An algorithm for computing zeta functions of a variety using power elementary symmetric polynomials

David Pigeon

An algorithm for computing zeta functions of a variety using power elementary symmetric polynomials

David Pigeon

13 avril 2024

Keywords : Power elementary symmetric polynomials, Newton's identity, Zeta function of a variety, Algorithm.

Résumé

By exploiting power elementary symmetric polynomials and Newton's identities, we find an algorithm that allows us to compute the zeta function of certain varieties that are extensions of a variety where the zeta function is already known.

Table des matières

1 Power Elementary Symmetric Polynomials

Throughout this section, we fix two integers r and b greater than 1, a primitive rth root of unity ξ, and a polynomial P of degree b with coefficients in $\mathbb{Z}$ such that $P(0)$ is nonzero.

Through Newton's identities, we will deduce in this section the expression of the polynomial :

$$\prod_{i=0}^{r-1} P(\xi^i T)$$

in terms of the coefficients of P.

1.1 Notations

We will maintain these notations throughout the section.

1.1.1. *We decompose P into*

$$P(T) =: a_0 \prod_{j=1}^{b}(1-\alpha_j T) := \sum_{k=0}^{b} a_k T^k$$

where $(\alpha_j)_{1\leq j\leq b}$ is an element of $\overline{\mathbb{Q}}^b$.

1.1.2. *We have the following sequence of equalities :*

$$\prod_{i=0}^{r-1} P(\xi^i T) = \prod_{i=1}^{r}\prod_{j=1}^{b}(1-\alpha_j^s \xi^i T) = \prod_{j=1}^{b}(1-\alpha_j^{rs}T^r).$$

So $\prod_{i=0}^{r-1} P(\xi^i T)$ is an element of $\mathbb{Z}[T^r]$.

We then define the polynomial Q of degree b with integer coefficients by

$$Q(T^r) := \prod_{i=0}^{r-1} P(\xi^i T)$$

We decompose Q into

$$Q(T) =: a_0^r \prod_{j=1}^{b}(1-\beta_j T) := \sum_{k=0}^{b} b_k T^k$$

where $(\beta_j)_{1\leq j\leq b}$ is an element of $\overline{\mathbb{Q}}^b$.

1.2 Newton's Identities

1.2.1. *We recall the definitions of power elementary symmetric polynomials and Newton sums.*

*(1) Let k and s be two natural numbers. The **kth power sth elementary symmetric polynomial** is defined as*

$$\sigma_{k,s} := \begin{cases} \sum_{1\leq i_1<\cdots<i_k\leq b} X_{i_1}^s \cdots X_{i_k}^s & \text{if } 0<k\leq b\ ; \\ 1 & \text{if } k=0\ ; \\ 0 & \text{otherwise} \end{cases}$$

*(2) Let s be a natural number. The **sth power Newton sum** is the polynomial $\sigma_{1,s}$, simply denoted S_s.*

*These polynomials are related to each other by the **Newton's identities** (Girard, 1629) :*

$$(-1)^k k\sigma_{k,s} + \sum_{i=0}^{k-1}(-1)^i \sigma_{i,s} S_{s(k-i)} = 0 \quad (\forall k\geq 1, \forall s\geq 0). \tag{1}$$

Proposition 1.2.2. *Let k be an integer greater than 1. We denote $a_k := 0$ and $b_k := 0$ if k is strictly greater than b.*

We have the following two formulas :

(i)

$$\sum_{i=0}^{k-1} a_i S_{k-i}(\underline{\alpha}) + k a_k = 0;$$

(ii)

$$\sum_{i=0}^{k-1} b_i S_{r(k-i)}(\underline{\alpha}) + kb_k = 0.$$

where $\underline{\alpha} := (\alpha_1, \ldots, \alpha_b)$.

