Factorisation par fractions continues

Margot Funk, Antoine Hugounet

Février 2021

Table des matières

1 Théorie

1.1 Factorisation par fractions continues

Dans tout le reste de cette section, N désigne un entier naturel composé impair.

1.1.1 Méthodes de Fermat et Kraitchik

La méthode de factorisation de Fermat part du constat suivant.

Lemme 1.1. *Factoriser N est équivalent à l'exprimer comme différence de deux carrés d'entiers.*

Démonstration. Si $N = u^2 - v^2, u, v \in \mathbb{Z}$ alors $N = (u - v)(u + v)$. Réciproquement si l'on a une factorisation $N = ab$, alors $N = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2$. □

La méthode de Fermat exploite cette propriété et se montre particulièrement efficace lorsque N est le produit de deux entiers proches l'un de l'autre. Notons $N = ab$ une factorisation de N avec a et b proches, $r = \frac{a+b}{2}$, $s = \frac{a-b}{2}$ de sorte que

$$N = r^2 - s^2.$$

Comme s est petit en valeur absolue par hypothèse, l'entier r est donc plus grand que $\sqrt{N}$ tout en lui étant proche. Il existe donc un entier positif u *pas trop grand* tel que

$$\lfloor \sqrt{N} \rfloor + u = r$$

et donc tel que $(\lfloor \sqrt{N} \rfloor + u)^2 - N$ soit un carré. Trouver un tel entier u donne alors la factorisation de N. Comme les facteurs de N sont proches l'un de l'autre, on le trouve par essais successifs.

La méthode de Fermat est cependant inefficace lorsque les facteurs de N ne sont pas proches. D'après [**Tale**] ¶ *Fermat and Kraitchik*, la méthode est alors encore plus coûteuse que la méthode des divisions successives. Dans les années 1920, Maurice Kraitchik a amélioré l'efficacité de la méthode de Fermat. Son idée essentielle est que pour factoriser N, il *suffit* de trouver une différence de deux carrés qui soit un multiple de N.

Lemme 1.2. *Connaître deux entiers $u, v \in \mathbb{Z}$ tels que $u^2 \equiv v^2 \pmod{N}$ et $u \not\equiv \pm v \pmod{N}$ fournit une factorisation de N. Plus spécifiquement, les entiers $\operatorname{pgcd}(u-v, N)$ et $\operatorname{pgcd}(u+v, N)$ sont des facteurs non triviaux de N.*

Démonstration. Posons $g = \operatorname{pgcd}(u - v, N)$ et $g' = \operatorname{pgcd}(u + v, N)$. Comme $u \not\equiv \pm v \pmod{N}$, on a $g < N$ et $g' < N$. Enfin ni g et g' ne sont réduits à 1 : si l'un deux l'est, l'autre vaut N, contradiction. Donc g et g' sont tous deux des facteurs non triviaux de N. □

Remarque 1.3. Dans l'algorithme, nous nous contenterons de chercher des u, v tels que N divise la différence de leurs carrés, sans vérifier s'ils vérifient $u \not\equiv \pm v \pmod{N}$. Comme le polynôme $X^2 - v^2 \in \mathbb{Z}/N\mathbb{Z}$ a exactement quatre racines, il y a « une chance sur deux » pour que u et v nous fournissent un facteur non trivial de N.

Pour factoriser N, il s'agit donc de trouver de tels couples (u, v). Kraitchik cherche pour cela des couples $(u_i, v_i)_{1 \leqslant i \leqslant r}$ vérifiant

$$u_i^2 \equiv v_i \pmod N$$

et tels que l'entier $\prod_{i=1}^r v_i$ soit un carré (dans $\mathbb{Z}$). Posant $u = \prod_{i=1}^r u_i$ et $v = \sqrt{\prod_{i=1}^r v_i}$, il vient

$$v^2 \equiv u^2 \pmod N.$$

Pour chercher ces couples $(u_i, v_i)_{1 \leqslant i \leqslant r}$, Kraitchik propose d'utiliser le polynôme $K := X^2 - N \in \mathbb{Z}[X]$ qui fournit la congruence $u_i^2 \equiv K(u_i) \pmod N$ pour tout $u_i \in \mathbb{Z}$. Une question reste cependant en suspens : comment trouver en pratique des éléments $K(u_i)$ dont le produit est un carré ?

