Certification en temps quadratique en algèbre linéaire

Kévin Polisano

14/12/2012

1 Introduction

En calcul numérique ou hybrique symbolique-numérique - comme utilisé sur Maple - il est souvent utile de certifier *a posteriori* le résultat O (pour Output) d'un calcul sur des données I (pour Input). La **certification** (ou validation) consiste donc à partir de l'entrée I et d'un certificat $C(I)$ (c'est-à-dire d'une structure de données annexe, dépendant de I) de tester si le résultat O est correct, et ce avec une complexité meilleure que n'importe quel autre algorithme qui ferait de même sans l'aide du certificat.

Dans cet article, les auteurs présentent des certificats pour des calculs d'algèbre linéaire portant sur une matrice A définie positive $n \times n$ dont les coefficients sont des entiers de longueur en bits $log\|A\|$. La certification du résultat grâce à ces certificats (dont on s'efforce qu'ils occupent $O(n^{3+\epsilon}(log\|A\|)^{1+\epsilon}$ bits en mémoire) doit être réalisée en complexité temporelle $O(n^{2+\epsilon}(log\|A\|)^{1+\epsilon})$, c'est-à-dire que l'on souhaite que la validation soit essentiellement quadratique en n.

On s'autorise des certifications probabilistiques, c'est-à-dire des procédés aléatoires de vérification de type Monte Carlo, qui renvoient "Vrai" si le résultat est considéré exact à l'aide du certificat, et dont la probabilité d'obtenir "Vrai" en sortie alors que le résultat et/ou le certificat était incorrect soit au moins grossièrement inférieure à $1/2$. Si on répète l'expérience k fois de manière indépendante on obtient alors une probabilité d'erreur de $1/2^k$, on peut alors rendre cette probabilité aussi petite que l'on veut.

2 Exemple typique de certification

Nous allons traiter un exemple simple d'algorithme de certification, du à Rusins Freivalds, qui va servir de modèle pour les suivants. On souhaite vérifier qu'une certaine expression matricielle E est nulle. Par exemple, dans le but de vérifier que le résultat de la décomposition LU d'une matrice A est correcte, on voudrait s'assurer qu'on a bien $A = LU$ soit encore $E = A - LU = 0$.

La première idée naïve qui nous vient à l'esprit est alors d'effectuer le produit matriciel $L \times U$ et de comparer terme à terme avec les coefficients de A. Seulement, d'un point de vue complexité, cela n'est pas très judicieux. En effet, on a vu en cours que la décomposition LU (tout comme la plupart des problèmes en algèbre linéaire dense) peut s'effectuer en $O(n^\omega)$ avec $\omega > 2.37$ en se ramenant à ce que l'on sait faire de mieux, à savoir le produit matriciel. Ainsi, vérifier le résultat en effectuant un produit matriciel coûterait aussi cher que la décomposition LU elle-même.

L'idée, elle aussi utilisée mainte fois en cours, est d'évaluer l'expression en des vecteurs v et de vérifier que $Ev = Av - L(Uv) = 0$ ce qui ne coûte que 3 multiplications matrice/vecteur et une soustraction donc a une complexité en $O(n^2)$ bien plus satisfaisante. Faire cette vérification en calcul numérique (soit dans $\mathbb{Z}$) a très peu de chance de nous donner un vecteur Ev nul (du aux résidus/précision machine), c'est pourquoi on effectue cette certification (et toutes les suivantes) modulo p (ou plus généralement dans un corps finis), c'est-à-dire en prenant un vecteur aléatoire $v \in \mathbb{Z}_p^n$. Ici le certificat est donc le couple (p, v) et la certification est bien probabilistique puisque que l'on tire un vecteur v aléatoirement.

Le problème qui se pose, et qui se posera également dans toutes les certifications suivantes, est que l'on peut obtenir par ce procédé une expression Ev nulle alors que le résultat de la décomposition LU était incorrect, par exemple si on avait $Ev = (p, p, ..., p)^t \neq 0$ (dans $\mathbb{Z}$) alors, puisque l'on travaille modulo p, on trouverait bien $Ev \equiv 0$. Autrement dit l'algorithme de vérification nous aurait renvoyait "Vrai" bien que le résultat était incorrect. Il faut donc s'assurer que la probabilité d'être dans cette situation est raisonnable. Dans la suite on s'imposera une probabilité d'erreur max de $1/2$, mais dans le cas présent de la certification de Freivalds on peut montrer que cette probabilité est de $1/p$. En effet si l'expression matricielle E est non nulle, son noyau est de dimension au plus $n - 1$ dans $\mathbb{Z}_p^n$, donc contenant au plus p^{n-1} vecteurs parmi les p^n possibles. D'où une probabilité d'erreur de $p^{n-1}/p^n = 1/p$.

