Factorisation par le crible quadratique

Andreas Enge

enge@lix.polytechnique.fr

1 Factorisation et cryptanalyse

La factorisation des entiers est non seulement un problème fondamental en théorie des nombres, mais a trouvé un nouvel intérêt avec l'arrivée de la cryptographie moderne. En effet, la sécurité du cryptosystème à clef publique le plus répandu pour les échanges sur Internet, RSA [6], repose crucialement sur la difficulté de factoriser des nombres produits de deux facteurs premiers.

L'algorithme de choix pour factoriser ces entiers est le *crible algébrique*, dernière émanation d'algorithmes modernes à combinaison de congruences (dont l'idée de base remonte néanmoins à Kraïtchik dans les années 20 [3, chapitre 5, §§ 14-16]). Le crible quadratique est un autre membre de cette famille d'algorithmes, qui est plus rapide pour factoriser des entiers de taille moyenne (jusqu'à quelques centaines de bits) et qui ne demande pas de connaissances particulières en théorie des nombres. Dans la suite, nous nous contenterons d'exposer l'essentiel de cet algorithme, pour plus de détails, voir [4] ou [2].

2 Fermat et la combinaison de congruences

Soit N l'entier impair à factoriser. Une idée de Fermat consiste à l'écrire comme différence de carrés : $N = x^2 - y^2 = (x-y)(x+y)$ donne des facteurs intéressants à moins que l'un des deux ne soit égal à 1. Mais trouver x et y est loin d'être facile. On peut généraliser l'approche et chercher des x et y tels que $x^2 = y^2 \pmod N$. Alors, si p est un facteur premier de N, il divisera $x-y$ ou $x+y$; supposons que ce soit $x-y$. Si q est un autre facteur premier de N, le même argument tient ; si q divise $x+y$ et non $x-y$, ce qui se produit avec probabilité $1/2$, on aura que $\operatorname{pgcd}(N, x-y)$ est un diviseur propre de N (car il sera divisible par p, mais non par q). Par exemple, $1416^2 - 311^2 = 0 \bmod 2041$, et $\operatorname{pgcd}(1416-311, 2041) = 13$ est un facteur non trivial de 2041.

Comment trouver x et y ? On peut tirer des x au hasard entre 0 et N, calculer $x^2 \bmod N$ et espérer tomber sur un autre carré. Mais comme les carrés sont rares, cette approche est vouée à l'échec. L'idée de Kraïtchik est de ne pas se soucier de l'échec, mais de combiner plusieurs de ces congruences $x^2 \bmod N$ pour obtenir un carré du côté droit. Plus précisément, on peut procéder comme suit :

1. Fixons une *borne de friabilité* B et écrivons la *base de friabilité* $\mathscr{F} = \{p_0, \ldots, p_n\}$ contenant les premiers ne dépassant pas B et $p_0 = -1$.
2. Tirons des entiers x_j au hasard et calculons $x_j^2 \bmod N$; si le résultat est B-friable, c'est-à-dire $x^2 \bmod N = \prod_{i=0}^{n} p_i^{a_{ij}}$ a une décomposition en premiers ne faisant intervenir que les éléments de $\mathscr{F}$, nous le gardons sous l'appellation de *relation* ; sinon, nous passons à une autre valeur de x_j. Le processus est répété jusqu'à ce qu'un peu plus de n relations soient trouvées.
3. Tous les membres de gauche des relations sont déjà des carrés. En choisissant un sous-ensemble des relations tel que la somme des exposants $\sum_j a_{ij}$ soit paire pour tout i, le membre de droite du

produit de ces relations devient également un carré, et nous avons trouvé x (comme produit des x_j) et y (comme produit des $\prod_i p_i^{a_{ij}/2}$).

Par exemple, pour factoriser 2041, fixons $B = 10$; nous trouvons

$$\begin{array}{lcrcccccc} 46^2 \bmod 2041 & = & 75 & = & & 3 & \cdot 5^2 & & \\ 47^2 \bmod 2041 & = & 168 & = & 2^3 & \cdot 3 & & \cdot 7 & \\ 49^2 \bmod 2041 & = & 360 & = & 2^3 & \cdot 3^2 & \cdot 5 & & \\ 51^2 \bmod 2041 & = & 560 & = & 2^4 & & \cdot 5 & \cdot 7 & \\ 53^2 \bmod 2041 & = & 768 & = & 2^8 & \cdot 3 & & & \end{array} \tag{1}$$

