans la figure 2), et donc augmenter la somme des valeurs de $f(P)$.

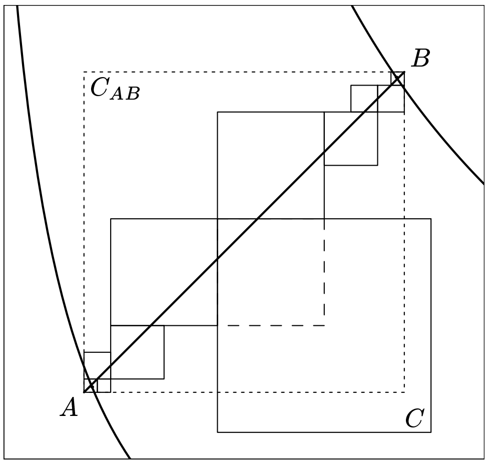

FIGURE 2. Partition en quadtrees optimale à l'intérieur du carré C_{AB}, la diagonale duquel intersecte deux courbes de E en A et B.

Si on considère la taille des carrés successifs de la partition auxquels appartiennent les points de AB, on peut montrer *primo* que le côté augmente puis décroit (au sens large), *secundo* que l'on ne peut avoir plus de quatre carrés consécutifs de même côté.

Entre deux carrés C_1 et C_2 rencontrés par AB on peut toujours, en restant à l'intérieur de C_{AB}, utiliser des carrés de taille au moins égale à la plus petite. Cela démontre le premier point.

Supposons qu'il existe au moins cinq carrés de la partition, consécutifs et de même côté c. On considère alors la partie intérieure à C_{AB} du pavage par des carrés de côté $2c$, et l'on voit que, dans toutes les configurations possibles des cinq carrés consécutifs par rapport à ce pavage, l'un de ces carrés de côté $2c$ pourrait recouvrir un ou plusieurs carrés de la partition, en contradiction avec son caractère optimal.

Nous avons ainsi montré que, pour chaque entier $k \ge 0$, il existe au plus 2×4 carrés de la partition, de côté $2^k\delta$, auxquels appartiennent les points de AB.

L'examen de toutes les configurations possibles montre que le nombre de points entiers de AB qui appartiennent à chacun de ces deux groupes d'au plus quatre carrés est inférieur ou égal à $2 \times 2^k\delta + 1$, de sorte que la contribution de ces points à la somme

$$\sum_{\substack{n\in\mathbf{Z}\\ x_A<n\le x_B}} f(P(n, n+b))$$

est inférieure ou égale à deux fois

$$\frac{(2^{k+1}\delta)+1}{(2^k\delta)^2} = \frac{2}{2^k\delta} + \frac{1}{2^{2k}\delta^2}.$$

En sommant de $k = 0$ à l'infini, on obtient la majoration du lemme :

$$\sum_{\substack{n\in\mathbf{Z}\\ x_A<n\le x_B}} f(P(n, n+b)) \le \frac{4}{\delta} + \frac{4}{3\delta^2} < 8 + \tfrac{16}{3} < 14.$$

□

Démonstration du théorème 2. On utilise le paramètre h (dont la valeur optimale sera $x^{1/4}\log^{1/2} x$), et on appelle N_1 le nombre de points entiers dans le domaine en équerre

$$\{m, n \leq \sqrt{x}, \min(m,n) \leq h\},$$

et N_2 le nombre de carrés dans le découpage en quadtrees du carré $[h, \sqrt{x}] \times [h, \sqrt{x}]$.

Le nombre d'opérations élémentaires effectuées lors de l'exécution de l'algorithme 1b est égal à $k_1 N_1 + k_2 N_2$, où k_1 et k_2 sont deux constantes (faciles à majorer numériquement si on le souhaite).

On a trivialement $N_1 < 2h\sqrt{x}$.

On majore N_2 en sommant $f(P)$ sur les points entiers P du carré $[h, \sqrt{x}] \times [h, \sqrt{x}]$, sommation effectuée en sommant dans un premier temps le long des parallèles à la première diagonale. Si on pose donc $P_1 = (u, h)$, $P_2 = (x^{1/2}, x^{1/2} + h - u)$ et

$$F(u) = \sum_{P_1}^{P_2} f(P),$$

on a $N_2 < 2 \sum_h^{x^{1/2}} F(u)$. D'après le lemme, on peut majorer $F(u)$ par $14(z+1)$, où z est le nombre de courbes $[x/mn] = q$ (entier) rencontrées par le segment P_1P_2 :

$$F(u) < 14\Big(\frac{x}{uh} - \frac{x}{x^{1/2}(x^{1/2}+h-u)+1}\Big).$$

On en déduit

$$\begin{aligned} N_2 &< 28\int_h^{x^{1/2}} F(u)\,du \\ &= 28\int_h^{x^{1/2}} \Big(\frac{x}{uh} - \frac{x^{1/2}}{(x^{1/2}+h-u)+1}\Big)\,du + O(x^{1/2}) \\ &= 28\Big(\frac{x}{h} - x^{1/2}\Big)(\log x^{1/2} - \log h) + O(x^{1/2}). \end{aligned}$$

Si h est de l'ordre de x^{β}, pour $\beta \leq \frac{1}{2}$, le terme principal est $14(1-2\beta)h^{-1}x\log x$, et l'optimisation s'obtient en égalant son ordre de grandeur à celui de N_1, qui est $hx^{1/2}$.

