Crible quadratique, fractions continuées et consorts

où l'on verra Ératosthène, Fermat, Legendre, Gauss,
Kraïtchik, Lehmer, Pollard, Lenstra et Pomerance
se disputant à l'envi l'honneur du meilleur

algorithme de factorisation

par Cyril Banderier,
maîtrise de mathématiques,
Université de Rouen.
21/8/1998.

Directeur du projet :

G. Duchamp

Table des matières

1 Factorisation et second principe de la thermodynamique

Le deuxième principe de la thermodynamique implique que l'entropie d'un système isolé va en augmentant, interdisant ainsi toute réversibilité : ceci implique qu'il est impossible de "revenir en arrière". Les mathématiciens d'antan, ignorant ce postulat, n'eurent de cesse de l'enfreindre. Si, à tout hasard, on demande à un babylonien, en -5000 avant J.C., de prendre deux nombre s et de les multiplier, il consultera ses tables, préalablement établies par additions successives (on supposera que nous sommes tombés sur un autochtone qui dispose et sait utiliser de telles tables). Même chose si on lui demandait de faire une division. Ce n'est que trois mille ans plus tard que nous pouvons tomber sur des comptables connaissant un mécanisme, un algorithme, pour effectuer plus rapidement ces opérations, "inverses" l'une de l'autre.

Mais, si l'on y prend garde, la division n'est pas vraiment l'opération "réciproque" de notre multiplication. La réversibilité se poserait plutôt sous cette forme : pouvez-vous me dire quels sont les deux nombres que j'ai multiplié entre eux pour obtenir le nombre 8051 ? Formulé plus mathématiquement : pouvez-vous me donner la factorisation de 8051 ? En cryptologie (théorie des codes secrets), une telle opération, facile dans un sens et épineuse dans l'autre, est appelée une fonction trappe. On a en général besoin soit de beaucoup de temps, soit d'aide, i.e. d'informations supplémentaires, pour pouvoir revenir en arrière : cette aide est souvent appelée la clef et la fonction trappe est alors dénommée "fonction à brèche secrète" (en anglais : trapdoor function).

Nous venons ainsi de donner un premier champ d'application démontrant l'intérêt de répondre à notre question : comment factoriser un nombre ?

2 La méthode d'Ératosthène

Le premier à avoir donné une méthode permettant de répondre à cette question est Ératosthène, vers -250. Il propose de faire les divisions successives jusqu'à $\sqrt{n}$, méthode au demeurant un peu grossière, voire barbare, adjectif qui n'est pas des plus adéquats pour un hellène... enfin passons ! En effet, d'après Tchebycheff, si $\pi(x)$ désigne le nombre de nombres premiers inférieurs à x, on a, pour $x \geq 11$:

$$\pi(x) > \frac{x}{\ln(x)} \ln \frac{2^{1/2}3^{1/3}5^{1/5}}{30^{1/30}}.$$

Donc la méthode d'Ératosthène, pour factoriser un nombre de 100 chiffres qui serait le produit de deux nombres premiers de 50 chiffres, nécessiterait plus de $10^{50}/\ln(10^{50}) \times 0.92$ divisions, à raison de mille divisions par nanoseconde sur un super-ordinateur hyperparallèle, quantique et extra-terrestre, il faudrait donc la bagatelle de 2×10^{28} années, ce qui est plus que l'âge de l'univers (même en prenant une bonne marge suite aux querelles de cosmologistes). Et que des physiciens révisionniste ne viennent pas nous dire que notre espace euclidien naïf et notre conception limitée du temps ne sont que leurres et surenchérir en déclarant que la physique quantique n'interdit pas les voyages temporels, quoiqu'en dise notre premier chapitre... Nous aurions du mal à comprendre les implications effectives de telles vociférations, toutefois, nous ne négligerons pas les potentialités d'un ordinateur quantique à la Deutsch-Lockwood : qui sait ce que deviendrait la théorie de la complexité avec des électrons qui font des micro-sauts dans le passé ?

3 La méthode de Fermat

En 1643, Fermat, en réponse à une colle de son ami le Père Mersenne, propose une méthode basée sur l'identité remarquable $x^2-y^2=(x+y)(x-y)$, ainsi on cherche désormais à factoriser n en l'écrivant comme différence de carrés, ce qui reste néanmoins toujours fastidieux.

Tout nombre composé impair peut s'écrire comme différence de deux carrés, en effet : $ab=(\frac{a+b}{2})^2-(\frac{a-b}{2})^2$. Cette méthode sera particulièrement recommandée si a et b sont proches l'un de l'autre.