En multipliant toutes les relations sauf la dernière, nous obtenons

$$(46\cdot 47\cdot 49\cdot 51 \bmod 2041)^2 = 311^2 = 2^{10}\cdot 3^4\cdot 5^4\cdot 7^2 = (2^5\cdot 3^2\cdot 5^2\cdot 7 \bmod 2041)^2 = 1416^2 \pmod{2041}$$

et $\operatorname{pgcd}(2041, 1416-311) = 13$. Nous aurions aussi pu multiplier toutes les relations sauf la première pour trouver $\operatorname{pgcd}(2041, 2000-41) = 1$, ce qui ne nous aurait pas avancés.

Il reste à résoudre deux problèmes :

- Comment trouver *efficacement* des relations ? C'est la tâche du *crible*.
- Comment trouver *systématiquement* et *efficacement* des sous-ensembles de relations à combiner, sachant qu'il en faut éventuellement plusieurs avant de trouver un facteur ? La réponse est donnée par l'*algèbre linéaire*.

3 Création de relations

3.1 Polynôme de Kraïtchik

La stratégie esquissée ci-dessus pour trouver des relations fonctionne, mais est relativement lente ; c'est dû au fait que les $x^2 \bmod N$ sont de l'ordre de N et assez rarement friables. Si on note $L_N(c) = e^{c\sqrt{\log N \log\log N}}$, on peut montrer que la valeur asymptotiquement optimal $B = L_N(\sqrt{2}/2)$ nécessite autour de $L_N(\sqrt{2})$ tests de friabilité. Pour améliorer, on a envie de considérer des nombres plus petits, qui ont plus de chances d'être friables. Kraïtchik propose de prendre des valeurs successives du polynôme $q(x) = (x+b)^2 - N$ pour $b = \lceil \sqrt{N} \rceil$ et x petit (positif ou négatif) ; ces valeurs sont au début de l'ordre de $\sqrt{N}$, et la borne asymptotiquement optimale $B = L_N(1/2)$ mène à environs $L_N(1)$ tests de friabilité, qui peuvent se faire en divisant successivement par les éléments de la base $\mathscr{F}$. Notons qu'il n'est pas utile d'inclure dans la base les premiers $p \geqslant 2$ tels que le symbole de Legendre $\left(\frac{N}{p}\right)$ vaut -1 ou, de façon équivalente, tels que l'équation $x^2 = N \pmod p$ n'a pas de solution.

3.2 Le crible quadratique

L'estimation du temps de calcul ci-dessus ne compte que le nombre de tests de friabilité et néglige le temps nécessaire pour exécuter un tel test, qui peut devenir prohibitif si on procède par divisions successives par les éléments de la base. Le *crible*, semblable au crible d'Ératosthène, permet d'accélérer cette phase en cherchant simultanément toutes les valeurs friables de $q(x)$ pour x dans un intervalle $[x_0, x_1, \ldots, x_{t-1}]$. Il repose sur l'observation que si un p de la base divise $q(x)$, alors il divise tous les $q(x+kp)$.

On commence par remplir un tableau auxiliaire $[y_0, \ldots, y_{t-1}]$ de 1. Puis, pour un p donné, on détermine les deux racines de $q(x)$ modulo p (s'il n'y en a qu'une seule, c'est que soit $p = 2$, soit p divise déjà $N\ldots$). On en déduit les deux indices minimaux i_0 et i_1 tels que p divise $q(x_{i_0})$ et $q(x_{i_1})$. Puis, parcourant le tableau des y_j de p en p, on multiplie tous les y_{i_0+kp} et y_{i_1+kp} par p. Après avoir criblé par tous les premiers de

la base, on repère les i tels que $y_i = q(x_i)$: ces éléments sont friables et donc sûr de fournir une relation. Pour ne pas rater des valeurs friables, il faudrait également cribler par des puissances de premiers.

