Panoramas & Synthèses
36, 2012, p. 19–49

ALGORITHMES DE COMPTAGE DE POINTS D'UNE COURBE DÉFINIE SUR UN CORPS FINI

par

Pierrick Gaudry

Résumé. **—** Le calcul de la fonction Zêta d'une courbe algébrique définie sur un corps fini, communément appelé comptage de points, est une tâche algorithmique dont l'étude a été poussée par d'importantes applications cryptographiques. Dans cet article de survol, nous donnons un aperçu des différentes méthodes disponibles pour s'attaquer à ce problème. Dans la littérature, celles-ci sont traditionnellement illustrées par des calculs records que nous mentionnerons afin de bien mettre en perspective les implications pratiques.

Abstract **(Counting algorithms of points of a curve defined over a finite field).** **—** The computation of the zeta function of an algebraic curve defined over a finite field, commonly known as point counting, is a task whose computational study was driven by significant cryptographic applications. In this survey article, we give an overview of the various available methods to address this problem. In the literature, these are traditionally illustrated by record computations that we mention to properly put into perspective the practical implications.

1. Introduction : énoncé du problème

Soit $q = p^n$ un puissance d'un nombre premier p et soit $\mathbb{F}_q$ le corps fini à q éléments. Soit $\mathcal{C}$ une courbe algébrique projective lisse de genre g définie sur le corps $\mathbb{F}_q$. La fonction Zêta de $\mathcal{C}$ est définie par

$$Z(t) = \exp\left(\sum_{k\geqslant 1} N_k \frac{t^k}{k}\right),$$

où N_k est le nombre de points de $\mathcal{C}$ définis sur $\mathbb{F}_{q^k}$, que l'on notera aussi $\#\mathcal{C}(\mathbb{F}_{q^k})$. Le théorème de Weil (dont on peut trouver une preuve dans [**65**] par exemple) affirme que $Z(t)$ est en fait une fraction rationnelle, ce qui signifie entre autres que connaître les premiers termes de la série suffit pour calculer tous les autres termes. Plus précisément, $Z(t)$ s'écrit

$$Z(t) = \frac{L(t)}{(1-t)(1-qt)},$$

Classification mathématique par sujets **(2000).** **—** 11G20, 11Y99.
Mots clefs. **—** Courbes algébriques, corps finis, comptage de points.

où $L(t) = a_0 + a_1 t + \cdots + a_{2g} t^{2g}$ est un polynôme de degré $2g$ dont les coefficients sont des entiers vérifiant

$$a_0 = 1, a_{2g} = q^g, \text{ et } a_{2g-i} = q^{g-i} a_i, \text{ pour } 0 \leqslant i \leqslant g.$$

De plus, les inverses des racines de $L(t)$ sont de module $\sqrt{q}$ (c'est l'équivalent de l'hypothèse de Riemann pour les corps de fonctions), ce qui donne la borne $|a_i| \leqslant \binom{2g}{i} q^{i/2}$, pour les coefficients de $L(t)$.

La jacobienne de la courbe $\mathcal{C}$ que l'on notera $\mathrm{Jac}(\mathcal{C})$ est une variété abélienne dont les propriétés arithmétiques sont liées à celle de $\mathcal{C}$. En particulier, le numérateur $L(t)$ de $Z(t)$ est le polynôme réciproque du polynôme caractéristique $\chi_\pi(t)$ de l'endomorphisme de Frobenius $\pi : x \mapsto x^q$ agissant sur $\mathrm{Jac}(\mathcal{C})$. Par ailleurs, le nombre de points $\mathbb{F}_q$ rationnels de $\mathrm{Jac}(\mathcal{C})$ est donné par $\chi_\pi(1)$, si bien que l'ordre du groupe des points rationnels de $\mathrm{Jac}(\mathcal{C})$ se déduit de $Z(t)$. La réciproque est vraie dans le cas du genre 1 ou 2, mais la connaissance de $\chi_\pi(1)$ ne suffit plus à déterminer facilement $Z(t)$ lorsque g est supérieur ou égal à 3.

Nous nous intéressons ici au problème de calculer $Z(t)$ ou éventuellement seulement $\chi_\pi(1)$. Ce type de calcul intervient pour les applications suivantes : en théorie des codes correcteurs d'erreurs, pour un genre fixé, le nombre de points sur une courbe indique la qualité du code que l'on peut créer à partir de celle-ci ; en cryptographie, connaître l'ordre du groupe des points de la jacobienne est nécessaire car si celui-ci est friable, le problème du logarithme discret est facile, et donc aucune sécurité n'est à espérer. Et bien sûr, si l'on est capable de calculer efficacement les invariants associés à une courbe, cela ne fait qu'enrichir les logiciels de calcul formel qui sont devenus des calculettes indispensables à bon nombre de mathématiciens.