Démonstration. The relation (i) is deduced from Newton's identities (1) by taking s equal to 1 and from the relations

$$a_k = (-1)^k a_0 \sigma_{k,1}(\underline{\alpha}) \quad (\forall 0 \leq k \leq b).$$

The relation (ii) is deduced from Newton's identities (1) by taking s equal to r and from the relations

$$b_k = (-1)^k a_0^r \sigma_{k,r}(\underline{\alpha}) \quad (\forall 0 \leq k \leq b).$$

□

Notation 1.2.3. *For any element* $\underline{u} := (u_1, \ldots, u_b)$ *in* $\mathbb{N}^b$ *and any element* $\underline{a}$ *in* $\mathbb{C}^b$*, we denote*

$$\underline{u}! := u_1! \cdots u_b! \quad , \quad |\underline{u}|_0 := \sum_{i=1}^{b} u_i,$$

$$|\underline{u}|_1 := \sum_{i=1}^{b} i u_i \text{ and } \underline{a}^{\underline{u}} := a_1^{u_1} \cdots a_b^{u_b}.$$

We also set

$$\sum_{|\underline{u}|_1 = k} := \sum_{\substack{\underline{u} := (u_1, \ldots, u_b) \in \mathbb{N}^b \\ |\underline{u}|_1 = k}}.$$

Proposition 1.2.4. *We have the equality*

$$Q(T) = a_0^r \left(\sum_{k=0}^{b} \left(\sum_{|\underline{u}|_1 = k} \frac{(-1)^{|\underline{u}|_0}}{\underline{u}!} \prod_{j=1}^{b} \left(r \sum_{|\underline{v}|_1 = rj} \left(-\frac{1}{a_0} \right)^{|\underline{v}|_0} \frac{(|\underline{v}|_0 - 1)!}{\underline{v}!} \underline{a}^{\underline{v}} \right)^{u_j} \right) T^k \right).$$

Démonstration. We have the following lemma :

Lemme 1.2.5. *We have the following equalities*

$$\frac{S_s(\underline{\alpha})}{s} = \sum_{|\underline{v}|_1 = s} \left(-\frac{1}{a_0} \right)^{|\underline{v}|_0} \frac{(|\underline{v}|_0 - 1)!}{\underline{v}!} \underline{a}^{\underline{v}} \quad (\forall s \geq 1).$$

Démonstration. We have the equality

$$-\log\left(\sum_{i=0}^{b} \frac{a_i}{a_0} T^i \right) = \sum_{s \geq 1} \frac{S_s(\underline{\alpha})}{s} T^s.$$

Using the multinomial formula, we have the sequence of equalities

$$\begin{aligned} \sum_{s \geq 1} \frac{S_s(\underline{\alpha})}{s} T^s &= -\log\left(1 + \sum_{i=1}^{b} \frac{a_i}{a_0} T^i \right) \\ &= \sum_{k \geq 1} \frac{\left(\sum_{i=1}^{b} -\frac{a_i}{a_0} T^i \right)^k}{k} \end{aligned}$$

$$= \sum_{s\geq 1}\left(\sum_{|\underline{v}|_1=s} \frac{1}{a_0^{|\underline{v}|_0}|\underline{v}|_0}\frac{|\underline{v}|_0!}{\underline{v}!}(-\underline{a})^{\underline{v}}\right)T^s.$$

□

□

1.3 A Example

We provide an example explaining the results of the previous section.

Let P be the polynomial $1-19T^2+30T^3$. Since $P(T)$ is $(1-2T)(1-3T)(1+5T)$, we have

$$\begin{aligned} Q(T) &= (1-2^rT)(1-3^rT)(1-(-5)^rT) \\ &= 1-(2^r+3^r+(-5)^r)T+(6^r+(-10)^r+(-15)^r)T^2-(-30)^rT^3. \end{aligned} \tag{2}$$

We will recover the relation (2) in the case where r equals 2 by exploiting Propositions (1.2.2) and (1.2.4).