3 Certification de singularité et consistance

Théorème 1. *Soit $A \in \mathbb{Z}^{n\times n}$ et $b \in \mathbb{Z}$. Les problèmes suivants possèdent un certificat de taille $n^{2+o(1)}(log\|[A,b]\|)^{o(1)}$ et une certification probabilistique en $n^{2+o(1)}(log\|[A,b]\|)^{o(1)}$:*

1. *Non singularité de A*
2. *Singularité de A*
3. *Consistance du système linéaire $Ax = b$*
4. *Inconsistance du système linéaire $Ax = b$*

Nous allons préciser pour chacun de ces 4 problèmes de quels certificats il s'agit et quel est l'algorithme de certification associé, en prouvant que leur complexité, respectivement spatiale et temporelle, est de $n^{2+o(1)}(log\|[A,b]\|)^{o(1)}$.

3.1 Certification de la non singularité

On rappelle que le certificat est la structure de données qui va permettre à l'algorithme de vérification du résultat (la certification) de s'exécuter en un temps essentiellement quadratique. Ici le certificat considéré est la structure de données suivante (p, B) où p est un nombre premier et B est la supposée inverse de A réduite modulo p : $B = A^{-1}\ mod\ p$. C'est-à-dire qu'on a réussi d'une manière ou d'une autre à obtenir une matrice que l'on pense être l'inverse de A (par des calculs numériques par exemple) et on veut s'assurer qu'elle ne résulte pas de calculs inexacts, et qu'en réalité A n'est pas inversible.

Comme précédemment nous allons travailler modulo p. Disposant de ce certificat (p, B), l'algorithme de certification est tout simplement la vérification de l'équivalence à zéro de l'expression $AB - I$ modulo p que nous venons d'expliquer et qui nous le savons maintenant s'effectue en temps quadratique. La complexité temporelle est donc bien celle annoncée. Reste à voir quelle place occupe le certificat (p, B) en mémoire. Cela dépend uniquement de la taille du nombre premier p considéré puisque les coefficients de la matrice B sont tous inférieurs à p (car réduits modulo p). Donc la taille du certificat est $|(p, B)| = |p| + n^2|p|$ où $|p| = log_2(p)$ est la taille du nombre premier p en nombre de bits.

On pourrait alors se dire que le nombre premier de plus petite taille $p = 2$ ferait l'affaire et réduirait au max la taille du certificat. Le problème est que,

on ne peut pas choisir n'importe quel p. En effet pour des nombres premiers p qui divisent $det(A)$ la matrice A n'est pas inversible modulo p donc la vérification modulo p de $AB - I \equiv 0 \ mod \ p$ est obsolète. La certification va renvoyer faux alors que la matrice pouvait tout à fait être inversible dans $\mathbb{Z}$. Il faut donc éviter de choisir de tels p, pour cela on va donc chercher à savoir combien il y a au maximum de diviseurs premiers de $det(A)$, disons M. Il suffira alors de prendre p parmi les $2M$ premiers nombres premiers pour avoir une probabilité au plus $1/2$ de tomber sur un p pathologique.

Sachant qu'on ne dispose pas de la valeur de $det(A)$, on ne peut pas compter exactement le nombre de premiers divisant $det(A) \in \mathbb{Z}$ mais on va pouvoir en donner une majoration. Dans un premier temps on va donc majorer $|det(A)|$ par une constante C_n, puis on cherchera le nombre maximum de nombre premiers divisant cette constante.