Il existe des algorithmes qui permettent de calculer $Z(t)$: puisque tous les objets considérés sont finis, il suffit d'énumérer. Évidemment cela ne suffit pas pour traiter des exemples de grande taille. Dans nos mesures de complexité des algorithmes, la taille que nous prendrons comme référence sera le produit $ng \log p$; en effet c'est approximativement le logarithme de l'ordre du groupe considéré. En ce sens nous sortons du paradigme classique où le temps de calcul doit être évalué en fonction de la taille de l'objet donné en entrée, mais nous contournons ainsi une discussion sur le modèle de la courbe $\mathcal{C}$. Cela dit, pour certains algorithmes le degré de l'équation qui est fournie pour la courbe est une donnée plus cruciale que son genre.

L'objectif ultime est l'obtention d'un algorithme dont le temps de calcul soit borné par un polynôme en $ng \log p$. Actuellement des résultats partiels vont dans cette direction, mais l'objectif n'est pas encore atteint si g et $\log p$ tendent tous les deux vers l'infini. Dans cet exposé nous tentons de donner un aperçu de la situation d'aujourd'hui.

2. Survol des algorithmes disponibles

2.1. Les algorithmes ℓ-adiques. — L'algorithme publié par Schoof en 1985 [**60**] pour compter le nombre de points d'une courbe elliptique sur un corps fini fut le premier algorithme a atteindre une complexité polynomiale. Il fut ensuite étendu aux variétés abéliennes par Pila [**55**], puis par Adleman et Huang [**1**] ainsi que Huang et Ierardi [**34**].

La complexité est polynomiale en $\log q = n \log p$ pour toute famille de courbes. Cette notion de famille, donnée par Pila, ne suffit pas pour conclure en toute généralité sur une complexité polynomiale en $n \log p$ uniformément pour toutes les courbes de genre fixé. Par exemple, ce résultat couvre toutes les courbes hyperelliptiques de genre fixé.

On peut noter que l'algorithme original de Schoof est déterministe. Les extensions ont parfois été conçues en insistant pour conserver cette propriété, au prix de complications importantes. Dans notre exposé nous omettrons souvent de mentionner le caractère probabiliste ou déterministe des algorithmes, même si les avantages d'un algorithme déterministe ne sont pas à négliger.

2.2. Les algorithmes sous-exponentiels. — Cette classe d'algorithmes a été initiée en 1994 par Adleman, DeMarrais et Huang [**2**] dans le cadre des courbes hyperelliptiques. Il s'agissait à l'origine d'une première étape dans un calcul de logarithmes discrets dans la jacobienne d'une courbe hyperelliptique. La complexité, bien qu'heuristique, était sous-exponentielle, sous la condition que le genre croisse suffisamment vite par rapport à $n \log p$.

L'algorithme initial a été étendu, amélioré, prouvé dans certains cas par diverses personnes [**53, 20, 21, 16**] et finalement une version a été prouvée dans un cadre très général par Heß [**33**]. Ce type d'algorithme ne fournit pas toute la fonction Zêta, mais seulement $\chi_\pi(1)$ et les diviseurs élémentaires du groupe des points de la jacobienne.

Toutefois, tous ces algorithmes requièrent que le corps fini et donc sa caractéristique ne soient pas trop grands par rapport au genre de la courbe. Ainsi on entre dans le champ d'applications des algorithmes p-adiques, et désormais ces algorithmes sous-exponentiels ne sont plus les plus rapides (du moins asymptotiquement). Toutefois, si l'on est intéressé par la structure du groupe ou par des calculs de logarithme discret, cette approche est la bonne.

2.3. Les algorithmes p-adiques utilisant le relèvement canonique

En 1999, Satoh [**58**] a inventé une nouvelle méthode pour compter les points d'une courbe elliptique. S'appuyant sur le relèvement p-adique canonique de la courbe, cette méthode fonctionne lorsque la caractéristique du corps de base est petite. À p fixé, la complexité est meilleure que celle de l'algorithme de Schoof.

De nombreuses améliorations et généralisations ont eu lieu. Finalement, on dispose d'un algorithme en temps polynomial en n pour calculer la fonction Zêta d'une courbe hyperelliptique de genre fixé sur $\mathbb{F}_{2^n}$. Cela ne change rien du point de vue théorique : ces cas étaient déjà couverts par l'algorithme de Pila. Toutefois, l'algorithme est cette fois-ci très pratique.

2.4. Les algorithmes p-adiques utilisant la cohomologie de Monsky-Washnitzer. — Peu de temps après la publication de l'algorithme de Satoh, Kedlaya [**38**] a proposé une autre méthode de comptage de points utilisant un relèvement p-adique, mais cette fois-ci, le relèvement est quelconque et la cohomologie de Monsky-Washnitzer est l'outil qui permet de calculer la fonction Zêta.