1.3.1. ***First method :*** *we apply relations (i) and (ii) of Proposition (1.2.2). We denote*

$$s_l := S_l(2,3,-5) \quad (\forall l \in \mathbb{N}).$$

By (i), we have

- $s_1 = -a_1 = 0$ *;*
- $s_2 = -2a_2 - a_1s_1 = 38$ *;*
- $s_3 = -3a_3 - a_1s_2 - a_2s_1 = -90$ *;*
- $s_4 = -a_1s_3 - a_2s_2 - a_3s_1 = 722$ *;*
- $s_5 = -a_1s_4 - a_2s_3 - a_3s_2 = -2850$ *;*
- $s_6 = -a_1s_5 - a_2s_4 - a_3s_3 = 16418$ *;*

By (ii), we have

- *for* $k=0$, $b_0 = a_0^2 = 1$ *;*
- *for* $k=1$,
$$b_1 = -b_0s_2 = -38;$$
- *for* $k=2$,
$$b_2 = -\frac{1}{2}(b_0s_4 + b_1s_2) = 361;$$
- *for* $k=3$,
$$b_3 = -\frac{1}{3}(b_0s_6 + b_1s_4 + b_2s_2) = -900.$$

Hence,

$$Q(T) = 1 - 38T + 361T^2 - 900T^3.$$

Second method : *by applying the explicit formula from Proposition (1.2.4).*

- *for* $k=0$, *we look for solutions of* $|\underline{u}|_1 = 0$, *the only solution is* $(0,0,0)$, *hence* $b_0 = 1$ *;*
- *for* $k=1$, *the solution of* $|\underline{u}|_1 = 1$ *is* $(1,0,0)$, *thus*

$$b_1 = -2\sum_{|\underline{v}|_1=2}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}.$$

The solutions of $|\underline{v}|_1 = 2$ *are* $(2,0,0)$ *and* $(0,1,0)$, *thus*

$$b_1 = -2\times 19 = -38;$$

- *for $k = 2$, the solutions of $|\underline{u}|_1 = 2$ are $(2,0,0)$ and $(0,1,0)$, thus*

$$b_2 = 2\left(\sum_{|\underline{v}|_1=2}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}\right)^2 - 2\sum_{|\underline{v}|_1=4}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}.$$

The solutions of $|\underline{v}|_1 = 4$ are $(4,0,0)$, $(2,1,0)$, $(0,2,0)$, and $(1,0,1)$, thus

$$b_2 = 2\times 19^2 - 2\times\frac{(-19)^2}{2} = 361;$$

- *for $k = 3$, the solutions of $|\underline{u}|_1 = 3$ are $(3,0,0)$, $(1,1,0)$, and $(0,0,1)$, thus*

$$\begin{aligned} b_3 = &-\frac{4}{3}\left(\sum_{|\underline{v}|_1=2}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}\right)^3 \\ &+4\left(\sum_{|\underline{v}|_1=2}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}\right)\left(\sum_{|\underline{v}|_1=4}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}\right) \\ &-2\sum_{|\underline{v}|_1=6}(-1)^{|\underline{v}|_0}\frac{(|\underline{v}|_0-1)!}{\underline{v}!}\underline{a}^{\underline{v}}; \end{aligned}$$

The solutions of $|\underline{v}|_1 = 6$ are $(6,0,0)$, $(4,1,0)$, $(2,2,0)$, $(0,3,0)$, $(3,0,1)$, $(1,1,1)$, and $(0,0,2)$, thus

$$b_3 = \frac{4}{3}\times 19^3 - 4\times 19^2\times\frac{19^2}{2} - 2\times\frac{30^2}{2} - 2\left(-\frac{(-19)^3}{3}\right) = -900.$$

We retrieve that

$$Q(T) = 1 - 38T + 361T^2 - 900T^3.$$

Remarque 1.3.2. *We can also use the resultant :*

$$\mathrm{Res}_X(X^r - 1, P(XT)) = \prod_{i=0}^{r-1} P(\xi^i T).$$

From which we have

$$\begin{aligned} Q(T^2) &= \det\begin{pmatrix} 1 & 0 & 0 & 30T^3 & 0 \\ 0 & 1 & 0 & -19T^2 & 30T^3 \\ -1 & 0 & 1 & 0 & -19T^2 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix} \\ &= 1 - 38T^2 + 361T^4 - 900T^6. \end{aligned}$$

2 Algorithms

We fix a prime number p, a power q of p, and an integer r greater than 1.