On obtient $h = O(x^{1/4}\log^{1/2} x)$, d'où $N_1 + N_2 = O(x^{3/4}\log^{1/2} x)$, ce qui achève la démonstration du théorème 2. $\square$

On notera pour terminer que le produit de la complexité en temps par la complexité en encombrement mémoire est $O(x^{5/4}\log^{1/2} x)$, donc meilleur que la valeur $O(x^{2/3}) \times O(x^{2/3}) = O(x^{4/3})$ fournie par la méthode de Lehman.

On pourrait enfin comparer avec l'algorithme décrit par [Lagarias et Odlyzko 1987], qui repose sur des algorithmes de calcul « rapide » des valeurs de la fonction zeta, dont la complexité en temps est $O(x^{1/2+o(1)})$, et la complexité en encombrement mémoire $O(x^{1/4+o(1)})$. L'implantation effective de ces algorithmes est très complexe et n'a pas encore été réalisée, de sorte qu'il n'est pas possible de savoir si notre algorithme est en pratique (et pour des valeurs de x voisines de 10^{12}) plus ou moins rapide que l'algorithme de Lagarias et Odlyzko.

4. ALGORITHME DE CALCUL D'UN BLOC DE VALEURS DE M(x)

On considère l'algorithme suivant de calcul d'une valeur isolée de $\mu(n)$. (Cet algorithme exige qu'on puisse parcourir les nombres premiers en ordre, jusqu'à $\sqrt{n}$; on peut par example employer une table de nombres premiers.)

Algorithme 2: calcul de $\mu = \mu(n)$.

initialisation :
poser $a \leftarrow n$, $\mu \leftarrow 1$, $p \leftarrow 2$.
boucle sur les nombres premiers :
tant que $p^2 \leq n$
  si p divise a
    poser $a \leftarrow a/p$ et $\mu \leftarrow -\mu$;
    si p divise a
      poser $\mu \leftarrow 0$, rendre μ et terminer ;
  remplacer p par son successeur et retourner au début de la boucle.
présence d'un diviseur supérieur à $\sqrt{n}$:
si $a > 1$
  poser $\mu \leftarrow -\mu$.
fin :
rendre μ et terminer.

Cet algorithme peut être exécuté, non pas sur un nombre n isolé, mais sur un tableau de k entiers consécutifs dont n est la borne supérieure : pour chaque nombre premier $p < \sqrt{n}$, on cherche le premier entier (s'il existe) divisible par p ; si $p < k$, on obtient tous les autres entiers divisibles par p (avec k/p ou $k/p - 1$ pas de longueur d), sinon l'entier éventuellement trouvé est le seul divisible par p.

Le gain est très important : la complexité de l'algorithme pour un nombre isolé est (au pire) en $O\Big(\frac{\sqrt{n}}{\log n}\Big)$, tandis que la complexité de l'algorithme pour k entiers consécutifs est en

$$O\Big(\sum_{p<k} \frac{k}{p}\Big) + O\Big(\sum_{k \le p \le \sqrt{n}} 1\Big) = O(k \log\log k) + O\Big(\frac{2\sqrt{n}}{\log n} - \frac{k}{\log k}\Big)$$

si $k \le \sqrt{n}$, et en

$$O\Big(\sum_{p \le \sqrt{n}} \frac{k}{p}\Big) = O(k \log\log n)$$

si $k > \sqrt{n}$.

La complexité unitaire (quotient par k des expressions ci-dessus) est une fonction décroissante de k : elle est essentiellement en $O\Big(\frac{\sqrt{n}}{k \log n}\Big)$ pour les petites valeurs de k, elle atteint son minimum en $O(\log\log n)$ pour $k = \sqrt{n}$, et reste ensuite à cet ordre de grandeur.

Ces résultats suggèrent comme le plus simple et le plus efficace de prendre une longueur constante pour les tableaux d'entiers consécutifs qui seront utilisés, égale ou supérieure à la racine carrée de la borne supérieure des entiers sur lesquels on doit travailler.

5. CALCULS SUR ORDINATEUR

Les algorithmes décrits ci-dessus ont été codés en C et exécutés sur des stations Sun Sparc 2.