1.1.2 Recherche de congruences de carrés

Morrison et Brillhart apportent dans l'article **ref** une réponse à cette question. Leur méthode est basée sur la connaissance de congruences de la forme

$$u_i^2 \equiv Q_i \pmod N,$$

où $|Q_i|$ est suffisamment petit pour être factorisé. [1] On fixe pour cela B une base de factorisation, c'est à dire un ensemble non vide fini de nombres premiers, et l'on dit qu'un entier $Q \in \mathbb{N} \setminus \{0, 1\}$ est B-*friable* si tous ses facteurs premiers sont dans B. La connaissance de suffisamment d'entiers Q_i qui sont B-friables permet de factoriser N.

Définition 1.4. Soient $B = \{p_1, \ldots, p_m\}$ une base de factorisation et $Q \in \mathbb{Z}$ un entier dont la valeur absolue est B-friable. Q s'écrit alors

$$Q = (-1)^{v_0} \prod_{i=1}^{m} p_i^{v_{p_i}(Q)}.$$

Puisque les éléments de B sont fixés et en nombre fini, l'élément Q peut être vu comme le vecteur des valuations $(v_{p_m}(Q), \ldots, v_{p_1}(Q), v_0) \in \mathbb{N}^{m+1}$.

On appelle B-*vecteur exposant de* Q et l'on note $v_B(Q)$ le vecteur [2]

$$v_B(Q) := (v_{p_m}(Q), \ldots, v_{p_1}(Q), v_0) \in \mathbb{F}_2^{m+1}.$$

1. La notation Q_i fera échos à celle utilisée pour décrire la méthode de factorisation avec les fractions continues.

2. Notez qu'il s'agit d'un élément de $\mathbb{F}_2^{m+1}$: seule la parité des valuations nous intéresse. L'élément v_0 est placé à droite et non au début car il correspondra au bit de poids faible du B-vecteur exposant dans le code.

Proposition 1.5. *Soit F une famille d'entiers dont les valeurs absolues sont B-friables. Si*

$$\#F \geqslant \#B + 2$$

alors on peut extraire une sous-famille de F dont le produit des éléments est un carré.

Démonstration. Posons $F = \{Q_1, \ldots, Q_k\}$ (de sorte que $k = \#F$) et $B = (p_1, \ldots, p_m)$ (de sorte que $\#B = m$). Par hypothèse de friabilité, on peut associer à chaque Q_j, $1 \leqslant j \leqslant k$, un B-vecteur exposant.

Fixons $j, j' \in [\![1, k]\!]$. L'entier Q_j est un carré si, et seulement si, les composantes de son vecteur de valuations $(v_{p_m}(Q), \ldots, v_{p_1}(Q), v_0) \in \mathbb{N}^{m+1}$ sont paires, i.e. si son B-vecteur exposant est nul. Par propriété des valuations, le B-vecteur exposant associé au produit $Q_j \cdot Q_{j'}$ est le vecteur somme $v_B(Q_j) + v_B(Q_{j'})$. Autrement dit, le produit d'une sous-famille $\{Q_{j_1}, \ldots, Q_{j_s}\}$ de F est un carré si, et seulement si, la somme des B-vecteurs exposants $v_B(Q_{j_1}), \ldots, v_B(Q_{j_s})$ est nulle. Soit V le $\mathbb{F}_2$-espace vectoriel $\mathbb{F}_2^{m+1}$, qui est de dimension $m+1$. Comme $k \geqslant m+2$, la famille $\{v_B(Q_1), \ldots, v_B(Q_k)\}$ est liée dans V et il existe de fait des éléments $l_1, \ldots, l_k \in \mathbb{F}_2$ tels que

$$\sum_{j=1}^{k} l_j v_B(Q_j) = 0.$$

L'élément $\prod_{j=1}^{k} Q_j^{l_j}$ est alors un carré. $\square$

Étant données des congruences de la forme $u_i^2 \equiv Q_i \pmod N$, la preuve de la proposition fournit un procédé d'algèbre linéaire pour extraire une sous-famille des Q_i dont le produit des éléments est un carré. On trouve tout d'abord des $Q_{i_1}, \ldots, Q_{i_k}$ dont la valeur absolue est B-friable[3]. Soit M la matrice

$$M := \begin{pmatrix} v_B(Q_{i_1}) \\ \vdots \\ v_B(Q_{i_k}) \end{pmatrix} \in \mathcal{M}_{k,\#B+1}(\mathbb{F}_2).$$

Soient $l_1, \ldots, l_k$ les éléments de $\mathbb{F}_2$ donnés dans la preuve de la proposition tels que

$$\prod_{j=1}^{k} Q_{i_j}^{l_{i_j}}$$

est un carré. Le vecteur $(l_1, \ldots, l_k)$ est un élément du noyau de la matrice transposée de M. Il peut donc être exhibé par pivot de Gauß.