En pratique, les multiplications des y_i par des p, à exécuter en multiprécision (avec des `BigInteger`), sont trop coûteuses. Pour les éviter, on passe aux logarithmes et se contente d'additionner des $\log_2 p$. On peut même se limiter à des approximation entières aux logarithmes et ne travailler qu'avec des `int`, voire des `short`. Dans l'exemple $N = 2041$ et $b = 46$ avec $B = 10$, regardons l'intervalle donné par $x_0 = 10$ et $t = 20$. Pour $p = 2$, comme b et x_0 sont pairs et N impair, nous avons $i_0 = 1$; après le crible, le tableau y est donné par

$$0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \ .$$

Pour $p = 3$, les deux solutions à $(x+46)^2 - 2041 = 0 \pmod{p}$ sont données par $x = 0$ et $x = 1$, de sorte que $i_0 = 0$ et $i_1 = 2$. En approximant $\log_2 3$ par 2, le tableau y devient

$$2 \quad 1 \quad 2 \quad 3 \quad 0 \quad 3 \quad 2 \quad 1 \quad 2 \quad 3 \quad 0 \quad 3 \quad 2 \quad 1 \quad 2 \quad 3 \quad 0 \quad 3 \quad 2 \quad 1 \ .$$

Les deux solutions à $(x+46)^2 - 2041 = 0 \pmod 5$ sont $x = 0$ et $x = 3$, de sorte que $i_0 = 0$ et $i_1 = 3$; avec $\log_2 5 \approx 2$, nous obtenons

$$4 \quad 1 \quad 2 \quad 5 \quad 0 \quad 5 \quad 2 \quad 1 \quad 4 \quad 3 \quad 2 \quad 3 \quad 2 \quad 3 \quad 2 \quad 5 \quad 0 \quad 3 \quad 4 \quad 1$$

et, après avoir criblé par 7 avec $\log_2 7 \approx 3$,

$$4 \quad 1 \quad 5 \quad 5 \quad 0 \quad 8 \quad 2 \quad 1 \quad 4 \quad 6 \quad 2 \quad 3 \quad 5 \quad 3 \quad 2 \quad 5 \quad 3 \quad 3 \quad 4 \quad 4 \ .$$

À titre d'exemple nous considérons encore 8 comme puissance d'un premier. L'équation $(x+46)^2 - 2041 = 0 \pmod 8$ admettant les deux solutions $x = 3$ et $x = 7$, nous obtenons $i_0 = 1$ et $i_1 = 5$ et ajoutons $2 = \log_2 8 - \log_2 2$ aux endroits correspondants, ce qui tient compte du fait que nous avons déjà criblé par 2 :

$$4 \quad 3 \quad 5 \quad 5 \quad 0 \quad 10 \quad 2 \quad 1 \quad 4 \quad 8 \quad 2 \quad 3 \quad 5 \quad 5 \quad 2 \quad 5 \quad 3 \quad 5 \quad 4 \quad 4. \ .$$

Dans l'intervalle des x, les valeurs de $\log_2 q(x)$ varient entre $10,09\ldots$ et $11,86\ldots$. N'ayant criblé ni par toutes les puissances, ni avec les valeurs précises des $\log_2 p$, nous ne les atteignons nulle part ; néanmoins, les y_i valant 8 ou 10 attirent l'attention et méritent qu'on les regarde de plus près. Effectivement, la valeur de 10 correspond à une relation

$$51^2 = 2^4 \cdot 5 \cdot 7 \pmod{2041},$$

tandis que la valeur de 8 fournit

$$55^2 = 2^3 \cdot 3 \cdot 41 \pmod{2041},$$

qui n'est pas une relation car 41 n'est pas contenu dans la base.

On n'est pas forcé de cribler par tous les premiers ou puissances de premiers de la base ; notamment, les très petits premiers contribuent de façon négligeable, mais sont assez coûteux à traiter. Le choix de la bonne taille t de l'intervalle à cribler est un compromis entre la mémoire disponible dans la machine (voire la mémoire cache du processeur) et le temps nécessaire pour initialiser le crible par les bonnes valeurs des i_0 et i_1.

3.3 Polynômes multiples

Les valeurs de $(x+b)^2 - N$ croissent à peu près comme $2x\sqrt{N}$. Quand x grandit, il peut être préférable de changer de polynôme. Pour cela, il a été suggéré de prendre $q(x) = (ax+b)^2 - N$ avec un b tel que a divise $b^2 - N$, disons $c = \frac{b^2-N}{a}$. On a alors $q(x) = a(ax^2 + 2bx + c)$, et $q(x)/a$ croît encore avec $\sqrt{N}$ pour un bon choix des paramètres.