On contrôle l'inégalité $\big|M(x)\big| \le \frac{1}{2}\sqrt{x}$ sur un intervalle $x_1 \le x \le x_2$ de la façon suivante :

- calcul de $M(x_1)$ par l'algorithme 1b ;
- calcul des valeurs de $M(x)$ entre x_1 et x_2, par l'algorithme 2, par tranches de longueur approximativement égale à $\sqrt{x}$;
- pour chaque tranche, vérification de $\big|M(x)\big| \le \frac{1}{2}\sqrt{x}$ par contrôle en des points espacés convenablement choisis. On exploite l'inégalité triviale

  $$|M(x+y) - M(x)| \le \tfrac{3}{4}y + \tfrac{3}{4},$$

  valable pour tous les entiers $x, y > 0$, de sorte que, lorsque $\big|M(x)\big| = c\sqrt{x}$, avec $c < \frac{1}{2}$, on a $|M(x')| \le \frac{1}{2}\sqrt{x'}$ pour $x' \le x + \frac{1}{3}(2c-4)\sqrt{x} - 1$. Le temps nécessité par ces contrôles est tout à fait négligeable.

Le contrôle de $\big|M(x)\big| \le \frac{1}{2}\sqrt{x}$ détecte la première valeur de x où cette inégalité est fausse, et lorsque cela survient, un programme spécifique simple explore l'intervalle de dépassement, fournissant les valeurs a_n, b_n, c_n et $M(c_n)$ données dans le tableau 1.

Les calculs ont été effectués jusqu'à $x = 10^{12}$. Sur les stations SUN Sparc 2 utilisées, et avec les algorithmes codés en C, il faut environ 4 heures CPU pour explorer et contrôler une tranche d'un milliard (soit donc en moyenne 14.4 μs pour calculer la fonction de Möbius d'un entier).

Le calcul total a donc nécessité 4 000 heures ($\approx$ 165 jours), qui ont été répartis sur trois machines.

Il s'agit d'une durée raisonnable dans l'état actuel d'utilisation des stations et les résultats obtenus sont pleinement efficaces pour leur exploitation dans [Dress et El Marraki 1993]. S'il y avait un enjeu réel pour poursuivre l'exploration du quotient $\big|M(x)\big|/\sqrt{x}$ au-delà de 10^{12}, les méthodes décrites ici, éventuellement utilisées avec des moyens de calcul plus puissants, pourraient permettre d'atteindre une limite que l'on peut prévoir entre 10^{13} et 10^{14}.

BIBLIOGRAPHIE

[Aho et al. 1974] A. V. Aho, J. E. Hopcroft, et J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[Cohen et Dress 1979] H. Cohen et F. Dress, "Calcul numérique de $M(x)$", pp. 11–13, Rapport de l'ATP A12311 « Informatique 1975 » du CNRS, 1979.

[Dress et El Marraki 1993] F. Dress et M. El Marraki, "Fonction sommatoire de la fonction de Möbius, 2 : Majorations asymptotiques élémentaires", *Experimental Math.* **2** (1993), 103–116.

[El Marraki 1991] M. El Marraki, "Majorations effectives de la fonction sommatoire de la fonction de Möbius", Thèse, Univ. Bordeaux, 1991.

[Finkel et Bentley 1974] R. A. Finkel et J. L. Bentley, "Quad trees : A data structure for retrieval on composite keys", *Acta Inf.* **4** (1974), 1–9.

[Klinger 1971] A. Klinger, *Pattern and search statistics*, pp. 303–337 in : *Optimizing Methods in Statistics*, J. S. Rustagi, Ed. Academic Press, Orlando, FL, 1971.

[Lagarias et Odlyzko 1987] J. C. Lagarias et A. M. Odlyzko, "Computing $\pi(x)$: an analytic method", *J. of Algorithms* **4** (1987), 173–191.

[Lehman 1960] R. S. Lehman, "On Liouville's function", *Math. Comp.* **14** (1960), 311–320.

[Möbius 1832] A. F. Möbius, "Über eine besondere Art von Untersuchung des Reihen", *J. reine Angew. Math.* **9** (1832), 105–123.

[Neubauer 1963] G. Neubauer, "Eine empirische Untersuchung zur Mertensschen Funktion", *Numer. Math.* **5** (1963), 1–13.

[Odlyzko et te Riele 1985] A. Odlyzko et H. te Riele, "Disproof of the Mertens conjecture", *J. reine Angew. Math.* **357** (1985), 138–160.

[Pintz 1987] J. Pintz, "An effective disproof of the Mertens conjecture", *Astérisque* **147–148** (1987), 325–333.

[Rosser et Schoenfeld 1962] B. Rosser et L. Schoenfeld, "Approximate formulas for functions of prime numbers", *Illinois J. of Math.* **6** (1962), 64–94.

[Rosser et Schoenfeld 1975] B. Rosser et L. Schoenfeld, "Sharper bounds for the Chebyshev functions $\theta(x)$ and $\Psi(x)$", *Math. of Computation* **29** (1975), 243–269.

[Schoenfeld 1969] L. Schoenfeld, "An improved estimate for the summatory function of the Möbius function", *Acta Arithmetica* **15** (1969), 221–233.

François Dress, Laboratoire d'Arithmétique Algorithmique eXpérimentale (A2X), UMR CNRS n° 9936, Université Bordeaux 1, F-33405 Talence cedex, France (dress@ceremab.u-bordeaux.fr)

Received January 5, 1993 ; accepted May 21