3. Nous le ferons en factorisant les Q_i à disposition par divisions successives.

1.1.3 Utilisation des fractions continues

L'introduction des fractions continues est motivée par le constat suivant. Si

$$u_i^2 = Q_i + kNb^2, \quad u_i, Q_i, b \in \mathbb{Z}, k \in \mathbb{N}^*$$

de telle sorte que $|Q_i|$ soit petit, alors

$$\left(\frac{u_i}{b}\right)^2 - kN = \frac{Q_i}{b^2}$$

est petit en valeur absolue et $\frac{u_i}{b}$ est une bonne approximation de $\sqrt{kN}$ **réf**. Fixons k un entier naturel non nul, posons $x = \sqrt{kN}$ et reprenons les notations **??** et celles développées à la fin de la sous-section **??**. En vertu de l'identité

$$A_{n-1}^2 \equiv (-1)^n Q_n \pmod N,$$

nous appelons *méthode de factorisation des fractions continues* la méthode de Kraitchik dans laquelle les entiers u_i sont donnés par les A_{i-1} et les v_i par les $(-1)^i Q_i$.

Ce choix de congruences est intéressant car on sait que les Q_n sont majorés (**??**) par $2\sqrt{kN}$. A l'inverse, les $x^2 - N \in \mathbb{Z}[X], x \in \mathbb{N}$ ont une croissance linéaire de pente $2\sqrt{N}$ lorsque x s'éloigne de $\sqrt{N}$. Pour une base de factorisation B fixée, les Q_n auront donc plus de chance d'être B-friables que les $K(x)$ de Kraitchik. Or, l'étape la plus coûteuse de l'algorithme est celle de la recherche des termes B-friables par divisions successives. Notons d'autre part qu'il est facile de générer le développement en fraction continue de x et les paires (A_{n-1}, Q_n) grace à un algorithme itératif dû à Gauß et exposé dans **réf**.

L'égalité $A_{n-1}^2 - kNB_{n-1}^2 = (-1)^n Q_n$ donne un critère pour sélectionner les premiers de la base de factorisation : les premiers p divisant Q_n vérifient nécessairement

$$\left(\frac{kN}{p}\right) = 0 \text{ ou } 1$$

En effet, supposons qu'un premier p divise Q_n. On a alors $A_{n-1}^2 \equiv kNB_{n-1}^2 \pmod p$. Comme $\mathrm{pgcd}(A_{n-1}, B_{n-1}) = 1$, p ne peut pas diviser B_{n-1} (sinon $A_{n-1}^2 \equiv 0 \pmod p$ et p diviserait aussi A_{n-1}). B_{n-1} est donc inversible modulo p et $\left(\frac{A_{n-1}}{B_{n-1}}\right)^2 \equiv kN \pmod p$.

1.1.4 Quelques éléments pour appréhender la complexité de la méthode

2 Explication du programme

2.1 Architecture du programme

2.1.1 Terminologie

Posons pour commencer quelques définitions qui seront utiles pour décrire le code.

Définition 2.1. On dira qu'un couple (A_{n-1}, Q_n) est une *paire* (A, Q). .

Définition 2.2. Un ensemble de paires (A, Q) indexé par $n_1, \ldots, n_k$ est dit *valide* si le produit $\prod_{i=1}^{k}(-1)^{n_i}Q_{n_i}$ est un carré (dans $\mathbb{Z}$ et non uniquement dans $\mathbb{Z}/N\mathbb{Z}$).

Définition 2.3. Si B est la base de factorisation utilisée par le programme, on désignera par l'expression *vecteur exposant associé à* Q_n le B-vecteurs exposant $v_B((-1)^n Q_n)$.

2.1.2 Structure générale

Notre programme comprend deux étapes principales. La première consiste à générer, à partir du développement en fractions continues de $\sqrt{kN}$, des paires (A, Q) avec Q_n friable pour une base de factorisation préalablement déterminée. On associe à chaque Q_n ainsi produit son vecteur exposant `mpz_t exp_vect`. Ce vecteur permet de retenir les nombres premiers qui interviennent dans la factorisation de Q_n avec une valuation impaire. Dans le but d'augmenter le nombre de paires (A, Q) acceptées lors de cette étape, nous avons implémenté la ≪ large prime variation ≫. Celle-ci permet d'accepter une paire si Q_n se factorise grâce aux premiers de la base de factorisation et à un grand facteur premier supplémentaire. Les fonctions de cette phase de collecte sont rassemblées dans le fichier `step_1.c`. Elles font appel, pour mettre en oeuvre la ≪ large prime variation ≫, aux fonctions du fichier `lp_var.c`.