Exemple 3.1 *Si l'on tente de factoriser 8051, le plus rapide est sans doute de remarquer, comme le signale avec remords C. Pomerance dans son article, que* $8051 = 8100-49 = 90^2-7^2$ *d'où* $8051 = 97*83$.

Toutefois cet algorithme n'est guère plus intéressant que celui des divisions successives, c'est dû au fait que la plupart des entiers n'ont pas des facteurs premiers proches.

4 La méthode de Gauss-Kraïtchik

C'est ce qui poussa, en 1801, Gauss à s'intéresser à une amélioration de l'idée de Fermat, il suffit en effet d'avoir x^2-y^2 égal à un multiple de n, pour en tirer des informations : $\text{pgcd}(x\pm y, n)$, si $x \not\equiv \pm y \bmod n$, sera un diviseur de n. Avec un peu de chance, il sera différent de n et de 1. En fait, la relation s'écrit plus mathématiquement $x^2 \equiv y^2 \bmod n$ et on dit alors que x^2 et y^2 sont des carrés (ou encore des "résidus quadratiques") modulo n. Gauss se préoccupe alors de trouver des petits résidus quadratiques pour que les calculs soient simples. C'est cette idée qui sera remise au goût du jour par M. Kraïtchik en 1911.

5 Factorisation par les fractions continuées

Cette factorisation par les résidus quadratiques avait d'ailleurs été une idée que Legendre, dans sa *Théorie des nombres* de 1798, avait donnée en remarque en signalant même que ces résidus pouvait être obtenus par le développement en fraction continuée de $\sqrt{n}$.

D.H. Lehmer et R.E. Powers ont clairement exposé cette méthode dans leur article *On factoring large numbers*, Bulletin of the American Mathematical Society, pp.770-776, 1931, en ajoutant l'idée suivante : parmi la liste des relations $A_{i-1}^2 \equiv Q_i \bmod n$ obtenues à partir du développement en fraction continuée de $\sqrt{n}$ (comme ce sera détaillé ci-dessous), il suffit de trouver Q_i et Q_j tel que leur produit soit un carré (donc $\exists x, y \in \mathbb{N} | x^2Q_i = y^2Q_j$) et on a alors $(xA_{i-i})^2-(yA_{j-1})^2 \equiv 0 \bmod n$, et avec un peu de chance le pgcd de n et de $xA_{i-i} \pm yA_{j-1}$ sera un diviseur non trivial de n.

Pour factoriser n, écrivons d'abord $\sqrt{n}$ sous forme de fraction continuée :

$$\sqrt{n} = b_0 + \frac{1}{b_1+\frac{1}{b_2+\frac{1}{\cdots}}}.$$

Rappelons que ce calcul se fait très simplement avec $x_0=\sqrt{n}$, $b_i=[x_i]$ et $x_{i+1}=\frac{1}{x_i-b_i}$.

On peut montrer par récurrence que x_i peut s'écrire $\frac{\sqrt{n}+P_i}{Q_i}$ avec P_i et Q_i entiers. De plus, les réduites A_n/B_n peuvent s'obtenir par $A_n=b_nA_{n-1}+A_{n-2}$(où $A_{-2}=0$ et $A_{-1}=1$).

Les réduites d'un développement en fraction continuée d'un réel $x > 1$ vérifient $|A_i^2 - x^2 B_i^2| < 2x$ d'où $|A_i^2 \bmod n| < 2\sqrt{n}$ et ainsi nous avons $|Q_i| < 2\sqrt{n}$ [Hardy,Koblitz], ce qui garantit une petite taille des résidus quadratiques ainsi obtenus.

En outre, on a de nombreux résultats sur le développement en fraction continuée de $\sqrt{n}$: on sait que la période est majorée par $\frac{1}{2}[\sqrt{n}]^2$. Dans les faits, on a bien mieux : une majoration par $(\ln n)^2 \ln \ln n$ pour "presque tous" les entiers. On sait aussi que le développement est du type $b_0, [b_1 \dots b_k(b_k)b_{k-1} \dots b_1, 2b_0]^\infty$.

6 Chassez l'algèbre linéaire, elle revient au galop

Nos mathématiciens modernes, plus soucieux du respect du second principe de la thermodynamique, comprirent que la réversibilité eût son coût (les algorithmiciens emploient volontiers le terme "complexité") et surent détecter l'importance du mot "système" dans l'énoncé du second principe.