If we know the zeta function of a variety X over $\mathbb{F}_q$ (i.e., a $\mathbb{F}_q$-scheme of finite type separated), we will deduce from the results of the previous section an algorithm calculating the zeta function of

$$X_{\mathbb{F}_{q^r}} := X \otimes_{\mathbb{F}_q} \mathbb{F}_{q^r}.$$

2.1 Usual algorithms

We fix an integer N greater than 1. Let $\mathbb{A} := \mathbb{Z}/p^N\mathbb{Z}$.

2.1.1. *The space required to store an element of $\mathbb{A}$ is $O(N)$.*
Using fast multiplication, we can perform divisions and multiplications in $\mathbb{A}$ in $O(N^{1+\epsilon})$.

2.1.2. *Consider polynomials A and B in $\mathbb{A}[T]$ of degrees n and m respectively such that $n \geq m$. Suppose A is monic. We recall an algorithm that allows us to compute the Euclidean division of A by B. The polynomial $T^m B(1/T)$ is invertible modulo T^{n-m+1} because its value at 0 is 1. Let $A := BQ + R$ be the Euclidean division of A by B where R is a polynomial in $\mathbb{A}[T]$ such that $\deg R < m$. We have*

$$\frac{T^n A(1/T)}{T^m B(1/T)} = T^{n-m}Q(1/T) + \frac{T^n R(1/T)}{T^m B(1/T)}.$$

So the polynomial $T^{n-m}Q(1/T)$ (and thus Q by reversing the coefficients) is obtained by letting T tend to infinity since

$$\lim_{T\to\infty} T^n R(1/T)/(T^m B(1/T)) = 0.$$

Thus R is simply obtained by the relation $R = A - BQ \mod T^m$. We deduce the following algorithm :

DivEucl(A, B)

Input : *polynomials A and B in $\mathbb{A}[T]$ of degrees n and m respectively such that $n \geq m$.*
Output : *the remainder of the Euclidean division of A by B.*

- ***Step 1 :*** *compute the polynomials $A_1(T) := T^n A(1/T)$ and $B_1(T) := T^m B(1/T)$.*
- ***Step 2 :*** *compute the polynomial $B_1(T)^{-1} \mod T^{n-m+1}$. If $n - m + 1$ equals 1, return $1 = B_1(0)^{-1}$. Otherwise :*
  - *Step 2.1 : recursively compute the inverse B_2 of the polynomial B_1 modulo $T^{[(n-m+1)/2]}$;*
  - *Step 2.2 : return the polynomial $B_2(T)+(1-B_2(T)B_1(T))B_2(T) \mod T^{n-m+1}$.*
- ***Step 3 :*** *compute the polynomial $A_1(T)/B_1(T) \mod T^{n-m+1}$.*
- ***Step 4 :*** *compute the polynomial $T^{n-m}Q(1/T)$ then the polynomial Q by reversing the coefficients.*
- ***Step 5 :*** *return the polynomial $A - BQ \mod T^m$.*

Steps 1 and 4 simply involve reversing the coefficients, which are done in $n+m$ operations and $n-m$ operations respectively.

Step 2.1 is performed in $O((n-m)^{1+\epsilon})$ operations. If we denote c_{n-m+1} as the complexity of Step 2, we have the relation

$$c_{n-m+1} = c_{[n-m+1]/2} + O(n-m),$$

hence c_{n-m+1} *is*

$$O((n-m)^{1+\epsilon}) = O(n^{1+\epsilon}).$$

Steps 3 and 5 involve polynomial multiplications, which are performed in $O((n-m)^{1+\epsilon}) = O(n^{1+\epsilon})$ *operations and* $O(m^{1+\epsilon})$ *operations respectively. Thus, the number of elementary operations is*

$$O(n^{1+\epsilon}).$$

2.1.3. *We recall an algorithm that allows computing the* s*-th term of a linear recurrent sequence with coefficients in* $\mathbb{A}$ *of order* b*. Here is the algorithm :*

SuiteRec$((e_i)_{0\le i\le b-1}, (f_i)_{0\le i\le b-1}, s)$

Input : *coefficients* $(e_i)_{0\le i\le b-1}$ *of the linear recurrent sequence of order* b *:*

$$x_{n+b} = e_{b-1}x_{n+b-1} + \cdots + e_0 x_n,$$

initial conditions $(f_i)_{0\le i\le b-1}$*, and an integer* s *greater than* 1*.*
Output : *the* s*-th term of the sequence* $(x_i)_i$*.*