3.4 Grands premiers

Pour faciliter le crible, on aurait envie d'agrandir la base de friabilité $\mathscr{F}$. Malheureusement, cela tend à alourdir la phase de l'algèbre linéaire. Une approche intermédiaire consiste à traiter séparément les «grands premiers» ; par cela, on entend un premier dépassant la borne de friabilité B, mais plus petit qu'une deuxième borne B_2 (souvent choisie telle que $B_2 < B^2$). On peut alors garder toutes les relations, appelées *partielles*, qui, à part éventuellement un grand premier, ne contiennent que des premiers plus petits que B. Notons que pour reconnaître une telle relation, il suffit de cribler sur tous les premiers (et leurs puissances...) jusqu'à la borne B ; si le résidu est borné par $B_2 < B^2$, il doit être premier. Il existe aussi des variantes avec deux grands premiers ou plus, mais elles demandent un «vrai» algorithme de factorisation pour les résidus.

4 Algèbre linéaire

4.1 Élimination de Gauß

La deuxième phase de l'algorithme consiste à trouver un produit de relations dont le membre de droite est un carré. Pour cela, il faut que tous les exposants de la factorisation en premiers du produit des relations soient pairs. En rangeant ces exposants dans une matrice $A = a_{ij}$, avec une relation par colonne, on est donc amené à trouver une combinaison linéaire des colonnes dont toutes les entrées soient paires. Comme uniquement la parité des entrées compte, on voit qu'il suffit de faire les calculs modulo 2, ou autrement dit sur le corps $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$, et qu'il faut trouver le noyau de A. La matrice ayant plus de colonnes que de lignes, ce noyau est non trivial. Il peut se déterminer comme d'habitude par élimination Gaussienne en mettant la matrice sous forme triangulaire.

En reprenant l'exemple (1), la matrice sur $\mathbb{F}_2$ devient

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix},$$

la première ligne correspondant au premier 2, la deuxième au premier 3 et ainsi de suite (le «premier» -1 n'apparaît pas dans l'exemple, mais est à prendre en compte en général). On commence par choisir un pivot dans le première colonne ; c'est forcément le 1 à la deuxième ligne, qu'on échange avec la première. Par addition modulo 2 de cette ligne aux autres, on élimine toutes les autres entrées 1 de la colonne. Ici, il n'y a rien à faire. La matrice devient

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

À la deuxième colonne, on peut choisir le 1 de la deuxième ligne comme pivot, et il suffit d'additionner la deuxième à la quatrième ligne pour obtenir la matrice suivante :

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Pour la troisième colonne, on choisit le 1 de la troisième ligne comme pivot, et additionne la troisième à la quatrième ligne, ce qui donne

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

À la quatrième colonne (et à la cinquième, qui ne correspond à plus aucune ligne), on ne trouve pas de pivot, et la matrice est dans sa forme finale. Pour résoudre $Ax = 0$, on peut donc choisir x_4 et x_5 librement et obtient comme base du noyau $(1,1,1,1,0)^T$ et $(1,0,0,0,1)^T$. La première solution, par exemple, nous dit qu'en multipliant toutes les relations sauf la dernière, nous obtenons un carré ; c'est la solution devinée à (1) qui permet effectivement de factoriser ce nombre. En général, il faut parcourir toute la base du noyau.

Plusieurs éléments de la base peuvent mener à des factorisations différentes, qu'il convient de raffiner autant que possible par des pgcd croisés. Par exemple, si $N = p_1p_2p_3p_4$ est le produit de quatre premiers, on peut trouver d'une part $N = (p_1p_2)\cdot(p_3p_4)$ et d'autre part $N = (p_1p_3)\cdot(p_2p_4)$; en faisant le pgcd de p_1p_2 et p_1p_3, par exemple, on arrive à complètement factoriser le nombre.

Notons que l'arithmétique dans $\mathbb{F}_2$ correspond à des opérations logiques : la multiplication à «et», l'addition à «ou exclusif». Un vecteur d'éléments de $\mathbb{F}_2$ peut se représenter par un tableau d'entiers, dont chaque bit représente une entrée du vecteur, et qui sera manipulé par des masques et les opérations logiques `&`, `|`, `^` et `!` ; ou (probablement de façon moins efficace) par la classe `BitSet`.

4.2 Filtrage

Il se peut que la matrice devienne trop grande pour être manipulée de façon efficace dans la mémoire centrale de l'ordinateur, notamment quand on inclut la variation des grands premiers. Notons pour y remédier que la matrice est *creuse* à la base : chaque colonne ne contient que quelques entrées 1. Avant de travailler par élimination sur la forme dense de la matrice, on peut simplifier son écriture creuse, donnée par la liste des coefficients non nuls colonne par colonne ou ligne par ligne, pour réduire le nombre de lignes et de colonnes.