Effectivement, en décomposant en facteurs premiers les différents Q_i sus-mentionnés, on obtient, pour chaque Q_i, une liste d'entiers e_k avec $Q_i = (-1)^{e_0} p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$, que l'on peut stocker sous forme de vecteurs ; en fait on peut même se limiter à stocker ces vecteurs modulo 2, car, pour savoir si un produit sera un carré, seule la parité compte. Comme il est difficile de stocker des vecteurs de longueur infinie (la base des nombres premiers est bien infinie), on se limite à des factorisations partielles sur une sous-base finie préalablement choisie. Pour de nombreux Q_i, de tels factorisations peuvent néanmoins s'avérer complètes, on dit que de tels nombres qui n'ont que de petits facteurs premiers sont *friables* (en anglais : smooth). Si le nombre de factorisations complètes est plus grand que le cardinal de la sous-base de nombres premiers choisie, l'algèbre linéaire nous apprend que ces vecteurs sont liés : on peut donc trouver un produit égal à un carré.

Cette légère amélioration de l'algorithme de Lehmer et Powers, due en 1970 à M.A. Morrison & J.Brillhart (confer *A method of factoring and the factorization of* F_7, Mathematics of computation, pp183-205, 1975) leur permit de réussir la première factorisation du septième nombre de Fermat F_7. Il faut bien avoir conscience, qu'avant 1970, un nombre de vingt chiffres représentait les limites de la factorisation mais qu'avec l'arrivée de l'algorithme de factorisation par les fractions continuées de Brillhart et Morrison, il était désormais facile de factoriser un nombre de 50 chiffres (F_7 a 39 chiffres, de nos jours, sur un Pentium 100, il faut 22 minutes pour le factoriser avec l'algorithme de l'annexe 1). En fait, on peut même regarder le développement de $\sqrt{kn}$ pour un k préalablement choisi.

Pomerance a estimé la complexité de cet algorithme à $\exp(\sqrt{2(\ln n)\ln\ln n})$ et, en 1983, il a proposé une variante, avec S.Wagstaff, baptisée "Early Abort Strategy" d'une complexité de $n^{\sqrt{1.5 \ln\ln n/\ln n)}} = \exp(\sqrt{1.5(\ln n)\ln\ln n})$.

7 La rho méthode de Pollard

Elle est aussi appelée méthode de Monte Carlo, car elle fait appel à un choix aléatoire (souvent en analyse numérique, c'est ainsi que de nombreuses méthodes sérendipiteuses, i.e. tirant partie du hasard, sont dites de Monte-Carlo, voire pour certains anglophones, de Las-Vegas), cet algorithme a été exposé par J.M. Pollard, *A Monte-Carlo method for factorization*, BIT, vol 15, 1975. On prend un polynôme f et on calcule $x_{i+1} := f(x_i) \bmod n$ où $x_0 = 1$ et on calcule $pgcd(x_i - x_k, n)$ pour obtenir un diviseur de n.

Exemple 7.1 *Tentons ainsi de factoriser* $n = 8051$ *avec le polynôme* $f(x) = x^2 + 1$.
On choisit à chaque fois $k = 2^{plancher(\ln(i)/\ln(2))} - 1$, *c'est lié à la méthode de détection de période de Brent.*

i	x_i	k	x_k	$\mathrm{pgcd}(x_i - x_k, n)$
1	2	0	1	1
2	5	1	2	1
3	26	1	2	1
4	677	3	26	1
5	7474	3	26	1
6	2839	3	26	97

Une fois de plus, on obtient $8051 = 97 * 83$. *On pourra utiliser notre programme sous Maple (annexe 2) pour vérifier ces calculs.*

Cet algorithme a une complexité de $n^{\sqrt{2.5 \ln\ln n / \ln n}} = \exp(\sqrt{2.5 (\ln n) \ln\ln n})$.

8 La méthode p-1 de Pollard

C'est encore à J.M. Pollard que l'on doit cette méthode, confer *Theorems on factorization and primality testing*, Proceedings Cambridge Phil. Soc., pp 521-528, 1974.

Elle est basée sur le principe simple suivant : si on a $p - 1 | k!$ alors, pour a premier avec p, on aura $p | a^{p-1} - 1 | a^{k!} - 1$.

Exemple 8.1 *Pour* $n = 8051$ *et en prenant* $a = 2$.

k	$2^{k!} \bmod n$	$\mathrm{pgcd}(n, 2^{k!} \bmod n)$
2	3	1
3	63	1
4	6982	1
5	2520	1
6	4268	97

Pas de surprise : on obtient $8051 = 97 * 83$. *On pourra utiliser notre programme sous Maple (annexe 2) pour vérifier ces calculs.*

Bien sûr, cette méthode n'est pas rentable si aucun des facteurs p n'est du type "$p - 1$ friable".