Let $R(T) := T^b - e_{b-1}T^{b-1} + \cdots - e_0$*.*

If s *is less than* $b-1$*, return* f_s*. Otherwise :*

- ***Step 1 :*** *recursively compute* $T^s =: g_0 + g_1 T + \cdots + g_{b-1}T^{b-1} \mod R$ *using the recurrence relation*

$$\forall l \in \mathbb{N}^*, \quad T^l = \begin{cases} (T^{l/2})^2 \mod R & \text{if } l \text{ is even} \\ T(T^{(l-1)/2})^2 \mod R & \text{otherwise.} \end{cases}$$

- ***Step 2 :*** *return* $g_0 f_0 + \cdots + g_{b-1} f_{b-1}$*.*

Through the Euclidean division algorithm, multiplication in $\mathbb{A}[T]/(R)$ *has a complexity of* $O(b^{1+\epsilon})$*. The number of recursive calls is* $O(\log s)$*. Thus, the number of elementary operations is*

$$O(b^{1+\epsilon}\log s).$$

2.2 Calcul de Q en fonction de P

On fixe un entier b **supérieur à** 1**, un polynôme** P **à coefficients dans** $\mathbb{Z}$ **de degré** b **tel que** $P(0)$ **soit non nul.**

On garde les notations de la section 1.1.

Définition 2.2.1. *Soit* $R = \sum_{j=0}^{b} c_j T^j$ *un polynôme de* $\mathbb{Z}[X]$*. La* ***norme de Gauss de*** R *est définie par* $\|R\| := \max_{0\le j\le b} |c_j|$*.*

Proposition 2.2.2. *(i) On a*

$$\|Q\| \le \|P\|^r p_{r,b},$$

où $p_{r,b}$ *est le nombre de solutions dans* $\mathbb{N}^r$ *de* $i_1 + \cdots + i_r = b$*.*

(ii) Soit λ un réel supérieur à 1 tel que

$$|\alpha_j| \leq \lambda \quad (\forall 1 \leq j \leq b).$$

Alors on a

$$\|Q\| \leq 2^b |a_0|^r \lambda^{br}.$$

Démonstration. Montrons (i). Pour tout entier k tel que $0 \leq k \leq b$, on a

$$|b_k| \leq \sum_{i_1+\cdots+i_r=k} \|P\|^r \leq \|P\|^r p_{r,k} \leq \|P\|^r p_{r,b}.$$

Montrons (ii). Soit k un entier tel que $0 \leq k \leq b$. Des relations existant entre les coefficients et les racines du polynôme Q, on déduit les inégalités

$$\begin{aligned}\left|\frac{b_k}{a_0^r}\right| &\leq \sum_{1\leq i_1<\ldots<i_k\leq b} |\beta_{i_1}|\cdots|\beta_{i_k}| \leq \sum_{k=0}^{b} \binom{b}{k} \lambda^{rk} \\ &\leq 2^b \lambda^{rb}.\end{aligned}$$

car $|\beta_j| \leq \lambda^r$ pour tout entier j tel que $1 \leq j \leq b$.

□

Proposition 2.2.3. *Notons $\delta := P(0)$. On peut calculer le polynôme Q connaissant P avec une complexité de*

$$O\left(b^{1+\epsilon}(b+\log r)(M_Q + rb\log_p \delta)^{1+\epsilon}\right)$$

où M_Q est un entier naturel tel que $\|Q\| \leq p^{M_Q}$.

Démonstration. Voici toutes les étapes de l'algorithme, on donne leurs validités et leurs complexités.