Ces données sont traitées lors de la seconde phase dans l'espoir de trouver un facteur non trivial de N. Il s'agit de trouver des ensembles valides de paires (A, Q) par pivot de Gauss sur la matrice dont les lignes sont formées des vecteurs exposants. Chaque ensemble valide est à l'origine d'une congruence de la forme $A^2 \equiv Q^2 \pmod{N}$ permettant potentiellement de trouver un facteur non trivial de N. Les fonctions de cette phase sont regroupées dans le fichier `step_2.c`.

Avant d'effectuer la première étape, il convient de se doter d'une base de factorisation. Ceci est permis par une des fonctions de `init_algo.c`. Ces dernières se chargent plus généralement de l'initialisation et du choix par défaut des paramètres.

Finalement, en mettant bout à bout les deux étapes, la fonction `contfract_factor` du fichier `fact.c` recherche un facteur non trivial de N et `print_results` affiche les résultats.

2.1.3 Entrées et sorties

Nous avons regroupé dans une structure `Params` les paramètres d'entrée de la fonction de factorisation, à savoir :

- `N` : le nombre à factoriser, supposé produit de deux grands nombres premiers.
- `k` : le coefficient multiplicateur.
- `n_lim` : le nombre maximal de paires (A, Q) que l'on s'autorise à calculer. Ce nombre prend en compte toutes les paires produites et non uniquement les paires avec Q_n friable ou produites par la « large prime variation ».
- `s_fb` : la taille de la base de factorisation.
- `nb_want_AQp` : le nombre désiré de paires (A, Q) avec Q_n friable ou produites par la « large prime variation ».
- des booléens indiquant si la « early abort strategy » ou la « large prime variation » doivent être utilisées et des paramètres s'y rapportant.

Le programme stocke dans une structure `Results` un facteur non trivial de `N` trouvé (si tel est le cas) ainsi que des données permettant l'analyse des performances de la méthode.

Remarque 2.4. L'efficatité de la méthode dépend du choix des paramètres ci-dessus. Pour avoir plus de latitude dans les tests, nous les considérons comme des paramètres d'entrée du programme. C'est pourquoi notre programme ne s'attèle pas à la factorisation complète d'un entier, qui aurait nécessité une sous-routine déterminant des paramètres optimaux en fonction de la taille de l'entier dont on cherche un facteur.

Remarque 2.5. Notre programme n'est pas supposé prendre en entrée un nombre admettant un petit facteur premier (inférieur aux premiers de la base de factorisation par exemple). En effet, comme il ne teste pas au préalable si `N` est divisible par de petits facteurs, il mettra autant de temps à trouver un petit facteur qu'un grand facteur.

2.2 Pivot de Gauss et recherche d'un facteur non trivial : `step_2.c`

Avant de nous pencher sur les détails de la phase de collecte, regardons l'implémentation de la seconde phase, qui aide à mieux comprendre la forme sous laquelle nous collectons les données. A l'issue de la première phase, on espère avoir collecté `nb_want_AQp` paires

(A, Q) avec Q_n friable [4]. Le nombre réel de telles paires est stocké dans le champ `nb_AQp` d'une structure `Results`.

Les données de ces `nb_AQp` paires sont stockées dans quatre tableaux : `mpz_t *Ans`, `mpz_t *Qns`, `mpz_t *exp_vects` et `mpz_t *hist_vects`. A un indice correspond un paire (A, Q) donnée.

2.3 Collecte des paires (A, Q) : `step_1.c` et `lp_var.c`

Décrivons à présent la phase de collecte des données. Le nombre de paires (A, Q) collectées sera stockée dans le champ texttnb_AQp de la structure `Results`. Concernant les vecteurs historiques `exp_vect`, il suffit d'initialiser `exp_vects[i]`

évoquer à un moment `init_hist_vects`. Le reste demande un peu plus d'explications

2.3.1 La fonction `create_AQ_pairs`

Sachant que seules les paires (A, Q) dont on a pu factoriser Q_n nous intéressent pour la seconde phase, nous avons décidé de ne stocker que celles-ci. Ce choix a en outre un avantage : étant donné un nombre `nb_want_AQp` représentant le nombre voulu de telles paires, il est possible d'arrêter le développement en fraction continue dès que ce nombre est atteint. Cela évite d'avoir à stocker toutes les paires (A, Q), pour ensuite sélectionner celles qui nous intéressent, en courant le risque d'en avoir trop ou pas assez.