9 Factorisation par les courbes elliptiques

Méthode découverte par H.W. Lenstra Jr en 1985 et qu'il a exposée dans *Factoring integers with elliptic curves* , Report 86-18, Matematisch Instituut, Universiteit van Amsterdam, 1986.

Cette méthode achoppe sur certains nombres, toutefois c'est elle qui donne les meilleurs résultats pour trouver un facteur $< 10^{30}$ de n. On trouvera dans l'annexe 1 son implémentation par I.Vardi sous Mathematica ; elle est également implémentée sous Maple avec la commande `ifactor(n,lenstra)` ; renvoyons les lecteurs curieux à la lecture de [Atkin], [Koblitz], [Morain] et [Ribenboim].

Il appert ainsi qu'il existe de nombreux algorithmes de factorisations, pour lutter contre l'embarras du choix signalons que H. Riesel, dans *Prime numbers and computer methods for factorization*, Birkhaüser, 1985, propose un algorithme de factorisation qui comporte un algorithme pour choisir quelle méthode de factorisation employer ; c'est ce que l'on pourra appeler un méta-algorithme ou une métastratégie !

La recherche d'un algorithme rapide de factorisation n'a pas seulement comme intérêt d'enterrer la cryptographie, mais il permet aussi de servir de point de repère quant à l'évolution des performances des ordinateurs, tout comme le calcul des décimales de π ou comme le théorème de Fermat a pu servir de référence tout au long de l'histoire de la théorie des nombres.

10 Factorisation par crible quadratique

En 1984, in *The quadratic sieve factoring algorithm*, Lecture Notes in Comp. Sci., Springer, C. Pomerance propose son algorithme de factorisation par crible quadratique (Montgomery proposera une version légérement améliorée car elle utilise plusieurs polynômes : Multiple polynomial quadratic sieve ou MPQS).

La possibilité d'utiliser plusieurs polynômes a le grand mérite de permettre une parallélisation de l'algorithme (i.e. on lance l'algorithme sur plusieurs machines à travers le monde avec différents polynômes générateurs).

Les algorithmes parallèles ne sont qu'une version moderne de cet adage international : l'union fait la force.

C'est ainsi que Lenstra et Manasse ont développé un algorithme parallèle utilisant le courrier électronique et (avec entre autres Lenstra Jr, Pollard, Odlyzko, Pomerance et Morain) le 15 juin 1990, ils purent annoncer au monde, via ce même courrier la factorisation de $F_9 = 2^{2^9} + 1 = 2424833 * 7455602825647884208337395736200454918783366342657 *$ $7416400626275308015247871419019374740599407810975190239058213161444157595047050080928187116939407 37.$

Dans [Boender], on pourra trouver deux variantes de MPQS :
-PMPQS : single large prime variation of MPQS,
-PPMPQS : double large prime variation of MPQS.

PMPQS vient de partial MPQS : dans notre factorisation partielle, on ne retenait auparavant que les nombres friables, ici on s'autorise à retenir le dernier cofacteur lorsque celui-ci est premier.

PPMPQS vient de partial-partial MPQS, on s'autorise ici à retenir les deux derniers cofacteurs lorsque ceux-ci sont premiers.

Il s'agit une fois de plus de se ramener à $x^2 \equiv y^2 \bmod n$, pour ce faire on construit une suite de couples $a^2 \equiv b \bmod n$. Nous sommes alors assurés que $b = P(a) \approx 2a\sqrt{n}$ est petit si a l'est.

11 D'une méthode à l'autre

Le 2 avril 1994, un challenge, proposé par Martin Gardner en 1976 dans le Scientific American, fut relevé : non, il ne fallait pas plus d'un milliard d'années pour factoriser RSA129, un nombre de 129 chiffres, qui ne sut résister aux affres du crible quadratique.

Puis en 1996, le crible sur corps de nombres, découvert par Pollard, donnait les mêmes résultats, en six fois moins de temps.