AlgoRec(P, r, M)

Entrée : un polynôme P de degré b à coefficients entiers tel que $P(0)$ soit non nul, un entier r supérieur à 1 et un entier naturel M_Q.
Sortie : le polynôme Q tel que

$$Q(T^r) = \prod_{i=0}^{r-1} P(\xi^i T).$$

On calcule Q en trois étapes. Les calculs se font dans l'anneau

$$\mathbb{A} := \mathbb{Z}/p^N\mathbb{Z}$$

où $N := [\log_p(b)(M + rb\log_p \delta)] + 1$.
On note $\delta = P(0)$ et

$$s_l := S_l(\underline{\alpha}) \quad \forall l \in \mathbb{N}.$$

- ***étape 1 :*** on calcule les $\delta^k s_k$ pour k allant de 1 à b ;
- ***étape 2 :*** on calcule les $\delta^{rk} s_{rk}$ pour k allant de $[b/r] + 1$ à b ;
- ***étape 3 :*** on calcule $k\delta^{rk} b_k := \sum_{i=0}^{k-1} \delta^{rk} b_i s_{r(k-i)}$ pour k allant de 1 ;
- ***étape 4 :*** on calcule b_k pour k allant de 1 ;
- ***étape 5 :*** on renvoie le polynôme $\sum_{k=0}^{b} b_k T^k$.

étape 1 : on applique les relations de récurrences linéaires (i) de la proposition (1.2.2). Pour tout entier k tel que $1 \le k \le b$, le calcul de $\delta^k s_k$ à partir des $(\delta^j s_j)_{1\le j\le k-1}$ se fait en $2k-1$ opérations dans $\mathbb{A}$. Le nombre d'opérations pour le calcul de $(s_k)_{1\le k\le b}$ est donc $O(b^2)$. Ainsi la complexité de l'étape 1 vaut

$$O\left(b^2 N^{1+\epsilon}\right).$$

étape 2 : la suite $(\delta^{rk} s_{rk})_{1\le k\le [b/r]}$ est déjà calculé dans l'étape 1. En appliquant l'algorithme **SuiteRec**, on calcule $\delta^{rk} s_{rk}$ en $O(b^{1+\epsilon}\log rk)$ étapes pour tout entier tel que $[b/r] + 1 \le k \le b$.

Le nombre d'opérations dans l'étape 2 est

$$O\left(b^{1+\epsilon}\log\left(r\frac{b!}{[b/r]!}\right)\right) = O(b^{2+\epsilon} + b^{1+\epsilon}\log r).$$

Ainsi la complexité de l'étape 2 vaut

$$O\left(b^{1+\epsilon}(b+\log r)N^{1+\epsilon}\right).$$

étape 3 : on applique les relations de récurrences linéaires (ii) du corollaire (1.2.2). Pour être exacts, les calculs doivent se faire modulo $\delta^{rk} b M_Q$ (on prend alors $N = [\log_p(b)M_Q + rb\log_p \delta] + 1$).

Pour tout entier k tel que $1 \le k \le b$, le calcul de b_k à partir des $(b_j)_{1\le j\le k-1}$ se fait en $2k+1$ opérations dans $\mathbb{A}$. Le nombre d'opérations dans l'étape 3 est donc $O(b^2)$.

Ainsi la complexité de l'étape 3 vaut

$$O\left(b^{2+\epsilon}N^{1+\epsilon}\right).$$

□

Remarque 2.2.4. *La complexité qui nous intéresse est celle en r. On déduit de la proposition (2.2.2) que pour avoir des algorithmes efficaces, il sera plus intéressant d'avoir des informations sur les racines de P que sur la norme de Gauss de P car $p_{r,b}$ est exponentielle en r (voir la section (I.3) de [F-S09]).*

Remarque 2.2.5. *(1) On pourrait aussi exploiter le corollaire (1.2.4), il faudrait chercher les solutions des équations du type*

$$|\underline{v}|_1 = m \tag{3}$$

où $\underline{v}$ est un élément de $\mathbb{N}^b$ et m est un entier naturel.
D'après la section (I.3) de [F-S09], on voit que le nombre d'étapes est exponentiel en r. La formule de la proposition (1.2.4) n'est pas efficace.

(2) On peut s'attendre à ce que la complexité d'un algorithme exploitant le résultant ait une complexité au moins en r^2 ce qui est moins bon que notre algorithme exploitant les formules de récurrence. On pourrait certainement faire mieux car la matrice dont on calcule le déterminant est creuse.