Commençons par déterminer le nombre maximum de nombres premiers divisant un entier q donné. On va montrer qu'il y en au plus $ln(q)$. Soit $p_1, ..., p_k$ les diviseurs premiers de q. Comme tous les $p_i \geqslant 2$ on a trivialement l'inégalité suivante : $2^k \leqslant p_1 \cdots p_k \leqslant q$ donc $k \ln 2 \leqslant \ln q$. On n'a donc pas tout à fait ce que l'on voulait, à savoir $k \leqslant ln(q)$. Il faut donc affiner davantage la majoration. Si $k \geq 3$ alors $e^k \leqslant 3^k \leqslant (2 \times 5) \times 3^{k-2} \leqslant (p_1 p_3)(p_2 p_4 p_5 p_6 \cdots) = p_1 p_2 \cdots p_k \leqslant p$ et on obtient par passage au log la borne attendue.

L'inégalité d'Hadamard fournit la réponse au premier point, le déterminant en valeur absolue est borné par la constante (dépendant de la dimension n)

$$|det(A)| \leqslant (n^{1/2} \|A\|)^n$$

On en conclut qu'il existe au plus

$$M = log((n^{1/2} \|A\|)^n) = n(log(n)/2 + log\|A\|)$$

qui divise le déterminant $det(A)$.

Contrairement à ce que je suggérais, c'est-à-dire prendre p dans l'ensemble des $2M$ premiers nombres premiers pour assurer une probabilité de $1/2$ de choisir un mauvais p, ils prennent quant à eux un p de longueur $|p| = log_2(M) \sim log(n)$ bits de façon à ce que le certificat ait bien une taille $|(p, B)| = (n^2 + 1)|p|$ en $n^{2+o(1)}(log\|A\|)^{o(1)}$. C'est-à-dire que p est dans l'intervalle $[2, M]$ (à ne pas confondre avec l'ensemble des M premiers nombres premiers!) qui comporte beaucoup moins de nombres premiers, asymp-

totiquement $M/log(M)$. Ceci étant ma suggestion est quasi linéaire en nombres de bits (voir *).

3.2 Certification de la singularité

Le certificat est une suite de $2M$ nombres premiers p_i et vecteurs v_i non nuls vérifiant $Av_i = 0 \ mod \ p_i$, autrement dit on dispose dans chaque corps $\mathbb{Z}\backslash p_i\mathbb{Z}$ d'un vecteur du noyau $v_i \in \mathbb{Z}_{p_i}^n$.

remarque : en pratique de tels vecteurs peuvent être obtenus par l'algorithme de Wiedemann, qui est une généralisation de l'algorithme de Coppersmith.

La certification consiste alors à tirer aléatoirement un couple (p_i, v_i) et vérifier que $Av_i = 0 \ mod \ p$ (complexité temporelle en $O(n^2)$). On se pose de nouveau la question de savoir combien de p_i sont susceptibles de poser problème, c'est-à-dire qui renverraient un test vrai alors que l'hypothèse est incorrecte. En effet si A est en réalité inversible, $\det(A) \neq 0$, on a vu qu'il existe au plus M nombres premiers qui peuvent diviser $det(A)$. Pour de tels p la matrice A n'est pas inversible modulo p, la preuve est simple si il existait B telle que $AB = I \ mod \ p$ en passant au déterminant on aurait $det(A)det(B) = 1 \ mod \ p$ soit $0 = 1 \ mod \ p$ car $det(A) = 0 \ mod \ p$ (p divise $det(A)$), absurde. Donc puisque A n'est pas inversible modulo p il existe bien un $v_i \in \mathbb{Z}_{p_i}^n$ non nul tel que $Av_i = 0 \ mod \ p$ (test vrai), tandis que $Av_i \neq 0$ dans $\mathbb{Z}$ (car A inversible donc de noyau réduit au vecteur nul). Par conséquent pour de tels p on peut très bien se faire berner par la certification. Mais comme il y en a au plus M, et qu'on tire p parmi $2M$ nombres premiers on a bien une probabilité max de $1/2$ de se tromper.

Reste à vérifier la complexité spatiale, comme $|(p_i, v_i)| = (n+1)|p_i|$ et qu'on dispose de $2M = 2n(log(n)/2 + log\|A\|)$ tels couples, ils occupent en mémoire $2n(n+1)(log(n)/2 + log\|A\|)|p_i|$ bits, donc on a de nouveau besoin d'une longueur de $p_i \sim log(n)^{1+o(1)}$ pour avoir une complexité spatiale en $n^{2+o(1)}(log\|A\|)^{1+o(1)}$. Comme il a été démontré que le k-ième nombre premier est $\leqslant k(log_e(k) + loglog_e(k) - 1/2)$ pour $k \geqslant 20$, si on choisit les $2M$ premiers à partir de 21, ceux-ci auront bien une taille en bits essentiellement linéaire*.