Le 12 avril 1996, Arjen K Lenstra et son équipe annoncèrent par courier électronique la factorisation de
RSA130 = 18070820886874048059516561644059055662781025167694013491701270214...
50056662540244048387341127590812303371781887966563182013214880557
= 39685999459597454290161126162883786067576449112810064832555157243 $*$
45534498646735972188403686897274408864356301263205069600999044599
Il utilisa le crible sur corps de nombres avec le polynôme
$5748302248738405200X^5 + 9882261917482286102X^4 - 13392499389128176685X^3$
$+ 16875252458877684989X^2 + 3759900174855208738X - 46769930553931905995$ et sa racine 125 74411 16841 80059 80468 modulo RSA130.

Il estime le temps de calcul à 500 années MIPS. Le fichier contenant les relations détectées faisait plus de 3,5 Go. La matrice finale était de taille $3504823 * 3516502$.

De fait, les records de factorisation sont souvent soit des nombres du projet Cunningham, i.e. du type $b^n \pm 1$ avec b petit, soit des nombres de RSA Inc., qui sont bien plus durs que les précédents pour lesquels on a désormais des algorithmes performants, voir les nombreux résultats obtensibles à *ftp://nimbus.anu.edu.au/pub/Brent/factors*. RSA Inc. propose désormais la factorisation de nombres de partition, ou d'autres conçus exprès pour s'y casser les dents...

12 La méthode des corps de nombres

Toute cette section est traduite de [Elkenbracht-Huiwing]. L'algorithme de factorisation par crible sur corps de nombres (en anglais : number field sieve ou NFS) a été introduit par Pollard en 1988, c'est la méthode la plus rapide, elle se divise en fait en deux algorithmes distincts :
- le NFS spécialement adapté à des nombres du type $n = ar^t + bs^u$ (alias SNFS : special NFS),
- le NFS applicable à des nombres arbitraires (GNFS : general NFS).

Voici leur complexité :
pour le SNFS : $\exp((c + o(1))(\ln n)^{1/3}(\ln \ln n)^{2/3})$ où $c = \frac{32}{9}^{1/3} \approx 1.523$,
pour le GNFS : $\exp((c + o(1))(\ln n)^{1/3}(\ln \ln n)^{2/3})$ où $c = \frac{64}{9}^{1/3} \approx 1.923$.

Une version améliorée du GNFS avec beaucoup plus de polynômes ; due à D. Coppersmith, a une complexité de $\exp((c + o(1))(\ln n)^{1/3}(\ln \ln n)^{2/3})$ où $c = \frac{1}{3}(92 + 26\sqrt{13})^{1/3} \approx 1.902$.

C'est donc l'algorithme généraliste le plus rapide. Rappelons que le MPQS a une complexité de $\exp((1 + o(1))(\ln n)^{1/2}(\ln \ln n)^{1/2})$ reste néanmoins plus intéressant pour des nombres ayant moins de 105 chiffres. Soit n le nombre que nous souhaitons factoriser, dans la suite, nous supposerons n nombre composé (un test de primalité, confer [Morain], permet rapidement de trancher cette question) impair qui n'est pas une puissance de nombre premier.
Tout comme pour le MPQS, nous allons essayer de trouver une solution à l'équation $v^2 \equiv w^2 \bmod n$. Pour au moins la moitié des $v \bmod n, w \bmod n$ vérifiant l'équation et v, w premiers avec n, le pgcd de n et de $v - w$ donne un facteur non trivial de n. Pour obtenir v et w, nous allons d'abord choisir deux polynômes primitifs P_1 et P_2 de degré d_1 et d_2 à coefficients entiers et irréductibles sur $\mathbb{Z}$ (un polynôme est primitif si le pgcd de ses coefficients vaut 1). Soit m une racine commune, modulo n, de P_1 et P_2. Dans le SNFS, on arrive à obtenir de petits coefficients à partir de n, ce qui n'est pas le cas pour le GNFS, d'où une complexité supérieure pour ce dernier.

En général, au lieu de choisir au hasard un polynôme irréductible, on le construit de la manière suivante : on choisit son degré d_1 puis on prend $m = [n^{1/d_1}]$ puis on écrit n en base m, on a donc $n = c_{d_1} m^{d_1} + c_{d_1-1} m^{d_1-1} + \ldots + c_0$. Le tour est joué : il nous suffit de prendre $P_1(x) = c_{d_1} x^{d_1} + c_{d_1-1} x^{d_1-1} + \ldots + c_0$ et $P_2(x) = x - m$; on aura même $c_{d_1} = 1$ si $(2d_1)^{d_1} < n$ (voir néanmoins le chapitre suivant sur l'irréductibilité).