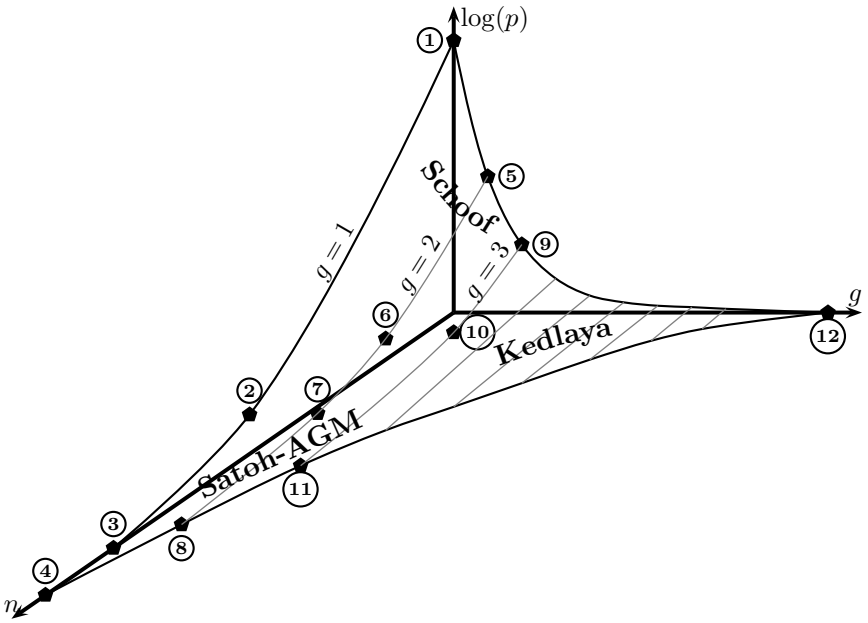

FIGURE 1. Records de calcul de fonctions Zêta de courbes. Afin de rendre le diagramme lisible, celui-ci est uniquement figuratif : les coordonnées des points de records ne sont pas les vraies coordonnées. Les pointes sur les axes devraient être beaucoup plus pointues, en particulier pour celle le long de l'axe des n.

Cet algorithme a une bonne complexité à caractéristique p fixée. Décrit tout d'abord dans le cas des courbes hyperelliptiques sur un corps de caractéristique impaire, il a été étendu en plusieurs étapes à des classes de courbes de plus en plus larges. Finalement, pour les courbes C_{ab}, sur un corps de caractéristique p fixée, la complexité est polynomiale en ng.

2.5. Les algorithmes p-adiques utilisant une approche à la Dwork

Toujours au début des années 2000, Lauder et Wan [**45**] ont proposé une autre approche p-adique au problème de comptage de points : leur algorithme suit la preuve de Dwork de la rationalité de la fonction Zêta. L'avantage de leur méthode est qu'elle est extrêmement générale : elle permet de compter le nombre de solutions d'une équation polynomiale sur un corps fini, avec une complexité qui dépend de manière polynomiale en la taille du corps fini et en le degré du polynôme, et exponentielle en le nombre de variables et en la caractéristique du corps.

Telle quelle leur méthode a une complexité moins bonne que l'algorithme de Kedlaya dans les cas où ce dernier s'applique. Des améliorations ont été effectuées pour des classes particulières de courbes [**41, 43, 44**], notamment en utilisant la cohomologie de Dwork, ce qui a permis de retrouver la même complexité que l'algorithme de Kedlaya pour les courbes hyperelliptiques.

N°	p	n	g (type)	Réf.	Algorithme	Commentaire
①	$\approx 10^{2500}$	1	1	[22]	Schoof	
②	5	569	1	[58]	Satoh	Première implantation d'un algorithme p-adique
③	2	50 021	1	[30]	AGM	
④	2	130 020	1	[30]	AGM	La valeur de n permet l'utilisation de base normale gaussienne
⑤	$\approx 5 \cdot 10^{24}$	1	2	[28]	Schoof	
⑥	$2^{32} - 5$	3	2	[6]	Schoof	Utilisation de l'opérateur de Cartier-Manin
⑦	2	4 001	2	[31]	AGM/Richelot	
⑧	2	32 770	2	[47]	AGM/Borchardt	La valeur de n permet l'utilisation de base normale gaussienne
⑨	$\approx 2,8 \cdot 10^{17}$	1	3 (Picard)	[3, 68]	Ad-Hoc	Utilisation d'un algorithme exponentiel adapté aux courbes de Picard
⑩	251	7	3 (hyperell.)	[27]	Kedlaya	
⑪	2	4 098	3 (hyperell.)	[47]	AGM/Borchardt	
⑪	2	5 002	3 (non-hyperell.)	[57]	AGM/Borchardt	
⑫	2	1	350 (hyperell.)	[66]	Kedlaya	

Légende pour la Figure 1