3.3 Certification de la consistance

Le certificat utilisé est un vecteur d'entiers x et un entier δ tels que $Ax = \delta b$, avec les composantes x_i et δ bornés par $n^{n/2}\|(A, b)\|^n$. Encore une fois on

dispose d'une solution (x, δ) et on souhaite vérifier son exactitude, auquel cas le système est bien consistant.

Vérification : zéro équivalence de $Ax - \delta b$ modulo un p_i aléatoire parmi un ensemble que nous allons préciser. La complexité temporelle est donc toujours la même.

Comme précédemment nous donnons un moyen d'obtenir un tel certificat, et nous précisons sa complexité spatiale :

• **Cas où A est carrée et non singulière.**

Une solution est alors simplement donnée par la règle de Cramer, les composantes de x_i sont des mineurs de (A, b) et $\delta = det(A)$.

Comme les $n+1$ valeurs x_i et δ sont des mineurs, ils sont effectivement bornés par la constante d'Hadamard $n^{n/2}\|(A, b)\|^n$ et donc occupent en mémoire chacun $n(log(n)/2+log\|(A, b)\|)$ bits, comme il y en a $n+1$ la complexité spatiale est bien en $n^{2+o(1)}(log\|[A, b]\|)^{1+o(1)}$.

• **Cas où A est rectangulaire de rang r.**

On se souvient que toute matrice rectangulaire $n \times p$ de rang r est équivalente à la matrice J_r de taille $n \times p$ définie par :

$$J_r = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}$$

Ainsi il existe des matrices inversibles P et Q telles que $A = PJ_rQ$.

Par ailleurs en permutant lignes et colonnes on peut mettre A sous la forme

$$A = \begin{pmatrix} B & C \\ D & E \end{pmatrix}$$

où B de taille $r \times r$ est inversible (mineur principal).

En écrivant les matrices de passages en blocs $P = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}$ et $Q = \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix}$ on obtient :

$$\begin{pmatrix} B & C \\ D & E \end{pmatrix} = A = PJ_rQ = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix} \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix} = \begin{pmatrix} P_1Q_1 & P_1Q_2 \\ P_3Q_1 & P_3Q_2 \end{pmatrix}$$

En identifiant il vient :

$$\begin{cases} B = P_1Q_1 \\ C = P_1Q_2 \\ D = P_3Q_1 \\ E = P_3Q_2 \end{cases}$$

D'où $E = P_3Q_2 = (P_3Q_1)(Q_1^{-1}P_1^{-1})(P_1Q_2) = DB^{-1}C$. Ainsi sans perte de généralité on peut écrire :

$$A = \begin{pmatrix} B & C \\ D & DB^{-1}C \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Supposons qu'il existe une solution $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, elle vérifie alors :

$$Ax = \delta b \Leftrightarrow \begin{pmatrix} B & C \\ D & DB^{-1}C \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \delta \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \Leftrightarrow \begin{cases} Bx_1 + Cx_2 = \delta b_1 \\ Dx_1 + DB^{-1}x_2 = \delta b_2 \end{cases}$$

En remplaçant Cx_2 dans la deuxième équation on obtient la condition suivante :

$$Dx_1 + DB^{-1}(\delta b_1 - Bx_1) = \delta b_2 \Leftrightarrow DB^{-1}b_1 = b_2$$

Voyons comment obtenir une telle solution $x = (x_1, x_2)^t$. Si x_1 est solution de $Bx_1 = \delta b_1$ (une telle solution existe car B inversible et est donnée par la règle de Cramer, on se ramène au cas précédent), alors il est clair que $x = (x_1, 0, ..., 0)^t$ est solution de $Ax = \delta b$ (prendre $x_2 = 0$ dans le système d'équation ci-dessus). Et la complexité spatiale ne change alors pas par rapport au cas précédent.

Reste encore une fois à estimer la probabilité de se tromper. La question à se poser est donc toujours la même : combien de nombres premiers p_i peuvent certifier que $Ax = \delta b$ alors que (x, δ) n'est pas une solution correcte ?