Soit α_i, pour $i = 1, 2$, une racine de P_i dans $\mathbb{C}$. Soit $\mathbb{Q}_n$ l'anneau des rationnels dont le dénominateur est premier avec n. Notre but est de trouver un ensemble E de paires (a, b) d'entiers premiers entre eux tels que $\Pi_E(a - b\alpha_1)$ et $\Pi_E(a - b\alpha_2)$ sont des carrés : β^2 dans $\mathbb{Q}_n[\alpha_1]$ et γ^2 dans $\mathbb{Q}_n[\alpha_2]$.

On peut rechercher les couples sur un pavé $[a \ldots a'] \times [b \ldots b']$, mais il existe aussi une méthode dite crible linaire où l'on ne fait varier que a et l'on prend $b = 1$. On fait appel à l'analyse pour déterminer les bornes de l'intervalles de manières à arriver à un compromis entre taille et effectivité (si l'on prend une fenêtre de crible trop petite, nous avons peu de chances d'obtenir de nombres friables).

Nous avons deux homomorphismes d'anneaux $\varphi_i : \mathbb{Q}_n[\alpha_i] \to \mathbb{Z}/n\mathbb{Z}$ définis par $\varphi_i(\alpha_i) = m \bmod n$. Nous avons alors $\varphi_1(\beta^2) \equiv \varphi_2(\gamma^2) \bmod n$ et donc $\varphi_1(\beta)^2 \equiv \varphi_2(\gamma)^2 \bmod n$, si $\varphi_1(\beta)$ et $\varphi_2(\gamma)$ sont premiers entre eux, nous retombons bien sur un facteur, à savoir $\mathrm{pgcd}(n, \varphi_1(\beta) - \varphi_2(\gamma))$, qui sera non trivial dans au moins la moitié des cas.

Une condition nécessaire pour que $\Pi_E(a - b\alpha_i)$ soit un carré dans $\mathbb{Q}_n[\alpha_i]$ est que sa norme $N(\Pi_E(a - b\alpha_i))$ soit un carré dans $\mathbb{Q}$. Notons, $F_i(x, y) = y^{d_i} P_i(x/y) \in \mathbb{Z}[x, y]$ la forme homogène de P_i. Puisque nous avons $N(a - b\alpha_i) = F_i(a, b)/c_{i,d_i}$, nous pouvons en déduire que si le cardinal de E est pair et si $\Pi_E F(a, b)$ est un carré dans $\mathbb{Z}$, alors $N(\Pi_E(a - b\alpha_i))$ est un carré dans $\mathbb{Q}$.

Le NFS cherche alors une paire (a, b) d'entiers premiers entre eux tels que les deux entiers $F_i(a, b)$ se factorisent complètement sur une base K_i de nombres premiers. On retombe alors dans le schéma décrit ci-dessus pour le PMQS.

Malheureusement, la condition que la norme soit un carré dans $\mathbb{Q}$ n'est pas suffisante pour qu'on ait un carré dans $\mathbb{Q}_n[\alpha_i]$.

Dans [Pomerance], on trouve aussi, la suite suivante d'objections :

1. $\mathbb{Z}[\alpha]$ peut être différent de l'anneau des entiers algébriques.
2. De plus, il est possible qu'il ne soit même pas un anneau de Dedekind, donc on n'aura pas la décomposition en idéaux premiers.
3. En outre, si nous avons une telle décomposition, il se peut que l'idéal produit soit le carré d'un idéal non principal.
4. De surcroît, même principal, il peut être engendré par un non entier algébrique (par exemple, c'est le cas de $-9\mathbb{Z}$).
5. Le comble serait qu'il soit engendré par un entier algébrique mais que cet entier ne soit pas dans $\mathbb{Z}[\alpha]$!
6. Et quand bien même, comment trouver cet entier, i.e. comment extraire une racine carrée dans $\mathbb{Z}[\alpha]$?

La cinquième objection est levée par le résultat suivant : si f est un polynôme unitaire irréductible sur $\mathbb{Z}$, alors toute racine complexe α de f et pour tout entier γ de $\mathbb{Q}(\alpha)$, $f'(\alpha)\gamma$ est dans $\mathbb{Z}[\alpha]$. Ce qui nous intéresse est que, si γ^2 est un carré dans les entiers de $\mathbb{Q}(\alpha)$, alors $f'(\alpha)^2\gamma^2$ est un carré dans $\mathbb{Z}[\alpha]$.

Bien que l'on connaisse déjà des algorithmes pour la sixième objection, des recherches sont toujours en cours pour améliorer ces extractions de racines carrées. On peut par exemple suivre Couveignes en calculant des racines modulo p puis remonter en utilisant le théorème chinois ; on peut aussi suivre Montgommery qui utilise une méthode itérative (une autre méthode due à Cohen est ici inapplicable car elle aboutit à des nombres gargantuesques).