Ce choix amène à avoir une grande fonction, en l'occurence `create_AQ_pairs`, qui au fur à mesure du développement de $\sqrt{kN}$ en fraction continue, teste si le Q_n qui vient d'être calculé est factorisable. Si c'est le cas, on crée son vecteur exposant et ajoute les données de la paire aux tableaux `Ans`, `Qns` et `exp_vects`. Pour ce faire, la fonction utilise les sous-routines `is_Qn_factorisable` et `init_exp_vect`.

2.3.2 La « early abort strategy »

La fonction `is_Qn_factorisable` teste si un Q_n est friable [5] par divisions successives avec les premiers de la base de factorisation. Un moyen d'améliorer les performances de la méthode est de décider de ne pas poursuivre les divisions successives si après un nombre `eas_cut` de divisions la partie non factorisée de Q_n est trop grande (supérieure à une borne `eas_bound_div` proportionnelle à la borne déjà connue $\sqrt{kN}$).

---

4. ou résultant de la « large prime variation » mais cela n'a aucune incidence sur les fonctions de cette partie

5. ou presque friable, voir paragraphe suivant.

2.3.3 La « large prime variation »

Etant donnée une base de factorisation $B = \{p_1, \cdots, p_m\}$, la « large prime variation » consiste à accepter lors de la collecte, non seulement des Q_n B-friables mais aussi des Q_n produits d'un entier B-friable et d'un entier lp_n inférieur à p_m^2. On dira que Q_n est *presque friable* et l'on appelera *grand premier (large prime)* le premier lp_n en question.

Pour que des Q_n presque friables soient exploitables, il faut qu'ils aient un grand premier lp en commun. En effet, si on trouve deux entiers presque friables $Q_{n_1} = X_{n_1}lp$ et $Q_{n_2} = X_{n_2}lp$, on peut former une nouvelle paire (A, Q) avec laquelle on peut travailler pour chercher une congruence de carrés.

Remarquons pour cela qu'on a les conguences :

$$\begin{cases} A_{n_1-1}^2 \equiv (-1)^{n_1} X_{n_1} lp \pmod N \\ A_{n_2-1}^2 \equiv (-1)^{n_2} X_{n_2} lp \pmod N \end{cases}$$

En les multipliant, on obtient :

$$(A_{n_1-1}A_{n_2-1})^2 \equiv \underbrace{(-1)^{n_1+n_2} X_{n_1} X_{n_2}}_{\substack{\text{associé au vecteur exposant} \\ v_B\left((-1)^{n_1}X_{n_1}\right)+v_B\left((-1)^{n_2}X_{n_2}\right)}} \underbrace{lp^2}_{\text{carré qui ne pose pas problème}} \pmod N$$

On forme donc la nouvelle paire $(A_{n_1-1}A_{n_2-1} \pmod N, Q_{n_1}Q_{n_2})$ associée au vecteur exposant $v_B\Big((-1)^{n_1}X_{n_1}\Big) + v_B\Big((-1)^{n_2}X_{n_2}\Big)$. Elle sera traitée lors de la deuxième phase exactement de la même manière que les paires « classiques ».

En pratique, pour repérer les paires qui ont le même grand premier, nous constituons au fur et à mesure de la collecte une liste chainée dont les noeuds stockent les données d'une paire dont le Q_n est presque friable (les entiers Q_n, A_{n-1}, le vecteur exposant et le grand premier associé à Q_n). Nous maintenons cette liste triée par taille des grands premiers. Lorsque que survient un Q_n presque friable, il est repéré par la fonction `is_Qn_factorisable` qui fournit également son grand premier lp. La liste chainée est alors parcourue pour savoir si l'on a déjà rencontré ce lp. Deux cas se présentent alors. Si lp est absent de la liste, on crée à la bonne place un noeud. Si lp est déjà présent dans la liste, au lieu de rajouter un noeud, on utilise le noeud possédant ce lp pour obtenir une nouvelle paire (A, Q) selon la méthode énoncée plus haut et ajoute ses composantes aux tableaux `Ans`, `Qns` et `exp_vects`. La fonction `insert_or_elim_lp` se charge de cela.