Soit $(\hat{x}, \hat{\delta})$ la vraie solution, il y a falsification si $x - \hat{x} \equiv 0 \ mod \ p_i$ (n composantes) et $\delta - \hat{\delta} \equiv 0 \ mod \ p_i$. C'est-à-dire pour les p_i qui divisent les $n+1$ différences. Comme les composantes sont écrites sur M bits (car bornées par $n^{n/2}\|(A,b)\|^n$), les différentes sont écrites sur $k = M + 1$ bits. Donc il y a au plus $k = 1 + n(log(n)/2 + log\|(A,b)\|)$ p_i pathologiques.

En choisissant p_i parmi $3k+3$ premiers, on a donc une chance sur 3 de tirer un mauvais p_i. Enfin on effectue deux fois le test de zéro équivalence de $Ax - \delta b$ modulo le p_i tiré, qui a donc une probabilité au pire $1/2 \times 1/2 = 1/4$ d'échouer. Par la formule des probabilités totales on a donc une probabilité d'erreur de $1/3 + 2/3 \times 1/4 = 1/2$, ce que l'on voulait.

3.4 Certification de l'inconsistence

Certificat : $2n(log(n)/2+log\|(A,b)\|)$ nombres premiers p_i et vecteurs v_i tels que $v_i^T A = 0 \ mod \ p_i$ et $v_i^T b \neq 0 \ mod \ p_i$ contredisant $v_i^T Ax = v_i^T b$ sur les entiers.

Vérification : tirer aléatoirement (p_i, v_i) et tester les 2 conditions.

Afin de comprendre le pourquoi du comment de ce certificat il a été nécessaire de consulter l'article de la référence [3] : *Certifying inconsistency of sparse linear systems*. Comme toujours les vérifications sont faites dans un corps, on dispose du lemme suivant :

- **Inconsistence sur un corps** K ($A \in K^{n\times n}$,$b \in K^{n\times 1}$)

Lemme. *Il n'existe pas de* $x \in K^{n\times 1}$ *tel que* $Ax = b$ *si et seulement si il existe* $u \in K^{1\times n}$ *tel que* $uA = (0, ..., 0) \in K^{1\times n}$ *et* $ub \neq 0$.

Preuve :

$\boxed{\Rightarrow}$ Ecrire que $Ax = b$ est équivalent à $< L_i, x > = b_i$ soit encore b est une combinaison linéaire des lignes de A. Donc a contrario si le système est inconsistant dans ce corps, i.e il n'existe pas de $x \in K^{n\times 1}$ tel que $Ax = b$ alors b n'est pas combinaison linéaire des lignes, donc la matrice (A, b) est de rang : $rang(A|b) = rang(A) + 1$. Le théorème du rang nous donne alors

$$\begin{array}{rcl} dim(Ker(A|b)) = dim(KerA) - 1 & \Rightarrow & \exists u \in K^{1\times n} \in Ker(A)\backslash Ker(A|b) \\ & \Rightarrow & uA = 0, ub \neq 0 \end{array}$$

$\boxed{\Leftarrow}$ Il existe $u \in K^{1\times n}$ tel que $uA = (0, ..., 0) \in K^{1\times n}$ et $ub \neq 0$. Raisonnons par l'absurde, si on avait $Ax = b$ alors en multipliant à gauche par u on aurait $0 = uAx = ub \neq 0$, contradiction.

On comprend alors mieux l'utilisation de ce certificat pour vérifier l'inconsistence modulo p_i. Pour déterminer un tel certificat (les (p_i, v_i)) il suffit de choisir aléatoirement un v_i dans le noyau-gauche (noyau de la transposée) de A modulo p_i par l'algorithme de Wiedemann, on a alors une grande probabilité d'avoir $ub \neq 0$.

Par ailleurs, il se peut qu'un système consistant dans $\mathbb{Z}$ apparaissent inconsistant dans $\mathbb{Z}\backslash p_i\mathbb{Z}$. En effet dans la preuve ci-dessus on a montré au passage

que le système est inconsistant si et seulement si le rang de (A, b) est strictement supérieur au rang de A. Il y a donc un problème si malencontreusement le rang r de A devient plus petit modulo p, ce qui se produit quand p divise $det(B)$ où B est un mineur principal de taille $r \times r$. Ainsi B n'est plus inversible modulo p et le rang est $< r$. Comme B est un mineur, l'inégalité d'Hadamard nous assure de nouveau qu'il y a au plus M nombres premiers qui divisent $det(B)$, donc on va choisir aléatoirement p_i parmi les $3M$ premiers nombres premiers de façon à avoir une probabilité $1/3$ de se fourvoyer et qu'ils ne soient pas trop grand pour vérifier la complexité spatiale et temporelle.