Pour les autres objections, on va tester notre candidat $a - \alpha b$ sur un lot de caractères (par exemple le symbole de Legendre $(\frac{a}{b})$ pour différents p). Ainsi, si notre nombre est un carré dans tous ces corps, il aura de fortes chances d'être réellement un carré.

13 Irréductibilité d'un polynôme modulo n

Nous avons vu au chapitre précédent que l'on recherche des polynômes irréductibles sur $\mathbb{Z}$, ce qui sera bien le cas s'ils le sont dans $\mathbb{Z}/n\mathbb{Z}$.

Or, en pratique, si l'on prend un P de degré 5, alors P est irréductible modulo n ssi $P | X^{n^5} - X$ et $pgcd(X^n - X, P) = 1$ (voir [Naudin], p.129). C'est donc une étape peu coûteuse algorithmiquement. Mais, concrètement, si nous tombons sur polynôme non irréductible : quel drame ! Loin de là, en effet : il existe diverses méthodes rapides de factorisation de polynômes. On peut utiliser l'algo de Berlekamp (*Factoring polynomials over large finite fields*, Math. Comp., n°111, 1970) ou bien l'algo LLL de Lenstra, Lenstra et Lovasz. Le LLL est destiné aux polynômes primitifs et est en temps polynomial. Ainsi, nous avons donc deux polynômes Q et R tels que $P = QR$. D'où $n = P(m) = Q(m)R(m)$ et, en faisant confiance à [Pomerance], j'affirme qu'un résultat de Brillhart, Filaseta et Odlyzko nous dit que cette factorisation n'est pas triviale. Drôle de drame !

14 La méthode Lanczos par blocs

Nous avons vu, comment au cours des différents algorithmes de factorisation envisagés, nous retombions sur un problème classique d'algèbre linéaire : déterminer une relation de dépendance entre vecteurs. Comme dans de nombreux autres problèmes d'analyse numérique, la méthode la plus intéressante est une méthode itérative, et en raison de la structure particulière des matrices sur lesquelles nous travaillons, une méthode itérative par bloc : celle de Lanczos. Nous avons en premier lieu une liste de vecteurs donnant les exposants sur une base de nombres premiers, nous les prenons modulo 2 et enlevons les colonnes nulles. Il s'agit donc de trouver un vecteur du noyau (c'est un $\mathbb{F}_2$ espace vectoriel) de la matrice ainsi obtenue. A cause de la taille impressionnante (on prend souvent une base de nombres premiers de cardinal supérieur au million, condition désormais sine qua non pour être l'heureux détenteur d'un record de factorisation du projet Cunningham & Co) des matrices utilisées, même si elles sont creuses, les méthodes classiques comme l'élimination gaussienne [Knuth 4.6.2 algorithme N], non seulement sont trop coûteuses en mémoire mais aussi en temps. Une autre méthode envisageable est celle de Wiedemann par bloc, dans tous les cas, l'étape de détermination de vecteurs liés sur une matrice $k \times k$ peut se faire en moins de $O(k^3)$ étapes.

Détaillons maintenant la méthode standard de Lanczos sur $\mathbb{R}$. On part d'une matrice $k \times k$ A symétrique et définie positive. Pour résoudre $Ax = b$, on pose $w_0 = b$ et $w_i = Aw_{i-1} - \sum_{j=0}^{i-1} c_{ij} w_j$ où, pour $i > 0$, $c_{ij} = \frac{w_j^T A^2 w_{i-1}}{w_j^T A w_j}$. Après au plus k itérations, on a $w_i = 0$. Si l est la plus petite valeur telle que $w_l = 0$ on a donc $w_i^T A w_i \neq 0$ pour $0 \leq i < l$ et $w_j^T A w_i = 0$ pour $i \neq j$ et $A\mathcal{W} \subset \mathcal{W}$ où $\mathcal{W}$ est l'espace engendré par $w_0, \ldots, w_{l-1}$. On en déduit que

$$x = \sum_{i=0}^{l-1} \frac{w_i^T b}{w_i^T A w_i} w_i$$

est une solution de $Ax = b$. Puisque $w_j^T A^2 w_{i-1} = 0$ pour $j < i - 2$, nous pouvons plus directement calculer w_i pour $i \geq 2$ par $w_i = Aw_{i-1} - c_{i,i-1}w_{i-1} - c_{i,i-2}w_{i-2}$.