2.3 Application au calcul de la fonction zêta d'une variété

On fixe une variété X sur $\mathbb{F}_q$.

Pour tout entier n supérieur à 1, on note $X(\mathbb{F}_{q^n})$ l'ensemble des points $\mathbb{F}_{q^n}$-rationnels de X. La **fonction zêta de** X est définie par

$$Z(X,T) := \exp\left(\sum_{n\geq 1}\frac{\#X(\mathbb{F}_{q^n})}{n}T^n\right).$$

Définition 2.3.1. *Soit μ un entier naturel et R un polynôme à coefficients entiers de degré b tel que $R(0)$ soit non nul.*
Si b est supérieur à 1, décomposons R en

$$R(T) =: R(0)\prod_{j=1}^{b}(1-\gamma_j T)$$

où $(\gamma_j)_{1\leq j\leq b}$ est un élément de $\overline{\mathbb{Q}}^b$.
*On dit que R est un **polynôme mixte (relativement à q) de poids inférieurs à** μ si R est constant ou si*

$$\forall j\in\{1,\dots,b\},\ \exists\mu'\in\mathbb{N},\ \mu'\leq\mu,\quad |\gamma_j| = q^{\mu'/2}.$$

Théorème 2.3.2. *La fonction zêta de X s'écrit*

$$Z(X,T) = \prod_{j=0}^{2d}P_j(T)^{(-1)^{j+1}}$$

où pour tout entier j tel que $0\leq j\leq 2d$, P_j est un polynôme à coefficients entiers mixte de poids inférieurs à j.

Démonstration. Le rationalité découle de la formule des traces de Grothendieck-Lefschetz (voir le théorème (13.4) de [Mil80]).

La mixité découle du théorème (4.1) de [LS04]. □

Proposition 2.3.3. *On a*

$$Z(X_{\mathbb{F}_{q^r}}, T^r) = \prod_{i=0}^{r-1} Z(X, \xi^i T)$$

où ξ est une racine primitive r-ième de l'unité.

Ce résultat ne dépend pas du choix de la racine primitive r-ième.

Démonstration. Pour tout entier naturel ν, on a

$$\sum_{i=0}^{r-1} \xi^{i\nu} = \begin{cases} r & \text{si } r \text{ divise } \nu\,; \\ 0 & \text{sinon.} \end{cases}$$

Ainsi

$$\begin{aligned} Z(X_{\mathbb{F}_{q^r}}, T^r) &:= \exp\left(\sum_{n\geq 1} \frac{\#X_{\mathbb{F}_{q^r}}(\mathbb{F}_{q^{rn}})}{n} T^{rn}\right) \\ &= \exp\left(\sum_{n\geq 1} \frac{\#X(\mathbb{F}_{q^{rn}})}{n} T^{rn}\right) \text{ car } \#X(\mathbb{F}_{q^{rn}}) = \#X_{\mathbb{F}_{q^r}}(\mathbb{F}_{q^{rn}}) \\ &= \exp\left(\sum_{i=0}^{r-1} \sum_{\nu\geq 1} \frac{\#X(\mathbb{F}_{q^\nu})}{\nu} \xi^{\nu i} T^\nu\right) \\ &= \prod_{i=0}^{r-1} Z(X, \xi^i T). \end{aligned}$$

□

Proposition 2.3.4. *Supposons que l'on connaisse la fonction zêta de X et que sa décomposition soit comme dans le théorème (2.3.2)*

$$Z(X, T) = \prod_{j=0}^{2d} P_j(T)^{(-1)^{j+1}}$$

Alors la fonction zêta de $X_{\mathbb{F}_{q^r}}$ peut être calculée en

$$O\left(db^{3+\epsilon} r^{1+\epsilon} (d \log_p q + \log_p \delta)^{1+\epsilon}\right)$$

où

$$b := \max_{0\leq j\leq 2d} \deg(P_j) \quad \textit{et} \quad \delta := \max_{0\leq j\leq 2d} |P_j(0)|.$$

Démonstration. Soit j un entier tel que $0 \leq j \leq 2d$. On note

$$Q_j(T^r) := \prod_{i=0}^{r-1} P_j(\xi^i T)$$

où ξ est une racine primitive r-ième de l'unité.