remarque : dans l'article sus-cité, ils vont plus loin, soit en fournissant une solution au système via l'algorithme de Giesbrecht (en 1997), soit en fournissant un certificat d'inconsistance sur $\mathbb{Z}$. Leur algorithme de construction travaille dans les extensions modulo p^e, et les preuves sont effectuées sur la forme normale de Smith, tout comme nous l'avons abordé en cours.

4 Certifications basées sur la LU decomposition

Nous savons que la décomposition LU est largement utilisée dans la résolution de système linéaire, et permet également de calculer le déterminant ou le rang d'une matrice. Nous allons voir qu'elle va aussi permettre de certifier un déterminant ou un rang. Comme les preuves qui vont suivres sont similaires à toutes celles présentées jusqu'ici, elles seront un peu plus concises.

Définition 1. A $(m \times n)$ possède une LU decomposition de rang r si $A = LU$ avec L $(m \times r)$ matrice triangulaire inférieure unitaire et U $(r \times n)$ matrice triangulaire supérieure sans 0 sur la diagonale.

Définition 2. Soit A de taille $m \times n$, un système de LU résidus de rang r et de longueur k est une suite de k triplets distincts $(p_1, L_1, U_1), ..., (p_k, L_k, U_k)$ où les nombres premiers p_i sont strictement croissants, les entrées de L_i, U_i sont normalisés modulo p_i et $A = L_i U_i \ mod \ p_i$ decomposition de rang r.

Lemme. Soit A de rang r et un système de LU résidus de rang s et de taille k, posons $h = n(log(n)/2 + log\|A\|)$ bornant la taille en bits de tout mineur de A. Alors $s \leqslant r$, et si $s < r$ on a $k \leqslant h$.

Preuve : $s \leqslant r$ toujours vérifié, le rang dans $\mathbb{Z}$ est plus grand ou égal au rang réduit. A possède un rang s modulo p_i strictement inférieur à son rang r dans $\mathbb{Z}$ que si un mineur $r \times r$ est divisible par p. Le nombre maximal de tels p est h, donc la longueur maximale du système de LU résidus est h.

remarque : On a vu en cours que A possède une décomposition LU de rang r si et seulement si A est de rang r et a un profil de rang générique. Sinon, il est toujours possible de trouver des matrices de permutations P et Q de sorte que PAQ ait un profil de rang générique.

Théorème. *Soit $A \in \mathbb{Z}^{n\times n}$, $h = n(log(n)/2 + log\|A\|)$. Il existe un système de LU résidus général de longueur $3h$ et dont les p_i ont une taille en bits $(logh)^{1+o(1)}$ qui certifie $rang(A)$. Le certificat occupe $n^{3+o(1)}(log\|A\|)^{1+o(1)}$ en espace et $n^{2+o(1)}(log\|A\|)^{1+o(1)}$ en temps.*

Le certificat est bien sur le système de LU résidus.

Validation : Tirer aléatoirement (p, L, U) et valider la zéro équivalence $PAQ = LU \ mod \ p$ (probabilité $1/p$ de se tromper). Le nombre de p à l'origine d'une falsification est au plus h, on choisit p parmi $3h$ nombres premiers donc probabilité $1/3$ soit au total une probabilité d'un mauvais certificat $1/3 + 2/3 \times 1/p \leqslant 1/2$ pour $p > 5$.

Théorème. Soit $A \in \mathbb{Z}^{n\times n}$, $h = n(log(n)/2 + log\|A\|)$. Il existe un système de LU résidus général de longueur $3h + 3$ et dont les p_i ont une taille en bits $(logh)^{1+o(1)}$ qui certifie $det(A)$.

Validation : Si le rang du système LU est inférieur à n, le prétendu déterminant doit être égal à 0, et la validation consiste à valider le rang par le théorème précédent. Tirer aléatoirement (p, L, U) et valider la zéro équivalence $PAQ = LU \ mod \ p$ (probabilité $1/p$ de se tromper) et vérifier que $d = \prod_{i=1}^{n} U_{i,i}$. Le nombre de p à l'origine d'une falsification est au plus $h + 1$ (diviseurs de $d - det(A)$), on choisit p parmi $3h + 3$ nombres premiers donc probabilité $1/3$ soit au total une probabilité d'erreur de $1/3+2/3\times1/p \leqslant 1/2$ pour $p > 5$.