Notamment, il convient d'éliminer les *singletons* : si un premier n'apparaît que dans une seule relation, il est impossible d'apparier cette relation avec une autre pour éliminer le premier. Autant donc jeter la relation et ce premier, ce qui enlève une ligne et une colonne de la matrice.

Une autre possibilité est de traiter séparément les *grands premiers*, qui apparaissent dans peu de relations. La borne choisie pour définir ce qui est «grand» peut différer de celle prise pendant le crible ; on peut même se contenter de n'introduire la notion de grand premier que pendant la phase de l'algèbre linéaire. Dans (1), par exemple, le premier 7 n'apparaît que dans deux relations, ce qui correspond à une ligne de la matrice avec uniquement deux entrées 1. On peut donc remplacer les colonnes correspondantes par leur somme et effacer la ligne ne contenant plus que des 0. De manière plus générale, k relations faisant intervenir le même grand premier peuvent être combinées pour donner $k-1$ colonnes. Cela peut faire apparaître de nouveaux singletons, et ainsi de suite. Une implantation efficace utilisera des structures de données avancées (table de hachage etc.), et fera attention à ne pas perdre le lien avec les relations originales — après tout, il faudra pouvoir remonter à une équation $x^2 = y^2 \pmod N$ pour pouvoir factoriser ! Pour plus de détails, voir [1].

4.3 Catastrophes et suites linéairement récurrentes

Pour aller plus loin, on peut commencer par oublier les lignes les plus denses (correspondant aux petits premiers), et ne travailler plus qu'avec les lignes très creuses, et ce de façon à minimiser le remplissage. Une fois la partie creuse réduite à néant, on peut appliquer la même transformation à la partie dense

et la résoudre par élimination de Gauß habituelle. Une version dynamique de cette approche est décrite dans [5].

Les algorithmes les plus modernes, sans doute en dehors de ce qui est faisable pendant ce projet, utilisent à fond le fait que la matrice est creuse : ils se réduisent essentiellement au calcul du produit de la matrice et d'un vecteur, qui peut se calculer sans jamais écrire la matrice au format dense. C'est le cas, par exemple, de [7].

5 Travail demandé

Le but du projet est d'arriver à une implantation parallèle du crible quadratique en CUDA. Comme d'habitude dans ce genre de calculs, toutes les astuces sont permises pour aller plus vite et donc factoriser des nombres plus grands. Voici quelques défis :

- $F_7 = 2^{2^7} + 1 = 340282366920938463463374607431768211457$, le septième nombre de Fermat ; factorisation publiée par Morris et Brillhart en 1975 et obtenue par la méthode des fractions continues, également à base de combinaison de congruences ;
- $2^{251} - 1 =$ 3618502788666131106986593281521497120414687020801267626233049500247285301247, un nombre de Mersenne qui n'est pas premier ;
- 2135987035920910082395022704999628797051095341826417406442524165008583957746445088405009430865999, le module RSA des premières cartes bleues, sans doute trop difficile pour le crible quadratique ; mais sait-on jamais ?
- des nombres tirés du projet Cunningham, `http://homes.cerias.purdue.edu/~ssw/cun/` ;
- les nombres que vous choisirez ; après tout, il est bien connu que des produits s'obtiennent plus facilement que des factorisations...

Références

[1] Stefania Cavallar. Strategies in filtering in the number field sieve. In Wieb Bosma, editor, *Algorithmic Number Theory — ANTS-IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 209–230, Berlin, 2000. Springer-Verlag.

[2] Richard Crandall and C. Pomerance. *Prime Numbers — A Computational Perspective*. Springer-Verlag, New York, 2000.

[3] M. Kraïtchik. *Théorie des nombres*. Gauthier-Villars, Paris, 1922.

[4] Carl Pomerance. A tale of two sieves. *Notices of the AMS*, 43(12) :1473–1485, December 1996.

[5] Carl Pomerance and J. W. Smith. Reduction of huge, sparse matrices over finite fields via created catastrophes. *Experimental Mathematics*, 1(2) :89–94, 1992.

[6] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public–key cryptosystems. *Communications of the ACM*, 21(2) :120–126, February 1978.

[7] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1) :54–62, January 1986.