D'après le paragraphe (1.1.2), Q_j est un polynôme de $\mathbb{Z}[T]$ de degré $\deg(P_j)$. On a le lemme suivant :

Lemme 2.3.5. *Le polynôme Q_j est exact modulo $2^b\delta^r q^{dbr}$.*

Démonstration. On a d'après le point (ii) de la proposition (2.2.2)

$$\|Q_j\| \leq 2^{\deg(P_j)}|P_j(0)|^r q^{j\deg(P_j)r/2} \leq 2^b\delta^r q^{dbr}.$$

□

On définit alors l'algorithme suivant :

ZetaExt$((P_j)_{0\leq j\leq 2d}, r, M)$

Entrée : la famille $(P_j)_{0\leq j\leq 2d}$ des polynômes apparaissant dans la décomposition de la fonction zêta de X, et un entier r supérieur à 1.
Sortie : la famille $(Q_j)_{0\leq j\leq 2d}$ des polynômes apparaissant dans la décomposition de la fonction zêta de $X_{\mathbb{F}_{q^r}}$.

- ***étape 1 :*** pour j allant de 0 à $2d$,
  - ◇ on calcule $Q_j = P_j^r$ si P_j est constant ;
  - ◇ on calcule le polynôme $Q_j =$ **AlgoRec**(P_j, r, M) sinon.
- ***étape 2 :*** on renvoie la famille $(Q_j)_{0\leq j\leq 2d}$.

On déduit de la proposition (2.2.3) que cet algorithme a une complexité de

$$O\left(db^{1+\epsilon}(b+\log r)(M+rb\log_p\delta)^{1+\epsilon}\right).$$

D'après le lemme (2.3.5), il suffit de calculer

$$\textbf{ZetaExt}((P_j)_{0\leq j\leq 2d}, r, M)$$

où M est un entier naturel tel que $p^M \geq 2^b\delta^r q^{dbr}$. La complexité du calcul est

$$O\left(db^{1+\epsilon}(b+\log r)(M+rb\log_p\delta)^{1+\epsilon}\right) = O\left(db^{3+\epsilon}r^{1+\epsilon}(d\log_p q+\log_p\delta)^{1+\epsilon}\right).$$

□

Exemple 2.3.6. *Soit C une courbe hyperelliptique de genre g sur $\mathbb{F}_p$ avec un point de Weierstrass rationnel. La fonction zêta de C s'écrit*

$$Z(C,T) = \frac{P_1(T)}{(1-T)(1-pT)}$$

où $P_1(T) = 1 + a_1T + \ldots + a_{2g}T^{2g} \in \mathbb{Z}[T]$. On peut calculer la fonction zêta de $C_{\mathbb{F}_{p^r}}$ de deux façons :

(1) D'après [Ked01], la complexité du calcul de $Z(C_{\mathbb{F}_{p^r}}, T)$ est un

$$O(g^{4+\epsilon}r^{3+\epsilon}).$$

*(2) Toujours d'après [Ked01], la complexité du calcul de $Z(C,T)$ est $O(g^{4+\epsilon})$. Par l'algorithme **ZetaExt**, on peut alors calculer la fonction zêta de $C_{\mathbb{F}_{p^r}}$ en*

$$O(g^{7+\epsilon}r^{1+\epsilon}).$$

L'algorithme développé a donc une meilleure complexité en r que l'algorithme de Kedlaya pour le calcul de la fonction zêta de $C_{\mathbb{F}_{p^r}}$.

Références

[F-S09] Philippe Flajolet et Robert Sedgewick, *Analytic Combinatorics*, Cambridge University Press, (2009).

[Ked01] K. Kedlaya, *Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology*, J. Ramanujan Math. Soc., 16, (2001).

[LS04] Bernard Le Stum, *Frobenius action, F-isocrystals and slope filtration*, Geometric aspects of Dwork theory. Vol. I, II, Walter de Gruyter GmbH & Co. KG, Berlin, 837-843, (2004).

[Mil80] James Stuart Milne, *Étale Cohomology*, Princeton Math. Series vol. 33, Princeton University Press, (1980).