5 Certification basés sur la similarité

Définition. *Une matrice carrée A est sous forme normale de Frobenius si elle est la somme directe de matrices de companions de polynômes unitaires*

$f_1(x), ..., f_k(x)$ *tels que* $f_i | f_{i+1}$ *pour tout* $1 \leqslant i \leqslant k-1$.

Exemple : $A_1 = \begin{pmatrix} 0 & -2 \\ 1 & 0 \end{pmatrix}$ est la matrice compagnon du polynôme unitaire x^2+2, la somme directe suivante donne lieu à une forme normale de Frobenius

$$M = A_1 \oplus A_2 \oplus A_3 = \begin{pmatrix} 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Propriété. *Toute matrice carrée sur un corps est semblable à une unique forme normale de Frobenius.*

Dans la section précédente notre certificat était une suite de décomposition LU sur des réductions croissantes modulo p, qui nous servait à certifier le rang et le déterminant. De la même manière nous allons nous appuyer ici sur une suite de décomposition de similarité sur des réductions croissantes modulo p dans le but de certifier le polynôme caractéristique. Nous définissons cette structure ci-dessous :

Définition. *Un système résiduel de similarité pour* $A, B \in \mathbb{Z}^{n \times n}$ *de taille* k *est une suite de quadruplet* $(p, S, T, \bar{B})$ *avec* p *premiers distincts,* $S, T, \bar{B} \in \mathbb{Z}_p^{n \times n}$ *et tels que* S *inversible avec* $T \equiv S^{-1}$, $B \equiv \bar{B}$ *et* $A = S\bar{B}T$ *modulo* p. *(où* $\bar{B}$ *est la forme normale de Frobenius semblable).*

Théorème. *Soit* $A \in \mathbb{Z}^{n \times n}$, $h_A = n(1 + log(n)/2 + log\|A\|)$. *Il existe un système résiduel de similarité de longueur* $6h_A + 6$ *qui certifie le polynôme caractéristique* $f(x)$, *occupant* $n^{3+o(1)}(log\|A\|)^{1+o(1)}$ *en espace et* $n^{2+o(1)}(log\|A\|)^{1+o(1)}$ *en temps.*

Preuve : Soit $c^A(x)$ le vrai polynôme caractéristique de A. Le i-ème coefficient de $c^A(x)$ est la somme des $\binom{n}{i}$ mineurs principaux $i \times i$ (résultat obtenu en regardant de plus près le développement du déterminant $det(A - XI_n)$). On sait que chaque mineur est majoré par 2^h par Hadamard, et très grossièrement $\binom{n}{i} \leqslant 2^n$ (un exercice classique est de prouver que $\sum_{i=1}^{n} \binom{n}{i} = 2^n$). Ainsi le i-ième coefficient de $c^A(x)$ est majoré par $\leqslant 2^{n+h}$ soit de longueur en bits $n + h = h_A$. Par conséquent $g(x) = f(x) - c^A(x)$ a des coefficients de taille $k = h_A + 1$, donc est nul pour au plus k premiers et peut générer une

falsification, mais a une probabilité $k/(6h_a + 6) = 1/6$ de se produire.

Quant à la validation propre de $f(x) = c^A(x)$, on procède comme suit :

- Tirer aléatoirement $(p, S, T, \bar{B})$, et vérifier la zéro équivalence de $ST - I$ et $A - S\bar{B}T$ modulo p en $O(n^2)$ avec une probabilité d'erreur de $2/p$.
- Vérifier dans $\bar{B}$ que $f_i | f_{i+1}$ (en $d^\circ(f_{i+1})^{1+o(1)}$), former $f_p(x) = \prod f_i(x)$ modulo p en $O(n^2)$. Par unicité de la forme de Frobenius on doit avoir $c^A(x) \equiv f_p(x)\ mod\ p$.
- Enfin vérifier que $f(x) \equiv f_p(x)\ mod\ p$

A partir de la certification du polynôme caractéristique on peut aisément vérifier la *signature* d'une matrice symétrique définie comme le triplet (n_+, n_0, n_-) indiquant respectivement le nombre de valeurs propres positives, nulles et négatives. En effet, il suffit de commencer par vérifier $c^A(x)$, alors n_0 est donné par le plus grand m tel que x^m divise $c^A(x)$. Quant au nombre de racines positives, il est donné par la règle des signes de Descartes sur les coefficients de $c_A(X)$.