La méthode fonctionne également sur d'autres corps que $\mathbb{R}$, du moment que $w_i^T Aw_i \neq 0$ quand $w_i \neq 0$, l'avantage de travailler sur $\mathbb{F}_2$ est lié au fait que l'ordinateur compte en binaire, ou presque : l'unité de base traitée peut être 32 bits ou 64 bits... d'où une implémentation par blocs, chaque bloc correspond donc à un sous espace vectoriel $\mathcal{W}_i$.

Dans la méthode standard, dans la moitié des cas, la condition $w_i^T Aw_i \neq 0$ quand $w_i \neq 0$ n'est pas respectée, dans la méthode par blocs, on retrouve cette condition sous la forme $W_i^T AW_i$ inversible où les vecteurs colonnes de W_i engendrent $\mathcal{W}_i$.

On pourra se référer à P.L. Montgommery *A block Lanczos algorithm for finding dependences over GF(2)*, Eurocrypt'95, Springer, 1995.

15 RSA

Bien sûr, il ne nous reste plus qu'à signaler la principale victime des avancées des algorithmes de factorisation : c'est la cryptographie, qui, d'année en année, se voit obliger d'augmenter la taille de ses clefs, heureusement que les tests de primalités sont bien plus rapides que les différentes méthodes de factorisation [Morain] !

En effet, c'est bien à ce niveau que se joue l'avenir de la cryptographie, du moins de RSA, codage à clef publique inventé en 1977 par R.L. Rivest, A. Shamir & L. Adleman (confer *A method for obtaining digital signatures and Public-Key cryptosystems*, communications of the A.C.M., février 1978). Cette méthode commence par choisir deux grands nombres premiers p et q, ainsi qu'un nombre e, premier avec $(p-1)(q-1)$, on rend public n ($n = pq$ mais on se garde bien de le dire...) et e, tout le monde peut vous écrire $C = M^e \bmod n$, que vous seul pouvez lire en calculant $M = C^d \bmod n$, où d est tel que $de \equiv 1 \bmod (p-1)(q-1)$, il est trop difficile d'obtenir d pour qui ne connaît pas la factorisation $p * q$ de n ; c'est ce qui fait la fiabilité de RSA tant que nous ne connaissons pas d'algorithme rapide de factorisation.

D'ailleurs on ne sait rien sur la difficulté intrinsèque de la factorisation, est-ce un problème de classe P ou NP, mystère... On conjecture néanmoins que les meilleurs algorithmes seraient du type $n^{\epsilon(n)}$ avec $\epsilon(n) \to 0$ quand $n \to \infty$. Les algorithmes actuels sont déjà de cette forme, marchant même souvent mieux que prévu, mais, comme le dit C. Pomerance avec humour, heureusement que les nombres que nous tentons de factoriser, eux, ne le savent pas !

16 Bibliographie

Atkin, A. O. L. & Morain, F., *Finding suitable curves for the elliptic curve method of factorization*, Math. Comp., janvier 1993
Boender, H. & te Riele, H., *Factoring integers with PMPQS*, Experimental math., n°4, 1996
Chabert, J.L. & coll., *Histoire d'algorithmes*, Belin, 1994
Dahan-Dalmedico, A. & Peiffer, J., *Une histoire des mathématiques*, Seuil, 1986
Elkenbracht-Huizing, M., *An implementation of the NFS*, Experimental math., n°3, 1996
Guy, R.K., *Unsolved problems in number theory*, Springer, 1994
Hardy G.H. & Wright E.M., *An introduction to the theory of numbers*, Clarendon Press, 1959
Knuth, D.E., *Seminumerical algorithms*, Addison-Wesley, 1981
Koblitz, N., *A course in number theory and cryptography*, Springer, 1987
Legendre, A.M., *Théorie des nombres*, Blanchard, 1955
Morain, F., *Courbes elliptiques et tests de primalité*, Thèse, Université de Lyon I, 1990
Morain, F., *Analysing PMPQS*, informal note, 1993
Naudin, P. & Quitté, C., *Algorithmique algébrique*, Masson, 1992
Pomerance, C., *A tale of two sieves*, Notices of AMS, décembre 1996
Ribenboim, P., *The new book of prime number records*, Springer, 1996

17 Annexe 1 : factorisation par les courbes elliptiques sous Mathematica

18 Annexe 2 : factorisation par les méthodes p-1 et rho de Pollard sous Maple

19 Annexe 3 : factorisation par les fractions continuées sous Maple