On termine en donnant un certificat pour la forme normale de Frobenius :

Théorème. *Soit $A, G \in \mathbb{Z}^{n\times n}$, G sous forme de Frobenius avec $\|G\| \leqslant 2^n e^{n/2} n^{n/2}$. Si les formes de Frobenius de A réduite modulo p_i sont égales à G réduite modulo p_i pour des entiers premiers distincts $p_1, ..., p_t$ avec $\prod_{i=1}^t p_i \geqslant 8^n e^n n^{2n} \|A\|^{3n}$ alors G est la forme de Frobenius de A.*

Dit autrement si on a la similarité sur suffisamment de corps $\mathbb{Z}\backslash p_i\mathbb{Z}$ alors on l'a également sur $\mathbb{Z}$. La démonstration est donnée plus en détail dans l'article [4] *Giesbrecht, M., and Storjohann, A. Computing rational forms of integer matrices.*, l'algorithme proposé permet de déterminer rapidement la forme normale de Frobenius en déterminant donc la solution modulo un ensemble de nombres premiers (vérifiant la propriété ci-dessus), et reconstruisant la solution générale via les restes chinois.

remarque : la définition de la forme normale de Frobenius m'a tout de suite fait penser, de par les divisibilités en cascades, à la forme normale de Smith abordée en cours. Vue la similarité (sans mauvais jeux de mots) entre ces deux définitions, il ne m'aurait pas étonné d'y trouver un lien sous-jacent. C'est dans la thèse de Clément Pernet et le mémoire de de Gérard Villard

que j'ai trouvé la réponse à ma question.

Théorème. *La forme normale de Frobenius de A est $diag(C_{f_1}, ..., C_{f_\varphi})$ si et seulement si la forme normale de Smith de la matrice $XI_n - A$ est $diag(f_1, ..., f_\varphi, 1, ..., 1)$.*

6 Conclusion

Dans cet article les auteurs ont proposé différents certificats permettant de valider l'exactitude d'un résultat avec une probabilité d'erreur max de 1/2. Nous avons vu que la probabilité d'erreur de vérification provenait du fait que l'on travaillait modulo des nombres premiers p, qui lorsqu'ils divisaient certaines quantités (typiquement un déterminant), engendraient de mauvaises certifications. L'exécution des algorithmes de vérification modulo p (ou même directement de calculs des quantités classiques d'algèbre linéaire : rang, déterminant, polynôme caractéristique, etc) est toutefois bénéfique car les calculs dans ces petits corps finis sont rapides (ici en temps quasi quadratique grâce à l'adjonction de certificats) et la taille des calculs intermédiaires est fixe, tandis que dans $\mathbb{Z}$ ils peuvent facilement exploser. De plus dans l'optique d'une certification on s'assure que la taille en mémoire de ces certificats n'excède pas $O(n^{3+\epsilon}(log\|A\|)^{1+\epsilon}$ bits en mémoire.

D'autres types de calculs ou de validations probabilistiques existent, par exemple les algorithmes de type Las Vegas, qui contrairement à ceux de Monte Carlo, fournissent toujours une solution correcte mais avec une rapidité aléatoire. Citons par exemple les travaux de Storjohann sur le rang et le déterminant, et ceux de Giesbrecht sur la forme normale de Frobenius. La complexité $n^{\eta+o(1)}$ de tous ces algorithmes dépend évidemment de la complexité de la brique de base de l'algèbre linéaire dense, à savoir le produit matriciel, donc de l'exposant $\omega \in [2, 3]$. La borne inférieure 2 est considérée comme optimale. A ce jour le meilleur algorithme connu (asymptotiquement parlant) est celui de Coppersmith-Winograd qui possède une complexité en $O(n^{2,376})$, mais qui n'est pas implémentable car la constante dans le grand O est prohibitive. C'est pourquoi l'amélioration du produit matriciel est encore aujourd'hui l'objet de recherches actives.